Improving LSB Algorithm Using Filtering and Matching

Javad Mohammadi Rad

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
September 2012
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Alexander Chefranov
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Alexander Chefranov
2. Asst. Prof. Dr. Gürcü Öz
3. Asst. Prof. Dr. Önsen Toygar

ABSTRACT

In the age of computers and communications, different techniques are developed in order to protect the information against illegal accesses and attacks. Steganography is one of these techniques which embeds secret information in a media such as image, sound, video, and etc. in a way that it is not detectable by others.

Focusing on the image, Least Significant Bit (LSB) method is one of basic methods of image steganography in which the least significant bit of pixel colors are replaced with the bits of secret message simply.

Selected Least Significant Bits (SLSB) method improves simple LSB method by embedding secret message bits to one of red, green, or blue color along filtering and matching approaches [1].

In the present thesis, a new filtering method which improves LSB method has been proposed. The filtering method selects a color or colors of pixel for embedding according to their most significant bits (MSB; the bits of color which do not participate in embedding) value. The colors whose most significant bits value are greater than a particular threshold will participate in embedding and the other ones will be skipped. This mechanism makes the recovery of hidden message possible. More, a matching technique ensures the most possible closeness of new generated color in the embedding process to its original color. In other words, first bit next to the least significant bits would change in favor of closing new generated color to its origin and lead to less visibility of stego image degrading.

Quality of embedding is measured by some statistical metrics namely AAD, MSE, LP-Norm, LMSE, SNR, PSNR, NCC. Applying the same cover images and secret messages to forenamed methods, the new proposed method offers up to %50 better results in some metrics in comparison to LSB and SLSB methods.

Keywords: LSB (Least Significant Bit), Filtering, Matching, Statistical Metrics.

ÖZ

Bilgisayar ve iletişim çağında, farklı teknikler yasadışı girişler ve saldırılara karşı bilgileri korumak amacıyla geliştirilmiştir. Steganografi bu tekniklerden biridir ki resim, ses, video gibi bir ortam içine başkalarının anlayamayacağı şeekilde gizli bilgileri gömer.

Görüntü üzerinde yoğunlaşırken, 'en önemsiz bit' (LSB) steganografinin en önemli temel yöntemlerinden biridir ki pikselin en az anlamlı bitini gizli mesaj ile değiştirir.

Seçilen en önemsiz biti (SLSB) filtreleme yöntemi boyunca kırmızı, yeşil veya mavi renklerden birine gizli mesaj bit gömme ve yaklaşımlar [1] 'eşleştirerek basit' (LSB) yöntemi geliştirir.

Bu tez çalışmasında, LSB yöntemi geliştirilmiş ve yeni bir filtreleme yöntemi önerilmiştir. Filtreleme yöntemi, en önemli biti (MSB; gömme işlemine katıılmayan renk biti) değerine göre gömmek için bir renk veya piksel renkleri seçer. Belli bir threshold değerinden yüksek olan en önemli bitin renkleri gömmeye katılacak, fakat diğerleri atlanacaktır. Bu mekanizma, gizli mesaj geri kazanımını mümkün kılar. Buna ek olarak eşleştirme tekniği, gömme işleminde üretilen yeni rengin orjinal renge mümkün olan en yakın renk olmasını sağlar. Diğer bir deyişle, en az önemli bitlerin yanındaki ilk bit orjinaline en yakın olabilecek şekilde yeni bir renge dönüşür ve stego görüntüsünü en aza indirir.

Gömme Kalitesi bazı istatistiksel ölçümler yani AAD, MSE, LP-Norm, LMSE, SNR, PSNR, NCC tarafından ölçülür. Önerilen yöntem, bahsedilen ölçümlerle aynı resimler ve gizli mesajlar kullanılarak, diğer iki metotla karşılaştırıldığında %50 daha iyi sonuç vermektedir.

Anahtar kelimeler: En önemsiz bit (LSB), Filtreleme, Eşleştirme, İstatistiksel ölçümler

ACKNOWLEDGMENT

I would never been able to complete this dissertation without the help of the people who have supported me with their best wishes.

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Alexander Chefranov for his efforts and supports for doing this research. I sincerely thank to the committee members of my thesis defense jury for their helpful comments on this thesis. I gratefully acknowledge the chairman of computer Engineering Department, Assoc. Prof. Dr. Muhammed Salamah.

Last but not least, I would also like to thank my dear wife and parents for their continuous supports.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AAD	Absolute Average Difference
b/p	bit per pixel
BMP	Bitmap
ILSB	Improved Least Significant Bit
ILSB M	Improved Least Significant Bit using LSB Matching
LMSE	Laplacian Mean Squared Error
LPNorm N	L^p-norm, new one
LPNorm O	L^p-norm, old one
LSB	Least Significant Bit
MAE	Mean Absolute Error
MSE	Mean Squared Error
NCC	Normalized Cross Correlation
PSNR	Peak Signal to Noise Ratio
RGB	Red Green Blue
SLSB	Selected Least Significant Bit
SNR	Signal to Noise Ratio

Chapter 1

INTRODUCTION

1.1 Steganography

According to development of the Internet and large amount of transmitting information in the modern world, necessity of information security is obviously felt. Ciphering techniques are widely used to encrypt and decrypt data. But sometimes data encryption does not seem enough and hiding of the data itself is needed more. The technique used for this idea is called *Steganography*. Steganography is the process of concealing information in a carrier such as text, image, voice, video, or protocol. Digital images are one of the common and most popular ones due to their frequency on the internet and high capacity of data transmission without degrading effect on images quality [2].

1.2 History

The word *Steganography* literally means *concealed writing* which is originally derived from Greek words *steganos* and *graphie* meaning *covered* and *writing* respectively [3]. "The first recorded uses of steganography can be traced back to 440 BC when Herodotus mentions two examples of steganography in his Histories. Demaratus sent a warning about a forthcoming attack to Greece by writing it directly on the wooden backing of a wax tablet before applying its beeswax surface. Wax tablets were in common use then as reusable writing surfaces, sometimes used for shorthand" [3].

1.3 Definitions

"An *image* is a collection of numbers that constitute different light intensities in different areas. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the Internet consist of a rectangular map of the image's pixels (represented as bits) where each pixel is located and its color. These pixels are displayed horizontally and row by row" [2].

In image steganography, different types of images are used according to their features. In the present thesis, Windows Bitmap (BMP) format has been applied for all embedding methods. Pixel color in BMP format consists of three basic colors red, green, and blue (RGB) each of which uses eight bits to represent corresponding color. These twenty four bits which determine the color intensity of the pixel is called *Color Depth* of the image.

The image in which the secret message is embedded is called *cover image* and the image containing the secret message is *stego image* [4].

Some algorithms which deal with images belong to *Spatial Domain*. It means that they exert the changes on the image itself and do not change pixels intensity before embedding [1].

In spatial domain, some algorithms belong to *Non-filtering algorithms* group. These types of algorithms are the simplest and most vulnerable ones. They start from the first pixel of image and embed the secret message sequentially in the pixels leaving remaining pixels unchanged [1].

The other ones, *Filtering algorithms* group that filter most significant bits of pixel colors. Bits of secret message are embedded in the least significant bits of the pixels which obtain better rates [1].

1.4 Outline

In Chapter 1 of this thesis, main field of the present research, steganography, will be introduced containing the origin of the steganography, the history and some useful definitions. In Chapter 2, two of previous methods in steganography will be explained and investigated by an appropriate tool called embedment map. In Chapter 3, the new proposed method, its types, accessories and evaluation will be mentioned and explained. Results and experimental setup will be declared in Chapter 4 and finally in the Chapter 5, we have a conclusion of this thesis.

Chapter 2

REVIEW OF THE PROBLEM AND KNOWN METHODS

2.1 Problem Definition

Steganography is the process of embedding a secret message in a carrier in a hidden manner not detectable by the others. Different methods are applied to embed the secret message with different characteristics. Simple LSB replaces the secret message bits consecutively in the color of the pixels from the beginning of the image. Due to this, it is able to cover a significant size of secret message. The SLSB method uses some analyses to choose one color among three ones for embedding and a filtering method for choosing some particular pixels. Because of this scattering of data, the size of secret message can be embedded will reduce and consequently the quality of embedding will be increased. In this thesis, we calculate and apply a particular threshold to increase the dispersal of secret message bit in the cover image to achieve a better quality of embedding. Some metrics evaluate this quality.

2.2 LSB Method

2.2.1 Description

LSB is a non-filtering algorithm in spatial domain. According to [2], Least Significant Bit (LSB) approach is a simple, basic method for embedding information in an image in which the least significant bit of the colors (RGB) of the pixels in the image is replaced with a bit of the secret message. Using a 24-bit image, a bit of each of the colors, red, green, and blue is used for embedding, since each one is considered as a byte. In other words, three bits can be embedded in each pixel. For instance, a 100 × 100 pixel image, can store amount of 30,000 bits or 3,750 bytes of secret message.

More details, consider two pixels of a 24-bit image is as follows:

(01111011   00111001   11010011)

(10010001   01001110   11000100)

whose decimal representations are respectively

(123   57   211)

(145   78   196).

When the number 33, whose binary representation is 100001, is embedded into the least significant bits of the two pixels, the result is as

(0111101**1**   0011100**0**   1101001**0**)

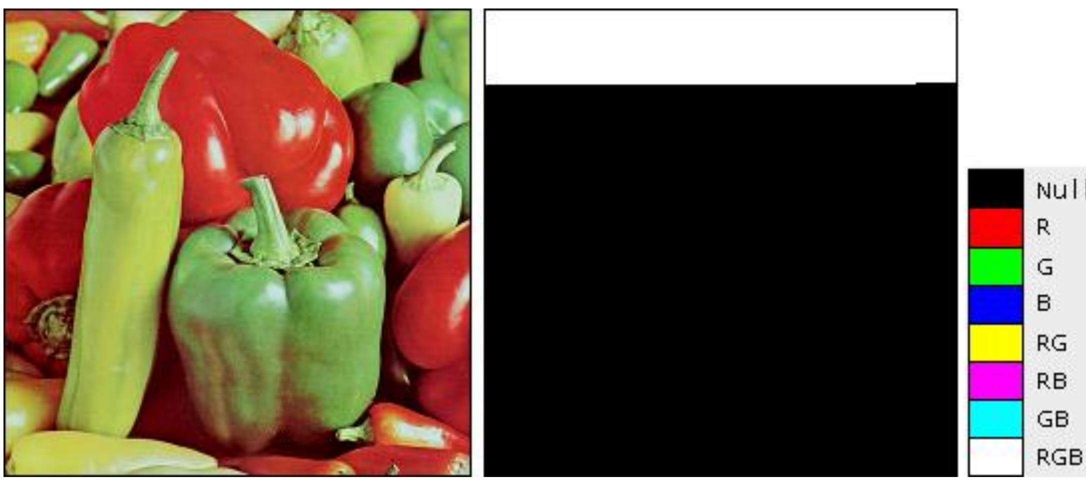
(1001000**0**   0100111**0**   1100010**1**).

Although the number 33 is embedded into the six bytes, but only the four underlined bits are changed according to the number and pixels values.

On average, almost half of the bits in an image may be modified in embedding process using the maximal covering capacity. These changes will not be perceived by the human eyes. Thus the secret message can be successfully hided. Choosing an appropriate image, it may be possible to even embed the message in the first two least significant bits and the difference of cover and stego images is still not seen. In forenamed example, bits of secret message are placed consecutively from the beginning to the end and this will cause the secret message be easily extractable. A more secure LSB method is to share a secret key as a seed for a pseudo-random number generator to specify some pixels for embedding. This type of LSB method will be categorized in randomized algorithms of spatial domain.

2.2.2 Embedment Dispersal

In order to know and show how secret message bits are embedded within stego image and because the stego image presents no changes to human eyes, embedment map has been used to illustrate embedding pixels. In other words, dispersal of hidden message bits in cover image is appeared in this map. Different colors in the map represent for embedding color or colors.



a b c

Figure 1: LSB embedment dispersal for two bits per color. a. Original Pepper.bmp image. b. Embedment map. c. Map guide.

According to the Figure 1-b and Figure 1-c, hidden message bits are embedded in all red, green, and blue colors of the white area consecutively. The embedment area is started from the first pixel of the Figure 1-a (cover image) to the beginning of the black area which contains no embedment.

2.3 SLSB Method

2.3.1 Description

This method belongs to filtering algorithms group of spatial domain. Due to modification of all red, green, and blue colors in LSB method, a distortion is generated. Although the changes are not visible to human eyes, but they would be detectable by some statistical analyses such as RS analysis [5] or Sample Pairs [6]. In return, Selected Least Significant Bit (SLSB) method benefits from choosing one color out of three (RGB). In order to choose the color for embedding message an analysis which is called Sample Pairs is performed [7]. The color with higher ratio that offers more diversity and would cause less noticeable changes will be selected [1]. Considering secret bit string 111 and a pixel as follows:

(11101000   11101000   11101000)

whose decimal representation is

(232   232   232).

Performing LSB method to embed bit string on this pixel will result

(1110100**1**   1110100**1**   1110100**1**)

and decimal values of new color will be

(233   233   233).

But using SLSB method and assuming selection of green color of the pixel for embedding cause to result

(11101000   11101**111**   11101000)

whose corresponding decimal values are

(232   239   232).

There is a leap only in green color as shown above.

According to [8] and [9], another important concept which works here in SLSB is LSB match adaptation. It ensures the new generated color in the embedding process to get close as much as possible to its original color. In other words, adaptation bit would change in favor of closing new generated color to its origin and in order to make less visibility of stego image degrading [1].

In the above mentioned example, the pixel changes from

(11101000   11101**111**   11101000)

to

(11101000   1110**<u>0</u>111**   11101000)

with decimal values

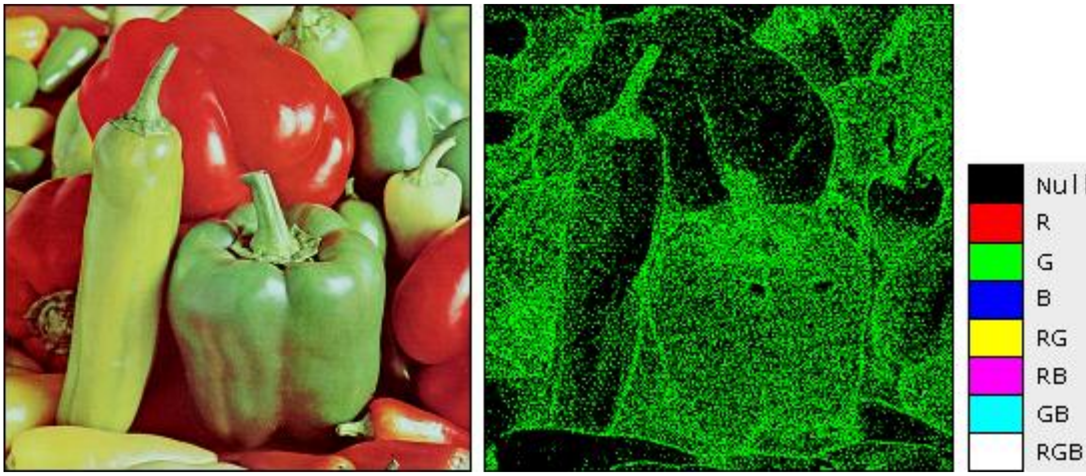
(232   231   232).

This pixel is much closer to the original pixel. The difference is just one unit in green color.

It is necessary to add, an application that is issued by the author of [1] is applied to perform SLSB method on the cover images in experiments. To be trustworthy, the values of the metrics obtained from my application are checked and compared with the issued application by the author of [1] to be calculated similarly. Although, some mistakes are found in his calculations and correct formulas and values are added.

2.3.2 Embedment Dispersal

The dispersal of hidden message bits in the pepper.bmp image using SLSB method and two bits per color (here green) is shown in the Figure 2.



a b c

Figure 2: SLSB embedment dispersal for two bits per color (green). a. Original Pepper.bmp image. b. Embedment map. c. Map guide.

As illustrated in the Figure 2-b and according to the Figure 2-c, SLSB method has chosen the color green out of red, green, and blue for embedding the secret message bits in the Figure 1-a (cover image) according to the statistical analysis result.

Chapter 3

PROPOSED METHOD DESCRIPTION AND ACCESSORIES

3.1 Proposed Method, Improved LSB

Considering each color of the pixels in cover image as a byte, each byte is divided to three parts in the proposed method. First part is most significant bits part which contains some of most significant bits of the color. Second part is matching or adaptation bit part used to perform adaptation concept. Adaptation concept was explained and exampled in Section 2.3.1 of the Chapter two. The last remained part which can be one or more than one bit is considered as least significant bit(s) part. To have a better perception, color structure in proposed method is illustrated in Figure 3. It is assumed that two bits are used in embedment process.

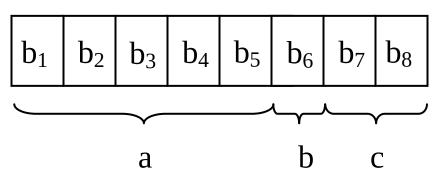


Figure 3: Color (Byte) structure in a pixel. a. Most significant bits (Filtering bits). b. Adaptation bit. c. Least significant bits. (Embedding bits)

If the most significant bits value of the color with considering parts b and c's bits as zero (in Figure 3, $b_1b_2b_3b_4b_5000$'s value is equal to $b_1\times2^7 + b_2\times2^6 + b_3\times2^5 + b_4\times2^4 + b_5\times2^3 + 0\times2^2 + 0\times2^1 + 0\times2^0$) is greater than or equal to a particular threshold, the color will be qualified to participate in embedment and the least significant bit(s) of the color will be replaced with corresponding bit(s) of secret message. Consequently, adaptation concept checks the most closeness of the stego color to the original color, otherwise the color is skipped. This procedure will be started from the first pixel of the image and iterated sequentially until the secret message is covered completely. If number of the pixels in cover image which are qualified to participate in embedding is not adequate, then the cover image is not able to embed the whole message by using that threshold and it is needed to decrease the threshold in order to embed the whole message.

Some flowcharts are used to define proposed method. The main procedure is as follows:

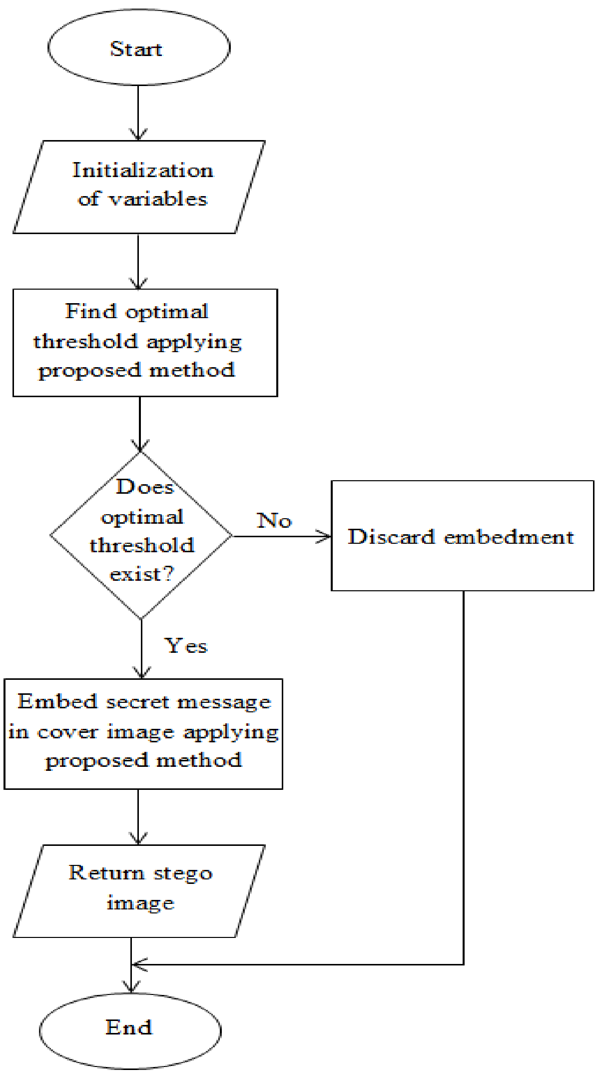


Figure 4: Embedment flowchart applying proposed method

Calculation of optimal threshold is as follows:

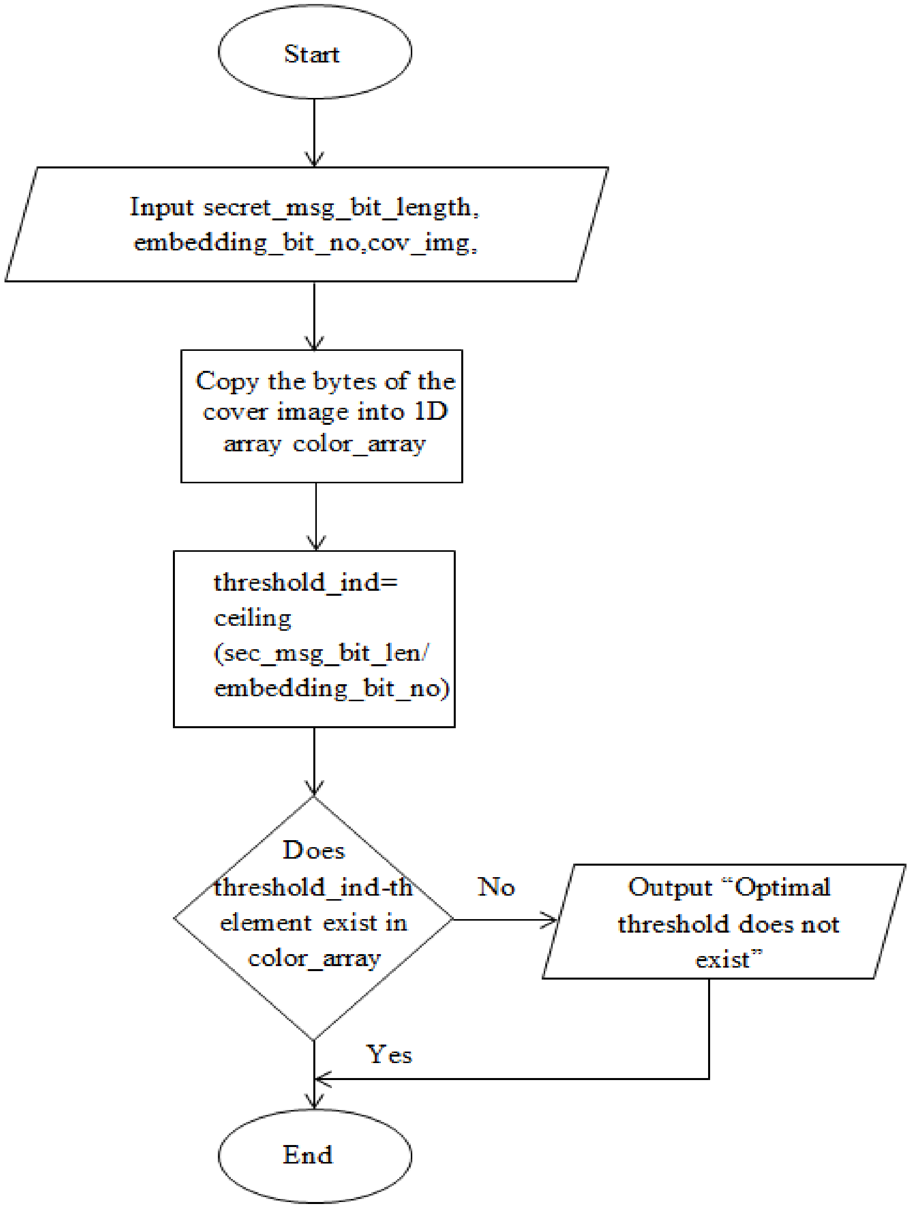


Figure 5: Flowchart of obtaining optimal threshold

Flowchart of initialization is as follow:

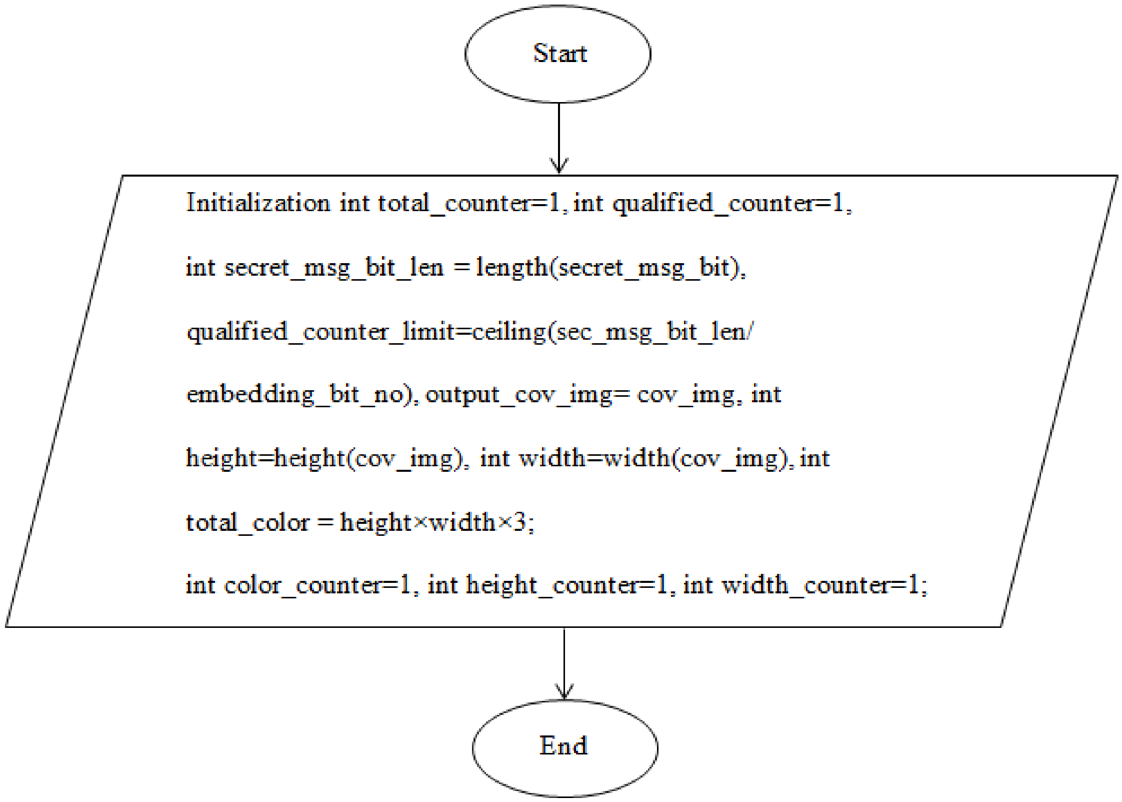


Figure 6 Flowchart of initialization

Flowchart of proposed embedment method is as follows:

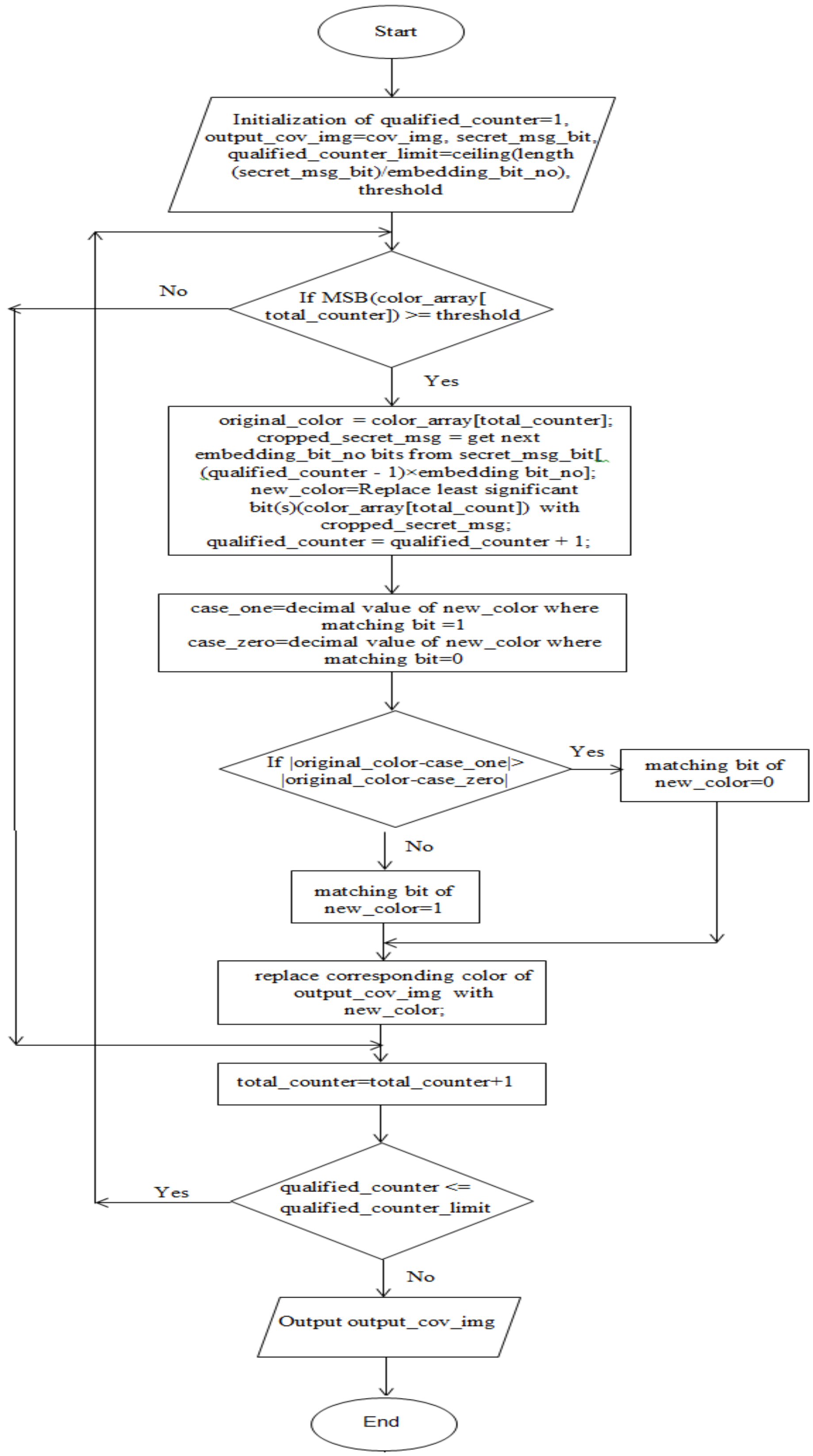


Figure 7 Flowchart of proposed embedment method

The Pseudo code of the proposed method is as follows:

```
/* inputs and variables */
image cov_img, output_cov_img;            /* input and output images */
bitstring secret_msg_bit;
int embedding_bit_no;               /* will be input by user */
int total_counter=1;                /* holds number of checked colors */
int qualified_counter=1;            /* holds number of qualified colors */
int secret_msg_bit_len= length(secret_msg_bit);
input embedding_bit_no;
input cov_img;
qualified_counter_limit =ceiling(secret_msg_bit_len/embedding_bit_no);
int threshold;
output_cov_img= cov_img;
input secret_msg_bit;
bitstring cropped_secret_msg;
int original_color;             /* holds the original value of the color */
bitstring new_color;            /* holds the new value of the color after embedment */
int height=height(cov_img);
int width=width(cov_img);
int total_color = height×width×3;
int color_counter=1;
int height_counter=1;           /* provides the height of the pixel to be checked */
int width_counter=1;            /* provides the width of the pixel to be checked */
intarray color_array[total_color];          /* to hold all colors of the pixels in 1D array */
```

```
int case_one;

int case_zero;

/* start of the code */

{

    /* putting all colors of the cover image into a 1D array */

    copy the bytes of the cov_img into 1D array called color_array;

    sort color_array in descending manner;

    threshold is the qualified_counter_limit-th value of color_array;

    if qualified_counter_limit-th value of color_array does not exist then

    {

       Output “optimal threshold does not exist”;

       Exit;

    }

    while (qualified_counter <= qualified_counter_limit)          /* ensures embedment of
    whole  the message */

    {

        if (most significant bits value(color_array[total_counter]) >= threshold)

        /* ensures the color to be qualified */

        {

           original_color = color_array[total_counter];    /* to be used in adaption */

           cropped_secret_msg = get next embedding_bit_no bits from secret_msg_bit[
           (qualified_counter - 1)×embedding bit_no];              /* to be replaced to lsb of
           the color */
```

```
            new_color=Replace least significant bit(s)(color_array[total_count]) with
            cropped_secret_msg;
            qualified_counter = qualified_counter + 1;          /* counter increment */
            /* matching (adaption) */
            case_one = decimal value of new_color by changing the matching bit to 1;
            case_zero = decimal value of new_color by changing the matching bit to 0;
            if |original_color – case_one| > |original_color – case_zero| then matching bit
            of new_color=0;
            else matching bit of new_color=1;
            replace corresponding color of output_cov_img  with new_color;
        }
        total_counter= total_counter + 1;
    }
    Show output_cov_img;
}       /* End of the code */
```

As an example, consider the pixel

(10110011   00110011   11011000)

with decimal representation

(179   51   216)

and assume that we are going to embed the bit string 111 in this pixel using three least significant bits and the threshold 185. Considering three least significant bits and one adaptation bit, we will have four most significant bits. To assess red color, most significant bits value of red color (considering adaptation and least significant bits as zero) will be **1011**0000 with decimal value of 176 which is less than 185. Then the color red is not qualified to participate in embedding.

Assessing green color, most significant bits value of green color, **0011**0000 (enumerating adaptation and least significant bits as zero) costs 48 in decimal representation which is less than 185. Thus, the message will not be embedded in green color.

Evaluating blue color, most significant bits value of the blue color enumerating adaptation and least significant bits as zero will be **1101**0000 which results 208 in decimal base. It is greater than 185. Thus, the color satisfies the criteria and the message will be embedded in blue color. The result is

(10110011   00110011   11011**111**)

whose decimal representation is

(179   51   223).

Referring to [8] and [9], considering matching (adaptation) concept, the blue color will be changed to

(10110011   00110011   1101**0111**)

with decimal representation of

(179   51   215).

The matching bit is changed from one to zero in order to reduce the difference of new and original colors. In other words, the matching bit is changed to zero, because

$$|216 - 215| < |216 - 223|.$$

216 is the original value of blue color of the pixel. 223 is the color generated after embedding the bit string 111 in the blue color without changing adaptation bit and 215 is

the color generated by embedding and changing the adaptation bit from one to zero. Thus, as shown above, changing adaptation bit from one to zero will cause to an improvement in embedment quality.

In another case, consider the bit string 111001 to be embedded in the pixel

(10111001   00110011   11000100)

with decimal values of

(185   51   196)

by using three least significant bits and the threshold 176. Most significant bits value of each color (enumerating the adaptation bit and least significant bits as zero) will be compared to the threshold. Most significant bits value of red color, **1011**0000, with decimal value of 176 is equal to the threshold, 176. Then the first three bits of secret message will be embedded in red color.

Evaluating the color green, most significant bits value of green color enumerating adaptation and least significant bits as zero, **0011**0000, with decimal value of 48 is less than the threshold 176. Thus green color will not participate in embedment.

About the color blue, most significant bits value of blue, **1100**0000, with decimal representation of 192, satisfies the criteria and consequently the second three bit of secret message will be embedded in the three least significant bits of color blue. New pixel colors will be

(10111**111**   00110011   11000**001**)

with decimal representation of

(191   51   193).

Using adaptation concept will change the pixel to

(1011**0111**   00110011   11000**001**).

Considering the adaptation bit of the red color as zero makes the new generated color closer to the original red color. In other words,

$$|185 - 183| < |185 - 191|$$

in which 185 is the original red color. 191 is new generated color not using adaptation concept and 183 is the new generated color using adaptation concept. Thus, adaptation concept will be applied.

But the story is not the same about the blue color and change of adaptation bit will not reduce the difference of new and original color. In other words,

$$|196 - 201| \nless |196 - 193|.$$

Thus, the adaptation bit of the color blue will not change.

According to [1], algorithms in which pixel intensities are not modified before embedding belong to spatial domain. These algorithms exert the changes directly on the cover image. According to this, the proposed method belongs to spatial domain and filtering algorithms group due to selecting some pixels among the all.

It seems necessary to add that an application is prepared by me to perform the LSB and ILSB and calculate the metrics for evaluation of method.

3.2 Calculating a Particular Threshold

Proposed method benefits from a particular threshold in order to determine the pixels that are going to participate in embedding. The colors whose most significant bits values are greater than the threshold are allowed to participate in embedding.

To obtain the optimal threshold, all color values of pixels, *n* values, will be sorted in descending order. Assuming *k* as the number of needed color for embedding message according to length of message and number of using bit in each color, the k-th value of sorted color values refers to the threshold. This is the maximal possible threshold which ensures embedding of the whole message with maximal dispersal. The visual presentation of this story is shown in Figure 4.

c_1	c_2	c_3	...	c_{k-1}	c_k	c_{k+1}	...	c_{n-2}	c_{n-1}	c_n

Figure 8: Optimal threshold determination. The optimal threshold is the k-th color, $\boldsymbol{c_k}$, in sorted array of colors ($\boldsymbol{c_i} \geq \boldsymbol{c_{i+1}}$).

3.3 Embedment Dispersal

As stated before, we have used embedment map to show how hidden message bits are embedded and distributed in the cover image. Since LSB method uses colors' capacity three times more than SLSB (three colors versus one color) we have applied new proposed method in two cases. In the first case, proposed method is allowed to use three colors of the pixels of cover image whose embedment map is as shown in the Figure 5-b.

a b c

Figure 9: Improved LSB method embedment dispersal using all colors for two bits per color. a. Original Pepper.bmp image. b. Embedment map. c. Map guide.

In this case, all colors of the pixel and all of their combinations are allowed to embed secret message. Due to this, the results are comparable to LSB method. With respect to the value of the most significant bits of the color and as the Figure 5-c shows, eight cases may occur to a pixel. All qualified and non-qualified pixels will be mapped to one of these eight colors. The pixels in the Figure 5-a corresponding to the red pixels in the Figure 5-b (embedment map) have most significant bits values of red color greater than or equal to a particular threshold. In other words, the pixels of the Figure 5-a corresponding to the red pixels in the Figure 5-b are qualified to participate in embedment with their red colors due to their most significant bits value greater than or equal to the threshold. The pixels in the Figure 5-a corresponding to the white pixels of the Figure 5-b (embedment map) have most significant bits values greater than or equal to the threshold in all red, green, and blue colors and naturally will participate in embedment with their all colors, red, green, and blue shown with white pixels in embedment map. In the same way, the pixels in the Figure 5-a corresponding to the yellow pixels in the Figure 5-b (embedment map) embed the secret message bits in their red and green colors because most significant bits values of these two colors are equal to or exceed the threshold. On the other hand, pixels of the Figure 5-a corresponding to the black pixels in the Figure 5-b which have most significant bits values smaller than the threshold do not participate in embedment.

In the second case, to have new proposed method comparable to SLSB method, proposed method has been applied to one of red, green, or blue color, but not to all of them. The result is shown in Figure 6.

a b c

Figure 10: Improved LSB method embedment dispersal using one color for two bits per color (red). a. Original Pepper.bmp image. b. Embedment map. c. Map guide.

Here, according to goodness of the embedment, red color has been chosen. Since the pixels in the Figure 6-a (cover image) corresponding to the red pixels the Figure 6-b (embedment map) have the most significant bits values greater than or equal to the threshold in their red colors, these pixels are qualified to participate in embedment with their red colors. Naturally, the threshold determined for embedment in one color, 152 (available in pepper.bmp results in Chapter 4), will be less than the threshold determined for embedment in all three colors, 184 (available in pepper.bmp results in Chapter 4). Because applying one color per pixel needs more bits to cover the lack of the other two colors (green and blue) versus LSB method allowed to apply three colors. Thus, the threshold is reduced in order to embed the whole message.

3.4 Evaluating Statistical Metrics

What makes us be able to measure the quality of an image in comparison to another, is expressing the differences quantitatively. In this direction, we have used some statistical metrics which survey difference of stego image versus cover images in different aspects.

3.3.1 AAD

According to [10] and [11], AAD (Average Absolute Difference) gives the average absolute value of difference of input and output images per pixel. Lower value of AAD is more desired. Complete similarity of input and output images will result value zero of this metric. Due to absolute value, it is always non-negative. MAE (Mean Absolute Error) is another title for this metric and it is calculated as follows:

$$\mathrm{AAD} = \frac{1}{\mathrm{XY}} \sum_{x,y} |p(x,y) - p'(x,y)|. \tag{3.1}$$

3.3.2 MSE

According to [10], MSE (Mean Squared Error) gives the average squared difference of input and output images per pixel. Power two in this metric formula ensures non-negative result. Greater value of MSE implies more differences between cover and stego images. MSE will result zero when two images are identical. It is calculated as follows:

$$\mathrm{MSE} = \frac{1}{\mathrm{XY}} \sum_{x,y} (p(x,y) - p'(x,y))^2. \tag{3.2}$$

3.3.3 L^p Norm

According to [12], “For a real number $p \geq 1$, the p-norm or L^p-norm of x is defined by

$$\|x\|_p = \left(|x_1|^p + |x_2|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}}. \tag{3.3}$$

The Euclidean norm from the above equation falls into this class and is the 2-norm, and the 1-norm is the norm that corresponds to the Manhattan distance. The L^∞-norm or maximum norm (or uniform norm) is the limit of the L^p-norms for $p \to \infty$. It turns out that this limit is equivalent to the following definition:

$$\|x\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}". \quad (3.4)$$

Referring to [10], the metric which is used in this thesis results in the L^p norm value per pixel due to division by number of pixels:

$$L^p \text{ Norm} = \left(\frac{1}{XY} \sum_{x,y} \left(|p(x,y) - p'(x,y)|^p\right)^{1/p}. \quad (3.5)$$

3.3.4 LMSE

According to [10], LMSE (Laplacian Mean Squared Error) concentrates on difference of cover and stego images per pixel using Laplace operator which implies to difference of each pixel and four main adjacent pixels in each image. LMSE is calculated as follows:

$$\text{LMSE} = \sum_{x,y} \left(L(p(x,y)) - L(p'(x,y))\right)^2 / \sum_{x,y} L(p(x,y))^2 \quad (3.6)$$

where

$$L(p(x,y)) = p(x+1,y) + p(x-1,y) + p(x,y+1) + p(x,y-1) - 4p(x,y). \quad (3.7)$$

3.3.5 SNR

According to [10], SNR (Signal-to-Noise Ratio), as the title expresses, returns the proportion of pixel intensity in cover image (signal) to the difference of color intensities in cover and stego images (noise). Obviously, greater result of the metric is more desired. More similarity of two images results greater value SNR due to tending the difference of two images to zero. For two identical images, SNR is infinity. It is calculated as

$$\text{SNR} = \sum_{x,y} p(x,y)^2 / \sum_{x,y} \left(p(x,y) - p'(x,y)\right)^2. \quad (3.8)$$

SNR can be expressed in decibel unit as follows:

$$\text{SNR (dB)} = 10 \log (\text{SNR}^2). \qquad (3.9)$$

3.3.6 PSNR

PSNR (Peak Signal-to-Noise Ratio) applies the maximal pixel intensity of cover image as intensity of all pixels in cover image (signal) and considers the distortion value of corresponding pixel intensities in cover and stego images as noise. Greater result of PSNR implies to less difference of cover and stego images. For two identical images, PSNR tends to infinity. Referring to [10], PSNR is calculated as

$$\text{PSNR} = XY \max_{x,y} p(x,y)^2 / \sum_{x,y} (p(x,y) - p'(x,y))^2. \qquad (3.10)$$

According to [10], PSNR in decibel unit is calculated as

$$\text{PSNR (dB)} = 10 \log (\text{PSNR}^2). \qquad (3.11)$$

3.3.7 NNC

Referring to [10], NCC (Normalized Cross Correlation) gives the correlation of pixels in two images. It is calculated as

$$\text{NCC} = \sum_{x,y} p(x,y)p'(x,y) / \sum_{x,y} p(x,y)^2. \qquad (3.12)$$

Paying attention to formula (3.12) and results in Table 4 of [1], it is clear that this metric, in opposition to its title, may not result in a normalized value. Due to this, NCC formula, (3.12), has been modified as follows:

$$\text{NCC} = \sum_{x,y} p(x,y)p'(x,y) / (\sqrt{\sum_{x,y} p(x,y)^2} \sqrt{\sum_{x,y} p'(x,y)^2}). \qquad (3.13)$$

According to [10], in formula (3.1) to (3.13), “p(x,y) represents a pixel, whose coordinates are (x,y) in the original, undistorted image and p’(x,y) represents a pixel, whose coordinates are (x,y) in the stego image which contains a secret message”. As shown in metric formulas, each metric evaluates the stego image in comparison to cover image from a distinctive aspect.

Chapter 4

EXPERIMENTS AND RESULTS

4.1 Experimental Setup

To evaluate the new proposed method and prove the claimed privileges, some materials are considered.

Thirteen well-known frequently-used 24-bit BMP images, whose dimensions are 512*512 and the size is 786,486 bytes namely Baboon.bmp, Barbara.bmp, Boats.bmp, Cove.bmp, F16.bmp, Goldhill.bmp, Lena.bmp, Monarch.bmp, Peppers.bmp, Sailboats.bmp, Tulips.bmp, Yacht.bmp, and Zelda.bmp are used. These images are applied to many researches which deal with images. They are all given in Appendix A.

The plain text considered as secret message contains a scientific article, [6], about steganography cut in length of 31,072 bytes.

4.2 Results Descriptions

LSB, SLSB, and ILSB method (Improved LSB), are exerted on the thirteen different cover images (Appendix A) and the hidden message with different number of embedding bits per pixel. The results of comparisons are structured in tables as in Figure 7 as follows:

	ILSB vs. LSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB													
	ILSB													
	ILSB M													
6 b/p	LSB													
	ILSB													
	ILSB M													
9 b/p	LSB													
	ILSB													
	ILSB M													

	ILSB vs. SLSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB													
	ILSB M													
2 b/p	SLSB													
	ILSB M													
3 b/p	SLSB													
	ILSB M													

Figure 11: Results structure

As mentioned before, proposed method is applied in two cases. In the first case, proposed method is used versus simple LSB method in which proposed method is allowed to use all three colors of the pixels. This makes the results be comparable to LSB method. In the second case, proposed method uses only one color out of three to have the results comparable to SLSB method results pixels. According to these cases, the results are tabled into two parts. In the first part or upper half of the Figure 7, results of ILSB (Improved LSB or new proposed method) versus LSB method and in the second part or lower half of the Figure 7, results of ILSB versus SLSB method are tabled. In each part, some metrics that have been explained before are obtained to evaluate the quality of embedment for different number of embedding bits.

In the first (upper) part of the Figure 7, ILSB and LSB methods have been evaluated for one, two, and three bits per color. In other words, for three, six, and nine bits per pixels due to use of three colors per pixel each of them are applied with and without LSB matching (adaptation) concept to prove the effect of this concept for improving the embedment quality.

In the second (lower) part of the Figure 7, ILSB and SLSB methods are evaluated for one, two, and three bits per pixels in only one color, but not in more than one color. The LSB match adaptation concept in this part is applied in ILSB by default due to its usage in SLSB method.

In the first (upper) part of the Figure 7, the expression *LSB* represents for the results of simple LSB method. *ILSB* represents for the results of new proposed method without using adaptation concept and finally, *ILSB M* represents for the results of new proposed method using adaptation concept.

In the second (lower) part of the Figure 7, the expression *SLSB* represents for the results of SLSB method. *ILSB M* represents for the results of new proposed method using adaptation concept.

In the first (upper) part of results, except using one bit, in all other cases, using adaptation concept causes to better quality of embedment. Adaptation concept does not work for one bit per color because

$$|c - c'| = |c - c''|$$

where c is original value of the color (before embedment). c'= c-1 and c''= c+1 are the values which would be generated after applying the matching concept (if least significant bit of the color and secret message bit differ). (c-1) is generated when first least significant bit is converted from zero to one during embedment and matching concept changes second least significant bit with value two from one to zero. The result will be

$$c' = c + 1 – 2$$

and thus,

$$c' = c - 1.$$

(c+1) is generated when first least significant bit is converted from one to zero and matching concept changes second least significant bit with value two from zero to one. The result will be

$$c' = c - 1 + 2$$

and thus,

$$c' = c + 1.$$

Naturally, according to the equation

$$|c – (c-1)| = |c – (c+1)|$$

there is no difference between using and not using the adaptation concept when only first least significant bit of a color participates in embedment.

About metrics, as introduced before, AAD (MAE), MSE, L^p Norm, and LMSE are types of error or distance between cover (original) image and stego image. Thus lower values of these metrics are more desirable. On the other hand, metrics such as SNR, PSNR, and NCC implies to likeness of cover and stego image and consequently, higher values of these metrics would be the reasons of better embedment of secret message and are more desirable.

It is necessary to add that the L^p Norm calculation formula by the author of SLSB article, [1], has been applied incorrectly according to [10].

Referring [1], the incorrect formula of L^p Norm by the author of [1] has been applied as

$$L^p \text{ Norm} = (\frac{1}{XY} \sum_{x,y} (|p(x,y) - p'(x,y)|^p) * (1/p) \qquad (4.1)$$

instead of correct L^p Norm formula, (3.5), from [10] which is

$$L^p \text{ Norm} = (\frac{1}{XY} \sum_{x,y} (|p(x,y) - p'(x,y)|^p)^{1/p} .$$

The incorrect values of L^p Norm by the author of [1] that has been calculated, are stated in the results as “LPNorm O2” and “LPNorm O3” to be comparable with correct L^p-Norm values calculated by (3.5) from [10]. P parameter is considered two and three which are shown in the results tables in this Chapter. The correct value of L^p Norm is shown as “LPNorm N3” with P parameter value of three. Calculation of Lp Norm for P=2 is skipped because of its similarity to MSE metric.

The next important point stated in metrics part, is calculation of NCC metric. As was stated before, NCC is a type of correlation. Thus its value is less than one. But in some calculations, the NCC values have been more than one which are shown in “Old NCC” column of the Table 4.3, Table 4.5, Table 4.10, and Table 4.11.

In order to solve this case, instead of using (3.12) from [10] where

$$\text{NCC} = \sum_{x,y} p(x,y)p'(x,y) / \sum_{x,y} p(x,y)^2,$$

we have changed it into (3.13) as follows:

$$NCC = \sum_{x,y} p(x,y)p'(x,y) \;/ (\sqrt{\sum_{x,y} p(x,y)^2}\sqrt{\sum_{x,y} p'(x,y)^2}).$$

The previous values of NCC are shown with "NCC Old" and the new values of NCC we have used are shown with "NCC New" in the results and all are smaller than or equal to one.

The "Threshold" which has assigned a column to itself in each result table, states the optimal threshold calculated according to cover image, applied method, and embedding bits.

Embedding bit states the average of embedding bit per pixel for the embedding pixels of cover image. According to this explanation, it is clear that average of embedding bit of LSB method for one, two, and three bits per color (three, six, and nine bits per pixel) is three, six, and nine bits per pixel respectively for the embedding pixels. Also, clearly, Average of embedding bit of SLSB method for one, two, and three bits per pixel is one, two, and three bits per pixel respectively for the embedding pixels.

There exist two types of the new proposed method. For the first type of new proposed method which is allowed to apply all three colors of pixels, average of embedding bit varies to cover image and number of embedding bit. But for the second type of new proposed method which uses only one of three colors of the pixels, average of embedding bit of embedding pixels is one, two, and three bits per pixel when message is embedded in one, two, or three bits of a color respectively. It is exactly similar to SLSB method’s embedding bit average. Totally, higher rate of embedding bit will cause to lower scatter of secret message bits and consequently lower quality of embedment.

Percentage of the cover image, as its title states is the percentage of the cover image which secret message takes to be embedded including qualified and non-qualified colors. This metric is considered %100 for SLSB method because it uses the whole cover image to embed secret message regardless of number of embedding bit. It is shown in embedment map of SLSB method in Section 2.3.2 of Chapter two. In other cases, when number of embedding bit increases, the percentage of image which is used decreases because of increase of the capacity of the embedding colors.

The results obtained from the thirteen cover images and one secret message are as follows:

Table 4.1: Results of metrics for ILSB, LSB, and SLSB applying Baboon.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4740524	0.9467392	0.4733696	0.7087682	1.2859038	0.0002034	52.29497	56.73323	0.9998942	0.9999959	-	3	31.64%
	ILSB	0.4718628	0.7287674	0.3643837	0.4394226	1.0964813	0.0002274	53.43139	57.86964	0.9998411	0.9999959	156	1.78	99.09%
	ILSB M	0.4718628	0.7287674	0.3643837	0.4394226	1.0964813	0.0002008	53.43139	57.86964	0.9991355	0.9999963	156	1.78	99.09%
6 b/p	LSB	0.5735817	2.4902611	1.2451305	4.0382932	2.2967111	0.0004642	48.09483	52.53308	0.9998295	0.9999904	-	6	15.82%
	ILSB	0.5694275	1.5450897	0.7725449	1.7144979	1.72618	0.0004712	50.16774	54.60599	0.9996716	0.9999905	184	2.89	82.28%
	ILSB M	0.5216141	1.2818909	0.6409454	1.288854	1.5695494	0.0003819	50.97877	55.41702	0.9994317	0.9999923	184	2.89	82.28%
9 b/p	LSB	0.8182335	7.4668159	3.733408	25.434678	4.2414645	0.0013749	43.32592	47.76417	0.9997917	0.9999719	-	9	10.55%
	ILSB	0.8123131	4.3443336	2.1721668	9.6400108	3.0694903	0.0012985	45.67805	50.1163	0.9997147	0.9999723	192	4.08	67.50%
	ILSB M	0.7209358	3.3655891	1.6827946	6.5629743	2.7002708	0.0010012	46.78667	51.22492	0.9997696	0.9999786	192	4.08	67.50%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4725266	0.4725494	0.2362747	0.1575394	0.778939	0.0002318	55.3128	59.75106	0.9999217	0.9999959	-	1	100.00%
	ILSB M	0.4710693	0.4710693	0.2355347	0.1570231	0.7780872	0.0001982	55.32643	59.76468	0.99948	0.999996	32	1	99.39%
2 b/p	SLSB	0.5228119	0.9092331	0.4546165	0.6105118	1.2235032	0.0004504	52.47052	56.90878	0.9999554	0.9999921	-	2	100.00%
	ILSB M	0.5215263	0.9015083	0.4507542	0.6012561	1.2172886	0.0004068	52.50758	56.94583	0.9995388	0.9999923	104	2	99.03%
3 b/p	SLSB	0.7298889	2.5223465	1.2611732	3.5037486	2.1905411	0.0012456	48.03923	52.47748	0.9998882	0.999978	-	3	100.00%
	ILSB M	0.7206306	2.4549446	1.2274723	3.362662	2.1607349	0.0010552	48.15686	52.59511	0.9996912	0.9999787	160	3	98.54%

In the whole above results, ILSB offers better values of all metrics in comparison to LSB and SLSB methods for various number of embedding bit per pixel

Table 4.2: Results of metrics for ILSB, LSB, and SLSB applying Barbara.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4728088	0.9401932	0.4700966	0.7013194	1.2813832	0.0003439	51.48043	57.31743	0.9998693	0.9999948	-	3	31.64%
	ILSB	0.4728966	0.786541	0.3932705	0.5165876	1.1572355	0.0003803	52.25539	58.09239	0.9998221	0.9999948	140	1.9	97.79%
	ILSB M	0.4728966	0.786541	0.3932705	0.5165876	1.1572355	0.0003381	52.25539	58.09239	0.9990332	0.9999953	140	1.9	97.79%
6 b/p	LSB	0.5718384	2.4813614	1.2406807	4.0284348	2.2948406	0.0007903	47.26571	53.1027	0.9997661	0.999988	-	6	15.82%
	ILSB	0.5722733	1.8423233	0.9211617	2.5068665	1.9592243	0.0008164	48.55895	54.39594	0.9996275	0.999988	168	3.2	68.31%
	ILSB M	0.5232162	1.5277939	0.7638969	1.890775	1.7834203	0.000648	49.37196	55.20895	0.9993426	0.9999903	168	3.2	68.31%
9 b/p	LSB	0.8151932	7.403801	3.7019005	25.099665	4.22276	0.0024349	42.51806	48.35505	0.9997192	0.999965	-	9	10.55%
	ILSB	0.8091011	5.3058815	2.6529408	14.957361	3.5535198	0.0023326	43.96503	49.80202	0.9996171	0.9999655	176	4.69	43.80%
	ILSB M	0.7135429	4.0516243	2.0258121	9.8772672	3.0944682	0.0017079	45.13631	50.97331	0.9995789	0.9999738	176	4.69	43.80%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4719315	0.4719391	0.2359695	0.1573181	0.7785742	0.0003779	54.47375	60.31074	0.9998516	0.9999949	-	1	100.00%
	ILSB M	0.4724503	0.4724503	0.2362251	0.1574834	0.7788468	0.0003246	54.46905	60.30604	0.999435	0.999995	24	1	99.69%
2 b/p	SLSB	0.5254478	0.9161415	0.4580708	0.6170133	1.227831	0.0007623	51.59298	57.42997	0.9999277	0.99999	-	2	100.00%
	ILSB M	0.5220528	0.9031258	0.4515629	0.6033618	1.2187081	0.0006526	51.65512	57.49212	0.9995022	0.9999902	96	2	95.67%
3 b/p	SLSB	0.7353973	2.5613327	1.2806664	3.5879211	2.207944	0.0021154	47.12795	52.96494	0.9998321	0.999972	-	3	100.00%
	ILSB M	0.7203827	2.4458847	1.2229424	3.3317108	2.1540851	0.0016291	47.32825	53.16524	0.9996204	0.9999733	128	3	96.79%

In the results of Table 4.2, ILSB offers worse value of AAD versus LSB for three bits per pixel and also, worse values of AAD, MSE, and L^P-norm in comparison to SLSB method for one bit per pixel. In all other cases, ILSB offers better results versus LSB and SLSB.

Table 4.3: Results of metrics for ILSB, LSB, and SLSB applying Boats.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4741058	0.9452209	0.4726105	0.706871	1.2847555	0.0017517	52.30967	56.59597	0.9998574	0.9999957	-	3	31.64%
	ILSB	0.4730759	0.7767944	0.3883972	0.5020905	1.1463073	0.0019275	53.16194	57.44825	0.9998767	0.9999957	156	1.89	91.95%
	ILSB M	0.4730759	0.7767944	0.3883972	0.5020905	1.1463073	0.0016867	53.16194	57.44825	0.9992362	0.9999959	156	1.89	91.95%
6 b/p	LSB	0.5695724	2.466713	1.2333565	3.9992828	2.2892917	0.0040736	48.14382	52.43012	0.9997343	0.99999	-	6	15.82%
	ILSB	0.5642166	1.7476578	0.8738289	2.2275391	1.8835719	0.0040386	49.64044	53.92674	0.9998463	0.9999901	160	3.35	46.04%
	ILSB M	0.5134583	1.4301071	0.7150536	1.6387431	1.7003724	0.0030976	50.51132	54.79762	0.9996671	0.999992	160	3.35	46.04%
9 b/p	LSB	0.8206863	7.5158234	3.7579117	25.696414	4.2559638	0.0130601	43.30524	47.59154	0.9996724	0.9999702	-	9	10.55%
	ILSB	0.7898865	4.7373886	2.3686943	11.919126	3.2944927	0.0099664	45.30962	49.59592	1.0001443	0.9999723	160	4.67	34.43%
	ILSB M	0.7188263	3.8980331	1.9490166	8.9199142	2.9910751	0.00833	46.15655	50.44285	1.000219	0.9999775	160	4.67	34.43%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4759979	0.4759979	0.237999	0.158666	0.7807914	0.0019301	55.28895	59.57526	0.9998702	0.9999956	-	1	100.00%
	ILSB M	0.4727936	0.4727936	0.2363968	0.1575979	0.7790354	0.0016036	55.31829	59.60459	0.9994345	0.9999958	12	1	98.45%
2 b/p	SLSB	0.528595	0.9248047	0.4624023	0.6250432	1.2331345	0.0038439	52.4045	56.69081	0.9999285	0.9999915	-	2	100.00%
	ILSB M	0.518383	0.8931541	0.4465771	0.5939445	1.2123343	0.0031479	52.55574	56.84204	0.9995092	0.9999919	160	2	97.69%
3 b/p	SLSB	0.7545395	2.7125473	1.3562737	3.9187113	2.2738135	0.0110398	47.73123	52.01753	0.9998362	0.9999751	-	3	100.00%
	ILSB M	0.7173729	2.4333534	1.2166767	3.3207283	2.1517156	0.0072779	48.20295	52.48926	0.9997558	0.9999777	144	3	36.31%

In the whole above results, proposed method (ILSB) offers better values of all metrics versus LSB and SLSB for various number of embedding bit per pixel.

Table 4.4: Results of metrics for ILSB, LSB, and SLSB applying Cove.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4721184	0.9409676	0.4704838	0.7039579	1.2829881	0.0021191	54.61885	57.93748	0.999857	0.9999974	-	3	31.64%
	ILSB	0.4738846	0.9147568	0.4573784	0.667544	1.2604735	0.0024932	54.74154	58.06017	0.9998598	0.9999974	237	2.77	91.95%
	ILSB M	0.4738846	0.9147568	0.4573784	0.667544	1.2604735	0.0020689	54.74154	58.06017	0.9992492	0.9999977	237	2.77	91.95%
6 b/p	LSB	0.6040993	2.7573814	1.3786907	4.7082176	2.4172743	0.0052958	49.94963	53.26826	0.9994562	0.9999935	-	6	15.82%
	ILSB	0.6020584	2.699295	1.3496475	4.5722453	2.3937765	0.0062334	50.04209	53.36073	0.9994388	0.9999936	241	5.73	46.04%
	ILSB M	0.5558777	2.3106689	1.1553345	3.6762594	2.2259179	0.0053722	50.71722	54.03585	0.9992363	0.9999947	241	5.73	46.04%
9 b/p	LSB	0.9229431	9.5795288	4.7897644	37.402306	4.8232511	0.0163027	44.54115	47.85979	0.9987465	0.9999788	-	9	10.55%
	ILSB	0.9105301	9.249321	4.6246605	35.731336	4.750327	0.0173663	44.6935	48.01213	0.9985928	0.9999795	241	8.66	34.43%
	ILSB M	0.8460464	8.0837021	4.041851	29.941958	4.4785128	0.0161445	45.27849	48.59713	0.9986584	0.9999823	241	8.66	34.43%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4725075	0.4725075	0.2362537	0.1575025	0.7788782	0.0020704	57.61051	60.92914	0.9999286	0.9999974	-	1	100.00%
	ILSB M	0.4738235	0.4738235	0.2369118	0.1579412	0.7796007	0.0016638	57.59843	60.91706	0.9995169	0.9999975	5	1	96.16%
2 b/p	SLSB	0.5254517	0.9143143	0.4571571	0.6144206	1.2261088	0.0042149	54.74364	58.06227	0.9999492	0.999995	-	2	100.00%
	ILSB M	0.5321884	0.9302826	0.4651413	0.6265691	1.2341371	0.0031503	54.66845	57.98708	0.9994575	0.999995	185	2	74.02%
3 b/p	SLSB	0.7335854	2.5485344	1.2742672	3.5631243	2.2028457	0.011061	50.29169	53.61032	0.9998652	0.999986	-	3	100.00%
	ILSB M	0.7430344	2.6214752	1.3107376	3.7453435	2.2397745	0.0078958	50.16914	53.48777	0.999441	0.9999857	225	3	40.48%

In the first (upper) part of the results, proposed method offers greater value of AAD versus LSB method for three bits per pixel. But in all the other cases of this part, ILSB offer better values versus LSB for various number of embedding bit per pixel. In the second part, almost all values of the metrics that ILSB offers are worse than the values that SLSB offers.

Table 4.5: Results of metrics for ILSB, LSB, and SLSB applying F16.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4730949	0.9442635	0.4721317	0.7063815	1.2844588	0.0019178	55.2077	56.9882	0.9999109	0.9999978	-	3	31.64%
	ILSB	0.4706917	0.8627815	0.4313908	0.6064059	1.2207542	0.0020991	55.59963	57.38013	0.9999492	0.9999978	208	2.45	96.53%
	ILSB M	0.4706917	0.8627815	0.4313908	0.6064059	1.2207542	0.0018217	55.59963	57.38013	0.9995502	0.9999979	208	2.45	96.53%
6 b/p	LSB	0.5736237	2.497345	1.2486725	4.0680237	2.3023336	0.0044508	50.98385	52.76435	0.9998282	0.9999947	-	6	15.82%
	ILSB	0.5661011	2.1610413	1.0805206	3.2345759	2.1329444	0.0049562	51.612	53.3925	0.9999526	0.9999948	208	4.62	55.02%
	ILSB M	0.5174637	1.7865906	0.8932953	2.4141337	1.9347619	0.0038407	52.43838	54.21888	0.9998382	0.9999958	208	4.62	55.02%
9 b/p	LSB	0.8192863	7.475956	3.737978	25.443567	4.2419585	0.0144203	46.22197	48.00247	0.9998004	0.9999846	-	9	10.55%
	ILSB	0.7977905	5.9841003	2.9920502	17.909294	3.7734034	0.0135591	47.18864	48.96915	1.0002541	0.9999853	208	6.38	40.45%
	ILSB M	0.7069626	4.4745483	2.2372742	10.984701	3.2060466	0.0105671	48.45114	50.23164	1.0003386	0.9999888	208	6.38	40.45%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4722519	0.4722519	0.2361259	0.1574173	0.7787378	0.0020958	58.2169	59.9974	0.9999195	0.9999978	-	1	100.00%
	ILSB M	0.4726982	0.4726982	0.2363491	0.1575661	0.778983	0.0017631	58.21279	59.9933	0.9995563	0.9999978	80	1	98.94%
2 b/p	SLSB	0.5280838	0.9260254	0.4630127	0.6270091	1.2344259	0.0039995	55.2924	57.07291	0.9999655	0.9999956	-	2	100.00%
	ILSB M	0.5174561	0.8879776	0.4439888	0.5883408	1.2085095	0.003249	55.47461	57.25511	0.999731	0.9999958	200	2	80.83%
3 b/p	SLSB	0.7566872	2.7271996	1.3635998	3.9502042	2.2798885	0.0121087	50.60146	52.38197	0.9999314	0.999987	-	3	100.00%
	ILSB M	0.6872101	2.205719	1.1028595	2.8184484	2.0372474	0.0068124	51.52313	53.30363	1.0001783	0.9999895	208	3	88.86%

In the first part of the Table 4.5,proposed method offers better values of the all metrics versus LSB. But for one-bit-per-pixel embedment, AAD, MSE, L^P-norm, SNR, and PSNR are worse than the corresponding values of SLSB.

Table 4.6: Results of metrics for ILSB, LSB, and SLSB applying Goldhill.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.474102	0.9445381	0.4722691	0.7053286	1.2838203	0.0012492	51.4494	57.23619	0.9998291	0.9999947	-	3	31.64%
	ILSB	0.4745178	0.8423843	0.4211922	0.5782242	1.2015426	0.0013315	51.94649	57.73329	0.9998104	0.9999947	124	2.25	81.96%
	ILSB M	0.4745178	0.8423843	0.4211922	0.5782242	1.2015426	0.0011585	51.94649	57.73329	0.9989864	0.9999952	124	2.25	81.96%
6 b/p	LSB	0.5680466	2.451683	1.2258415	3.9593074	2.2816384	0.0026227	47.30695	53.09375	0.9995901	0.9999879	-	6	15.82%
	ILSB	0.5670967	2.1556664	1.0778332	3.2291094	2.1317422	0.0027429	47.86577	53.65257	0.999587	0.9999879	160	4.47	74.11%
	ILSB M	0.5176735	1.7820549	0.8910275	2.4156761	1.9351738	0.0022335	48.69238	54.47918	0.9993272	0.9999902	160	4.47	74.11%
9 b/p	LSB	0.8100777	7.3343163	3.6671581	24.870818	4.209887	0.008402	42.54799	48.33479	0.9997441	0.9999645	-	9	10.55%
	ILSB	0.8058548	6.4386597	3.2193298	20.305133	3.9346765	0.0081616	43.11364	48.90043	0.9997992	0.9999648	176	6.75	64.40%
	ILSB M	0.7024078	4.7871704	2.3935852	12.777944	3.3717918	0.0062658	44.4008	50.1876	0.9994965	0.9999741	176	6.75	64.40%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4755363	0.475544	0.237772	0.1585197	0.7805515	0.0013771	54.42968	60.21648	0.9998806	0.9999947	-	1	100.00%
	ILSB M	0.4736557	0.4736557	0.2368279	0.1578852	0.7795086	0.0011489	54.44696	60.23376	0.999351	0.9999949	32	1	98.64%
2 b/p	SLSB	0.5280113	0.9225655	0.4612827	0.6227735	1.23164	0.0027262	51.55162	57.33842	0.9999385	0.9999897	-	2	100.00%
	ILSB M	0.5193825	0.8954201	0.44771	0.5961164	1.2138102	0.0023087	51.68132	57.46812	0.9994461	0.9999901	104	2	94.99%
3 b/p	SLSB	0.7428246	2.6245613	1.3122807	3.7294579	2.2366034	0.0076017	47.01102	52.79782	0.999856	0.9999706	-	3	100.00%
	ILSB M	0.710247	2.3720512	1.1860256	3.1877454	2.1226007	0.0053918	47.45035	53.23715	0.9995425	0.9999735	112	3	70.26%

In the results of Goldhill.bmp, ILSB offers worse value only for AAD in comparison to LSB and for three bits per pixel. In all other cases, ILSB offers better results in comparison to LSB and SLSB.

Table 4.7: Results of metrics for ILSB, LSB, and SLSB applying Lena.bmp

	ILSB vs. LSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4753304	0.9518929	0.4759464	0.714489	1.2893543	0.0019147	52.37424	57.36077	0.9998633	0.999996	-	3	31.64%
	ILSB	0.4757042	0.64674	0.32337	0.353934	1.0201902	0.0021109	54.05282	59.03935	0.9998064	0.999996	156	1.35	96.39%
	ILSB M	0.4757042	0.64674	0.32337	0.353934	1.0201902	0.0018188	54.05282	59.03935	0.999123	0.9999964	156	1.35	96.39%
6 b/p	LSB	0.5711288	2.4714584	1.2357292	3.9983063	2.2891053	0.0043314	48.23059	53.21712	0.9997726	0.9999908	-	6	15.82%
	ILSB	0.5673904	1.2072601	0.6036301	1.0423101	1.4623098	0.00423	51.34211	56.32864	0.9996812	0.9999909	200	2.18	89.40%
	ILSB M	0.5190353	0.9944458	0.4972229	0.7742462	1.3243419	0.0033302	52.18431	57.17084	0.9994398	0.9999927	200	2.18	89.40%
9 b/p	LSB	0.8185349	7.4514694	3.7257347	25.289439	4.2333757	0.0130505	43.4377	48.42423	0.9997414	0.999973	-	9	10.55%
	ILSB	0.8079567	3.315815	1.6579075	5.5361722	2.5513906	0.0104151	46.95422	51.94075	0.9998012	0.9999736	208	3.12	73.73%
	ILSB M	0.7288628	2.6521034	1.3260517	3.9700203	2.2836944	0.0084226	47.92422	52.91075	0.9997673	0.9999789	208	3.12	73.73%

	ILSB vs. SLSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4754982	0.4754982	0.2377491	0.1584994	0.7805181	0.0022184	55.38863	60.37516	0.9999025	0.999996	-	1	100.00%
	ILSB M	0.4738464	0.4738464	0.2369232	0.1579488	0.7796132	0.001897	55.40374	60.39027	0.9995242	0.9999961	60	1	98.94%
2 b/p	SLSB	0.5283127	0.9250031	0.4625015	0.6259715	1.2337446	0.0044288	52.49869	57.48522	0.9999411	0.9999923	-	2	100.00%
	ILSB M	0.5205841	0.8991623	0.4495811	0.5998205	1.2163191	0.0038225	52.62174	57.60827	0.9996531	0.9999925	96	2	83.80%
3 b/p	SLSB	0.7453995	2.6402512	1.3201256	3.7543869	2.2415758	0.0122447	47.94367	52.9302	0.9998807	0.9999779	-	3	100.00%
	ILSB M	0.7176132	2.4263382	1.2131691	3.2929535	2.1456997	0.0097243	48.31061	53.29714	0.9997618	0.9999797	112	3	77.92%

In the results of Lena.bmp, ILSB offers worse (Greater) value only for AAD in comparison to LSB and for three bits per pixel. In all other cases, ILSB offers better results in comparison to LSB and SLSB.

Table 4.8: Results of metrics for ILSB, LSB, and SLSB applying Monarch.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4705276	0.9385834	0.4692917	0.7039235	1.2829673	0.0009642	50.49602	57.48203	0.9999421	0.9999939	-	3	31.64%
	ILSB	0.4738846	0.6865921	0.3432961	0.3930003	1.0564235	0.0010689	51.85376	58.83977	0.9998883	0.9999938	112	1.55	95.51%
	ILSB M	0.4738846	0.6865921	0.3432961	0.3930003	1.0564235	0.0009362	51.85376	58.83977	0.9989156	0.9999944	112	1.55	95.51%
6 b/p	LSB	0.5668907	2.4395485	1.2197742	3.9304644	2.2760845	0.0024017	46.34765	53.33367	0.9998076	0.9999859	-	6	15.82%
	ILSB	0.5775833	1.7865067	0.8932533	2.2821579	1.8988427	0.0023513	47.7007	54.68671	0.9995059	0.9999855	152	3.24	75.22%
	ILSB M	0.5241241	1.4491806	0.7245903	1.6506678	1.7044868	0.0018903	48.60952	55.59554	0.9992947	0.9999885	152	3.24	75.22%
9 b/p	LSB	0.7973289	7.0769615	3.5384808	23.45642	4.1285194	0.0072067	41.72228	48.70829	0.9997688	0.99996	-	9	10.55%
	ILSB	0.8035774	4.9823647	2.4911823	12.76808	3.3709239	0.0069724	43.24639	50.23241	0.9998758	0.9999593	176	4.86	68.87%
	ILSB M	0.7330742	4.1225777	2.0612888	9.5977058	3.0649935	0.0056552	44.06906	51.05507	0.9997552	0.9999667	176	4.86	68.87%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4706802	0.4706802	0.2353401	0.1568934	0.7778729	0.0010335	53.49349	60.4795	0.9999659	0.9999939	-	1	100.00%
	ILSB M	0.4697838	0.4697838	0.2348919	0.1565946	0.7773788	0.0008905	53.50177	60.48778	0.999538	0.999994	24	1	97.28%
2 b/p	SLSB	0.5290337	0.9330177	0.4665089	0.6346118	1.2393951	0.0021377	50.52185	57.50786	0.9999598	0.9999878	-	2	100.00%
	ILSB M	0.5189095	0.8911552	0.4455776	0.5906868	1.2101137	0.0016792	50.72121	57.70723	0.999516	0.9999885	80	2	90.89%
3 b/p	SLSB	0.7529602	2.6979294	1.3489647	3.8741557	2.2651629	0.006215	45.91044	52.89645	0.9999211	0.9999649	-	3	100.00%
	ILSB M	0.713623	2.3950195	1.1975098	3.2246068	2.1307509	0.0044585	46.42766	53.41367	0.9997123	0.9999688	64	3	75.16%

In the results of Table 4.8, ILSB offers worse value only for AAD in comparison to LSB and for three bits per pixel. In all other cases, ILSB offers better results in comparison to LSB and SLSB.

Table 4.9: Results of metrics for ILSB, LSB, and SLSB applying Peppers.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4713478	0.9418831	0.4709415	0.7062263	1.2843648	0.0011819	51.35999	56.82033	0.9998843	0.9999953	-	3	31.64%
	ILSB	0.4727707	0.6375656	0.3187828	0.3348745	1.0015388	0.0013719	53.05471	58.51505	0.9998278	0.9999953	156	1.43	98.10%
	ILSB M	0.4727707	0.6375656	0.3187828	0.3348745	1.0015388	0.001185	53.05471	58.51505	0.9990434	0.9999957	156	1.43	98.10%
6 b/p	LSB	0.5739822	2.4942856	1.2471428	4.0525792	2.2994162	0.0027609	47.13049	52.59084	0.9997676	0.9999889	-	6	15.82%
	ILSB	0.5693779	1.410778	0.705389	1.4857445	1.6457169	0.0029456	49.60537	55.06571	0.9996828	0.9999891	184	2.48	72.50%
	ILSB M	0.521183	1.1681938	0.5840969	1.1179949	1.4968802	0.002334	50.42481	55.88515	0.999432	0.9999911	184	2.48	72.50%
9 b/p	LSB	0.8178596	7.4619865	3.7309933	25.422667	4.2407967	0.0079699	42.37141	47.83176	0.9997083	0.9999676	-	9	10.55%
	ILSB	0.8110542	3.9338646	1.9669323	8.3023796	2.9203929	0.0077127	45.15176	50.61211	0.9997791	0.9999681	192	3.51	53.70%
	ILSB M	0.7198296	3.0520134	1.5260067	5.703879	2.5768978	0.0059862	46.25409	51.71444	0.9997663	0.9999754	192	3.51	53.70%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4748459	0.4748459	0.2374229	0.158282	0.780161	0.001352	54.33443	59.79477	0.9998806	0.9999952	-	1	100.00%
	ILSB M	0.4696121	0.4696121	0.2348061	0.1565374	0.7772841	0.0011123	54.38256	59.84291	0.9996726	0.9999953	0	1	94.82%
2 b/p	SLSB	0.530304	0.9312744	0.4656372	0.6308695	1.2369542	0.0026554	51.40918	56.86952	0.9999414	0.9999907	-	2	100.00%
	ILSB M	0.5234566	0.906086	0.453043	0.6060727	1.2205306	0.0023899	51.52826	56.98861	0.999421	0.9999911	152	2	95.26%
3 b/p	SLSB	0.752224	2.7013893	1.3506947	3.9004606	2.270278	0.0075528	46.78408	52.24443	0.9998641	0.9999729	-	3	100.00%
	ILSB M	0.7155266	2.4127464	1.2063732	3.2630908	2.1391938	0.0058837	47.27484	52.73518	0.9998223	0.9999758	80	3	94.40%

In the Table 4.9, ILSB offers worse (greater) value for AAD and LMSE in comparison to LSB and for three bits per pixel. But for all other cases, ILSB offers better results versus LSB and SLSB.

Table 4.10: Results of metrics for ILSB, LSB, and SLSB applying Sailboats.bmp

	ILSB vs. LSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4746666	0.9496727	0.4748363	0.7121468	1.2879438	0.0006758	52.61874	57.13862	0.9998886	0.999996	-	3	31.64%
	ILSB	0.4697533	0.8002586	0.4001293	0.5234057	1.1623045	0.0007552	53.36217	57.88206	0.9998431	0.9999961	176	2.25	98.19%
	ILSB M	0.4697533	0.8002586	0.4001293	0.5234057	1.1623045	0.0006505	53.36217	57.88206	0.999153	0.9999964	176	2.25	98.19%
6 b/p	LSB	0.571888	2.4829597	1.2414799	4.0323575	2.2955853	0.0015076	48.44478	52.96466	0.9997819	0.9999908	-	6	15.82%
	ILSB	0.5723343	2.0615921	1.0307961	2.9333013	2.0645527	0.0016472	49.25245	53.77233	0.9997164	0.9999908	200	4.26	46.97%
	ILSB M	0.5203094	1.6849747	0.8424873	2.1490072	1.8611714	0.0012813	50.12854	54.64843	0.9994452	0.9999926	200	4.26	46.97%
9 b/p	LSB	0.8265305	7.5969391	3.7984695	26.028051	4.2741947	0.0046191	43.58809	48.10797	0.999693	0.9999725	-	9	10.55%
	ILSB	0.7972908	5.8029671	2.9014835	16.831506	3.6961371	0.0044785	44.75797	49.27786	0.9999318	0.9999742	208	6.25	43.89%
	ILSB M	0.6958046	4.2377968	2.1188984	10.031312	3.1104723	0.0034404	46.12307	50.64296	1.0001053	0.999981	208	6.25	43.89%

	ILSB vs. SLSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4731903	0.4731903	0.2365952	0.1577301	0.8293903	0.0007657	55.64412	60.164	0.9999072	0.999996	-	1	100.00%
	ILSB M	0.4725494	0.4725494	0.2362747	0.1575165	0.7789013	0.0006636	55.65	60.16989	0.9994283	0.9999962	68	1	98.74%
2 b/p	SLSB	0.527401	0.920002	0.460001	0.6194496	1.4310983	0.0015095	52.75659	57.27647	0.9999824	0.9999923	-	2	100.00%
	ILSB M	0.5180893	0.8889847	0.4444923	0.5886434	1.2087167	0.00123	52.90553	57.42542	0.9994665	0.9999927	104	2	99.76%
3 b/p	SLSB	0.7401543	2.6000633	1.3000317	3.6726672	2.699136	0.0042122	48.24464	52.76452	0.9999349	0.9999782	-	3	100.00%
	ILSB M	0.7062798	2.3351402	1.1675701	3.0905444	2.1008034	0.0031146	48.71135	53.23123	0.9997135	0.9999804	192	3	96.19%

In the Table 4.10, ILSB offers better values of all metrics versus LSB and SLSB.

Table 4.11: Results of metrics for ILSB, LSB, and SLSB applying Tulips.bmp

	ILSB vs. LSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4726791	0.9440002	0.4720001	0.7071966	1.2849527	0.0019662	51.69562	57.46901	0.9999454	0.9999954	-	3	31.64%
	ILSB	0.4715729	0.7524185	0.3762093	0.4723689	1.1232269	0.0022067	52.68075	58.45414	0.9999058	0.9999954	140	1.83	92.98%
	ILSB M	0.4715729	0.7524185	0.3762093	0.4723689	1.1232269	0.0019153	52.68075	58.45414	0.998988	0.9999959	140	1.83	92.98%
6 b/p	LSB	0.5670815	2.446003	1.2230015	3.9560712	2.2810166	0.0048245	47.56077	53.33416	0.9997698	0.9999894	-	6	15.82%
	ILSB	0.5724754	1.6098747	0.8049374	1.9375369	1.798003	0.0047536	49.37742	55.15081	0.9994239	0.9999894	208	2.79	96.76%
	ILSB M	0.5193367	1.313015	0.6565075	1.4244703	1.6227746	0.0037536	50.26265	56.03604	0.99928	0.9999916	208	2.79	96.76%
9 b/p	LSB	0.8084869	7.2850952	3.6425476	24.507975	4.1893138	0.0155095	42.82099	48.59438	0.9997792	0.9999692	-	9	10.55%
	ILSB	0.8045731	4.2938995	2.1469498	10.136607	3.1213175	0.0116672	45.11682	50.89021	0.9994076	0.9999697	224	3.79	94.05%
	ILSB M	0.683815	2.9572449	1.4786224	5.5787226	2.5579105	0.0089598	46.73647	52.50986	1.0000367	0.999979	224	3.79	94.05%

	ILSB vs. SLSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.472847	0.472847	0.2364235	0.1576157	0.8292398	0.0021201	54.69814	60.47153	0.9999336	0.9999954	-	1	100.00%
	ILSB M	0.4707146	0.4707146	0.2353573	0.1569049	0.7778918	0.0018197	54.71777	60.49116	0.9994868	0.9999955	20	1	99.28%
2 b/p	SLSB	0.526722	0.9230614	0.4615307	0.6236102	1.4342576	0.0042872	51.79304	57.56643	0.999921	0.999991	-	2	100.00%
	ILSB M	0.5172768	0.8876305	0.4438152	0.5878309	1.2081603	0.0036246	51.96302	57.73641	0.9995252	0.9999914	112	2	94.82%
3 b/p	SLSB	0.7544594	2.7117653	1.3558826	3.9130313	2.7543096	0.012015	47.11282	52.88621	0.9998936	0.9999735	-	3	100.00%
	ILSB M	0.7141266	2.3994446	1.1997223	3.2317861	2.132331	0.0091927	47.64423	53.41763	0.9996728	0.9999766	112	3	80.61%

All values of the metrics obtained by applying Tulips.bmp as cover image are better for ILSB versus LSB and SLSB.

Table 4.12: Results of metrics for ILSB, LSB, and SLSB applying Yacht.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4738808	0.9469261	0.4734631	0.7094358	1.2863074	0.0016598	51.4245	57.47947	0.9998889	0.9999948	-	3	31.64%
	ILSB	0.4748497	0.8449898	0.4224949	0.5847359	1.2060362	0.0018427	51.91915	57.97411	0.99981	0.9999948	144	2.18	97.29%
	ILSB M	0.4748497	0.8449898	0.4224949	0.5847359	1.2060362	0.0015962	51.91915	57.97411	0.9989757	0.9999953	144	2.18	97.29%
6 b/p	LSB	0.5704842	2.4653282	1.2326641	3.9811007	2.2858171	0.0040953	47.26891	53.32388	0.9997739	0.999988	-	6	15.82%
	ILSB	0.5684776	2.0596123	1.0298061	3.0199979	2.0846955	0.004409	48.04981	54.10477	0.9996191	0.9999881	176	3.97	60.41%
	ILSB M	0.5196571	1.7078514	0.8539257	2.2696012	1.8953537	0.0034193	48.86316	54.91813	0.9993731	0.9999904	176	3.97	60.41%
9 b/p	LSB	0.818985	7.4763641	3.7381821	25.474762	4.2436914	0.0128679	42.45076	48.50572	0.9997449	0.9999646	-	9	10.55%
	ILSB	0.8107338	6.2931404	3.1465702	19.818714	3.9030032	0.0134638	43.19899	49.25395	0.9997078	0.9999652	176	6.03	43.45%
	ILSB M	0.7220573	4.9493446	2.4746723	13.751718	3.4553555	0.0108986	44.24218	50.29715	0.9996543	0.999973	176	6.03	43.45%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4747124	0.47472	0.23736	0.1582451	0.7801004	0.0018356	54.42329	60.47825	0.9998871	0.9999948	-	1	100.00%
	ILSB M	0.4736824	0.4736824	0.2368412	0.1578941	0.7795233	0.0015332	54.43279	60.48776	0.9992784	0.9999951	24	1	99.88%
2 b/p	SLSB	0.5299187	0.9326591	0.4663296	0.6340446	1.2390258	0.0036982	51.49043	57.5454	0.9999514	0.9999898	-	2	100.00%
	ILSB M	0.5200462	0.8938332	0.4469166	0.5924466	1.2113143	0.0028758	51.6751	57.73006	0.999391	0.9999904	120	2	97.23%
3 b/p	SLSB	0.753315	2.7108192	1.3554096	3.9196688	2.2739987	0.0104971	46.85666	52.91162	0.9998825	0.9999702	-	3	100.00%
	ILSB M	0.7072906	2.3521347	1.1760674	3.1390839	2.1117446	0.0068075	47.47304	53.52801	0.999584	0.9999743	160	3	84.23%

In the results of Table 4.11, ILSB offers better values of metrics versus LSB and SLSB except in AAD versus LSB (upper part) for three bits per pixel.

Table 4.13: Results of metrics for ILSB, LSB, and SLSB applying Zelda.bmp

ILSB vs. LSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
3 b/p	LSB	0.4725418	0.9431229	0.4715614	0.7058843	1.2841574	0.0054213	50.92261	56.45897	0.9998807	0.9999943	-	3	31.64%
	ILSB	0.472847	0.7306366	0.3653183	0.4376678	1.0950198	0.0061852	52.03127	57.56764	0.9998074	0.9999943	132	1.83	93.41%
	ILSB M	0.472847	0.7306366	0.3653183	0.4376678	1.0950198	0.005301	52.03127	57.56764	0.9989612	0.9999948	132	1.83	93.41%
6 b/p	LSB	0.5718422	2.475132	1.237566	4.0037015	2.2901345	0.0131455	46.73231	52.26867	0.9997388	0.9999867	-	6	15.82%
	ILSB	0.5703697	1.5579567	0.7789783	1.7584521	1.7408069	0.0131105	48.74274	54.2791	0.9996223	0.9999867	160	2.87	92.66%
	ILSB M	0.5214119	1.2865334	0.6432667	1.3124301	1.5790619	0.010487	49.57408	55.11044	0.9993331	0.9999893	160	2.87	92.66%
9 b/p	LSB	0.8158569	7.4128952	3.7064476	25.111163	4.2234046	0.0430685	41.96841	47.50477	0.9996937	0.999961	-	9	10.55%
	ILSB	0.8085785	4.7567749	2.3783875	11.588191	3.2637155	0.037698	43.89516	49.43152	0.999655	0.9999616	160	4.56	65.75%
	ILSB M	0.7169342	3.6723938	1.8361969	7.8054606	2.8609258	0.0281195	45.0188	50.55516	0.9996312	0.9999706	160	4.56	65.75%

ILSB vs. SLSB		AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC	Threshold	Embedding Bit (b/p)	% Image
1 b/p	SLSB	0.4745369	0.4745445	0.2372723	0.1581866	0.7800043	0.0060499	53.90552	59.44188	0.999903	0.9999942	-	1	100.00%
	ILSB M	0.4731445	0.4731445	0.2365723	0.1577148	0.7792281	0.0050472	53.91835	59.45471	0.9994758	0.9999944	8	1	96.29%
2 b/p	SLSB	0.5289421	0.9304314	0.4652157	0.6323916	1.2379481	0.0119826	50.98145	56.51781	0.9999474	0.9999887	-	2	100.00%
	ILSB M	0.5185776	0.8965149	0.4482574	0.5983047	1.2152937	0.0098948	51.14271	56.67908	0.9995648	0.9999892	80	2	84.32%
3 b/p	SLSB	0.7639084	2.7929001	1.39645	4.1007665	2.3084941	0.0341739	46.20774	51.7441	0.999852	0.9999661	-	3	100.00%
	ILSB M	0.717041	2.4250717	1.2125359	3.2956161	2.1462779	0.0241558	46.82104	52.3574	0.9995744	0.9999706	128	3	71.04%

In the Table 4.13, ILSB offers better values of metrics versus LSB and SLSB except in AAD versus LSB (upper part) for three bits per pixel.

Table 4.14: Average values of metrics for ILSB, LSB, and SLSB

	ILSB vs. LSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC
3 b/p	LSB	0.4731736	0.9444618	0.4722309	0.7070715	1.2848736	0.0016438	52.173288	57.155208	0.9998855	0.9999955
	ILSB	0.4729471	0.7700944	0.3850472	0.4930971	1.1344257	0.0018462	53.083924	58.065845	0.9998499	0.9999955
	ILSB M	0.4729471	0.7700944	0.3850472	0.4930971	1.1344257	0.0015906	53.083924	58.065845	0.9991039	0.9999959
6 b/p	LSB	0.5733892	2.4938046	1.2469023	4.0581646	2.2999422	0.0039049	47.958484	52.940405	0.9997397	0.9999896
	ILSB	0.5722448	1.8342042	0.9171021	2.4572565	1.9171051	0.0040543	49.381353	54.363272	0.9996443	0.9999896
	ILSB M	0.5226432	1.517177	0.7585885	1.8479122	1.7410205	0.0032361	50.212085	55.194005	0.9994185	0.9999917
9 b/p	LSB	0.8238464	7.5798425	3.7899213	26.095225	4.2714293	0.0123298	43.139998	48.121918	0.9996618	0.9999695
	ILSB	0.8130185	5.3414239	2.670712	15.034147	3.4771377	0.0111609	44.789984	49.771905	0.9997139	0.9999701
	ILSB M	0.7237769	4.1772417	2.0886208	10.423352	3.0578781	0.0088845	45.890604	50.872526	0.9997521	0.9999769

	ILSB vs. SLSB	AAD	MSE	LPNorm O2	LPNorm O3	LPNorm N3	LMSE	SNR	PSNR	Old NCC	New NCC
1 b/p	SLSB	0.4736202	0.4736243	0.2368121	0.1578781	0.7872122	0.0018045	55.170785	60.152705	0.999904	0.9999955
	ILSB M	0.4722941	0.4722941	0.2361471	0.1574314	0.7787602	0.0015127	55.182995	60.164916	0.9994752	0.9999957
2 b/p	SLSB	0.5276181	0.9237334	0.4618667	0.62444	1.2637744	0.003592	52.269761	57.251682	0.9999469	0.9999913
	ILSB M	0.5206099	0.8980643	0.4490321	0.5979534	1.2150182	0.0029563	52.392338	57.37426	0.9995171	0.9999916
3 b/p	SLSB	0.7473341	2.6578184	1.3289092	3.7991003	2.3234301	0.0101602	47.681741	52.663661	0.9998798	0.9999748
	ILSB M	0.7146445	2.4061018	1.2030509	3.2541785	2.1363815	0.0071846	48.114881	53.096802	0.9996977	0.9999773

In above Table, According to the above results, ILSB offers better values in comparison to LSB and SLSB methods in all cases for the all number of embedding bit per pixel.

To be more detailed, as the Table 4.1 states, proposed method (ILSB) offers better values of all metrics in comparison to LSB and SLSB methods for various number of embedding bit per pixel. In other words, ILSB offers smaller values of AAD, MSE, L^P-norm, and LMSE in comparison to LSB and SLSB that imply less difference between cover image and stego image. On the other hand, ILSB offers greater values of SNR, PSNR, and NCC in comparison to LSB and SLSB that imply more similarity between cover image and stego image for different number of embedding bit per pixel.

In Table 4.2, ILSB offers worse value of AAD versus LSB method for three bits per pixel and also, worse values of AAD, MSE, and L^P-norm in comparison to SLSB method for one bit per pixel. In all other cases of Table 4.2, ILSB offers better results in comparison to LSB and SLSB for various number of embedding bit per pixel. In other words, regardless of mentioned worse values, in all other cases ILSB offers smaller values of AAD, MSE, and L^P-norm in comparison to LSB and SLSB, but offers greater values of SNR, PSNR, and NCC in comparison to LSB and SLSB for various number of embedding bit per pixel.

About the results of Table 4.3, proposed method (ILSB) offers better values of all metrics in comparison to LSB and SLSB methods for various number of embedding bit per pixel. In other words, ILSB offers smaller values of AAD, MSE, L^P-norm, and LMSE and greater values of SNR, PSNR, and NCC versus LSB and SLSB.

In the first (upper) part of the Table 4.4, proposed method offers worse (greater) value of AAD versus LSB method for three bits per pixel. But in all the other cases of first part, ILSB offer better values versus LSB for various number of embedding bit per pixel. In the second (lower) part of Table 4.4, almost all values of the metrics that ILSB offers, are worse than the values that SLSB offers. It would be occurred due to the special conditions of Cove.bmp that mostly consists of absolute black and white pixels. Cove.bmp is the main cover image used in SLSB article, [1].

In the first part of the Table 4.5, proposed method offers better values of the all metrics versus LSB for various number of embedding bit per pixel. In the second (lower) half, for one-bit-per-pixel embedment, the values of AAD, MSE, L^P-norm, SNR, and PSNR are worse than the corresponding values of SLSB. In other words, ILSB offers greater value of AAD, MSE, and L^P-norm versus SLSB, but smaller value of SNR, PSNR, and NCC in comparison to SLSB for one bit per pixel. For two and three bits per pixel, ILSB offer better results versus SLSB.

In the results of Table 4.6, Table 4.7, and Table 4.8, ILSB offers worse (greater) value only for AAD in comparison to LSB and for three bits per pixel. In all other cases of mentioned tables, ILSB offers better results in comparison to LSB and SLSB for various number of embedding bit per pixel.

In the results of Table 4.9, ILSB offers worse (greater) value for AAD and LMSE in comparison to LSB and for three bits per pixel. In all other cases, ILSB offers better results versus LSB and SLSB for various number of embedding bit per pixel.

About the results of Table 4.10 and Table 4.11, ILSB offers better values of all metrics in comparison to LSB and SLSB methods for various number of embedding bit per pixel. In other words, ILSB offers smaller values of AAD, MSE, L^P-norm, and LMSE in comparison to LSB and SLSB that imply less difference between cover image and stego image. On the other hand, ILSB offers greater values of SNR, PSNR, and NCC in comparison to LSB and SLSB that imply more similarity between cover image and stego image for different number of embedding bit per pixel.

In table 4.12 and Table 4.13, ILSB offers better values of metrics versus LSB and SLSB except in AAD versus LSB (upper part) for three bits per pixel. Regardless of this exception, in all other cases, ILSB offers smaller values for AAD, MSE, L^P-norm, and LMSE that state less difference between cover image and stego image, but greater values for SNR, PSNR, and NCC that state more similarity between cover image and stego image in comparison to LSB and SLSB.

Finally in Table 4.14, average values of all metrics are calculated and shown in corresponding cells. According to the above results, ILSB offers better values in comparison to LSB and SLSB methods in all cases for the all number of embedding bit per pixel. In other words, ILSB offers smaller average values for AAD, MSE, L^P-norm, and LMSE that state less difference between cover images and corresponding stego images in comparison to LSB and SLSB. Also, ILSB offers greater values for SNR, PSNR, and NCC that state more similarity between cover images and corresponding stego images in comparison to LSB and SLSB in both two parts.

As an numerical instance, considering average values of the metrics of ILSB versus LSB, AAD, MSE, L^p Norm, and LMSE have respectively % 8.95, % 40.85, % 25.49, and % 19.32 improvements for two bits per color embedment.

About SNR, PSNR, and NCC higher value implies higher quality of embedment. For another example, considering average table and ILSB in comparison to LSB, metrics

SNR, PSNR, and NCC have respectively % 4.92, % 4.44, and % 0.0002 improvements for two bit per color embedment. The NCC's improvement is not significant due to its scale. NCC is correlation and its improvement occurs on the fifth decimal place.

Totally, proposed method (ILSB) offers better results of embedment quality in comparison to LSB and SLSB methods, for various number of embedding pixels.

Chapter 5

CONCLUSION

The focus of this study was to improve the LSB method's efficiency proposing an improved technique which can filter the qualified pixels for embedding secret message bits. In fact, we have proposed a new method of distribution and dispersal of secret message bits over the cover image in a recoverable manner. In proposed method, the color whose most significant bit value is greater than or equal to a particular threshold is qualified to participate in embedment and least significant bit(s) of that color will be replaced with corresponding secret message bit(s). Qualified color or colors for embedding in a pixel may differ from the other pixels. Therefore, the number of secret message bits that can be embedded in each pixel may be different from the others and depends on the colors value of the pixels. Keeping the most significant bits of colors unchanged has ensured the recovery of the secret message bits.

The proposed method and other methods were performed on thirteen different cover images and one secret message. Among the cover images Cove.bmp can be seen which is the main cover image applied in SLSB article and due to its special conditions which consists of absolutely white and black pixels, the obtained results of this image from ILSB are not better than the results obtained from the SLSB method. Regardless of this exception, according to the experimented cover images, and referring to obtained results, new proposed algorithm offers up to %50 better result in metrics compared to existing methods. Trying more cover images and secret message bits with different sizes may give new results which can be considered as a future work.

Different metrics applied for this research are chosen among the most famous metrics to evaluate the quality of embedments from different aspects. Their correctness are all checked and compared with different references. Some other useful metrics can be applied or even created according to what is important for us in embedments. Statistical analysis applied in SLSB method is not worked in this thesis. These two cases can be considered as future works to improve the proposed method.

REFERENCES

[1] J. J. Roque and J. M. Minguet, "SLSB: Improving the steganographic algorithm LSB," in *Proceedings The Ibero-American Congress on Information Security (CIBSI)*, Montevideo, 2009, pp. 398-408.

[2] T. Morkel, Jan H. P. Eloff, and Martin S. Olivier, "An overview of image steganography," in *New Knowledge Today Conference*, Sandton, 2005, pp. 1-11.

[3] (2012, June) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Steganography

[4] M. Kharrazi, H. T. Sencar, and N. Memon, "A Performance Study of Common Image Steganography and Steganalysis Techniques," *Journal of Electronic Imaging*, vol. 15, p. 041104, 2006.

[5] j. Fridrich, M. Goljan, and R. Du, "Reliable detection of LSB steganography in color and grayscale images," *IEEE Multimedia*, vol. 8, pp. 22-28, 2001.

[6] S. Dumitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," *IEEE Transactions on Signal Processing*, vol. 51, no. 7, pp. 1995-2007, July 2003.

[7] A. Ker, "Improved detection of LSB steganography in grayscale images," *Springer LNCS*, vol. 3200, pp. 97-115, 2004.

[8] M. Van Dijk and F. Willems, "Embedding information in grayscale images," in *22nd Symposium on Information and Communication Theory*, Enschede, 2001, pp. 147-154.

[9] M. Goljan and T. Holotyak, "New blind steganalysis and its implications," in *Security, Steganography, and Watermarking of Multimedia Contents*, 2006, pp. 1-13.

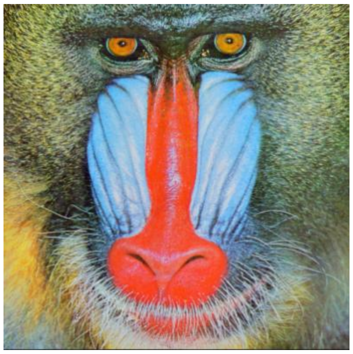
[10] F. Petitcolas and S. Katzenbeisser, *Information Hiding Techniques for Steganography and Digital Watermarking*. Boston: Artech House Books, 1999.

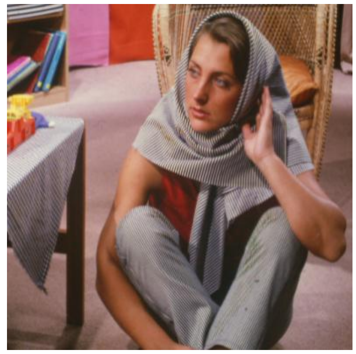
[11] (2012, June) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Absolute_difference

[12] (2012, June) Wikipedia. [Online]. http://en.wikipedia.org/wiki/Lp_space

APPENDIX

Appendix A: Applied Cover Images


Baboon.bmp

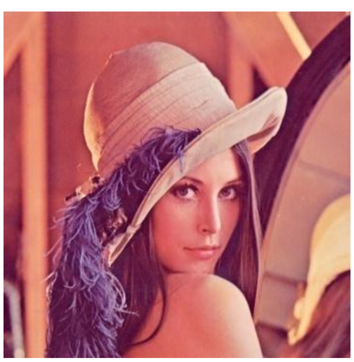

Barbara.bmp


Boats.bmp


Cove.bmp


F16.bmp


Goldhill.bmp


Lena.bmp


Monarch.bmp


Peppers.bmp


Sailboat.bmp


Tulips.bmp


Yacht.bmp


Zelda.bmp