Gray Coded Grayscale Image Steganoraphy using Hufman Encoding

Nithyanandam Pandian# nithyanandamp@ssn.edu.in
Assistant Professor Department of Computer Applications,
SSN College of Engineering,
Kanchipuram Dt. 603110, India

Ravichandran Thangavel dr.t.ravichandran@gmail.com
Principal
Hindustan Institute of Technology,
Coimbatore 641032,India

Corresponding Author

Abstract

Steganography is an art of hiding secret information on a cover medium through imperceptible methodology. The cover medium can be any digital entity ranging from an image, audio, video to any object which can be digitally represented. Embedding capacity, imperceptibility and robustness are the primary goal of steganography. However the requirements of the above said goal vary from application to application. The notion of the proposed work is to focus on improving embedding capacity and bring down the distortion occurring to the stego image. For accomplishing higher embedding capacity and undetectability it is not essential to rely only on the cover image and efficient embedding technique but also on the refinement of the algorithm. One such attempt has been taken in the proposed work by applying gray coding technique to the payload (secret image). The gray coded binary data is bit planed and runlength encoded, prior to the submission of Huffman enoding technique results in supporting higher embedding capacity. The experimental results shows that the assessment metrics such as peak signal to noise ratio, histogram and embedding capacity are highly improved.

Keywords: Steganography,Graycoding,Huffman Encoding, Peak Signal to Noise ratio, distortion,redundancy.

1. INTRODUCTION

Steganoraphy is a secret communication technique, in which the event of communication taking place itself is concealed. Cryptography and watermarking are the close related techniques to steganography. Fundamentally both cryptography and steganography are information securing technique but they differ in their implementation. Cryptography makes secret data unreadable by a third party, whereas steganography hides secret data from a third party. Both of their notions [1] remain the same. The cover medium suitable for a steganography [2] can be any entity that can be digitally represented such as a text file, image, audio, video and unused portion of TCP/IP packet headers. Steganography [3] and watermarking [3] are information hiding technique but the notion between them is different. Watermarking is claiming/proving the ownership of any digital entity, whereas steganography is a secret/hidden communication accomplished in a digital entity. The impact of embedding secret data on the cover medium should be lighter in order to achieve undetectability. The payload is relatively small in watermarking on converse with steganography. Because steganography is a secret communication technique, whereas watermarking is a authentication certificate made up of watermark merely take size of few bytes. In watermarking the watermarks are basically a metadata which help to prevent copyrights and other cyber related issues. Unlike watermarking, steganography does not have a constraint in choosing the cover medium to establish secret communication. Application of steganography is vast and in each application the degree of imperceptibility, embedding capacity and robustness [2] does not remain same. For example one application may focus on higher imperceptibility compromising embedding capacity whereas in other case lesser importance to robustness but imperceptibility and embedding capacity should be the best. All the above said parameters are interdependent and it is a tradeoff [29] in controlling each other.

LSB embedding [4],optimum pixel adjustment process [5][6], pixel value differencing [7][8][9] and matrix embedding [1][3][4][10] were some of the spatial domain embedding technique. All the spatial domain techniques are flexible to support fixed bits, variable bits, and adaptive embedding on their pixel intensities. F5 [3], Outguess[3], JP Hide and Seek[11] were some of the frequency domain technique. The limitation of frequency domain is its inflexibility of supporting [12] low payload to embed. Therefore ideally, spatial domain embedding technique [17] is preferred over frequency domain which helps to attain higher embedding capacity.

Supporting a higher payload on a cover image primarily depends upon the embedding mechanism (algorithm). But it also can be viewed in another direction of compressing the payload prior to embedding on the cover image. With respect to an image, there exists several redundancies such as [13] coding redundancy, interpixel redundancy and psychovisual redundancy. The intensity values in the two dimensional matrix of an image are represented by n bits, which is called as bit depth of an image. Bit depth vary from 2 to 32 bits depends upon the support offered by the hardware and software. Among the redundancy, coding redundancy [13] is the one which can be confidently used to attain error-free or lossless compression. Huffman encoding [14] is a variable length lossless compression technique, can be applied to any entity which can be digitally represented. Several coding techniques can be used in conjunction with Huffman encoding to compress the payload. Integrating all the above said entity under a single strand ends up with higher compression results in high embedding capacity of a steganography algorithm. This is discussed detail in background section.

The objective of the proposed work is to embed a grayscale secret image on a 24 bit RGB cover image using various coding technique. The gray coded binary data of secret image is bit planed and runlength encoded, prior it is submitted to Huffman encoding technique results in supporting higher embedding capacity and peak signal to noise ratio. The rest of the paper is organized as follows: Section 2 discusses the background study, Section 3 covers the related works. The proposed work is exhibited in Section 4, experimental results and discussions were given in Section 5. Finally, the concluding remark and future direction are given.

2. BACKGROUND STUDY

Different coding techniques involved in the proposed work should be studied individually helps to understand the role carried by each of them in the proposed work. Coding technique such as gray coding, bit plane coding, run length coding and Huffman coding are discussed in detail which helps in understanding the attainment of high embedding capacity and less distortion.

2.1 Run Length Coding

Run length coding [15] is a simple form of data compression. Image intensity is represented by a 8 bit binary magnitude in a gray scale image. It is obvious that alike pixel intensities are get repeated across the rows and columns of the image (interpixel redundancy [13]). It is unwise to code every repeated intensity with same 8 bit binary magnitude. Instead of coding all the repeated intensities (values) with fixed bits just encode the binary runs of each consecutive 0's and 1's. For example consider 129,129,129,129,129 are the five consecutive gray scale pixel intensities. Their 8 bit binary magnitude representation (10000001 10000001 10000001 10000001 10000001) sums to 40 bits. If the same is run length encoded it can be represented as follows: 1,6,2,6,2,6,2,6,2,6,1. Here the runs refer to consecutive number of 0's ad 1's. In a binary model,1 can be represented in single bit and 2 can be represented in 2 bit and 6 can be represented in 3 bits. Therefore two 1's, four 2's and five 6's (2 x 1 + 4 x 2 + 5 x 3) can be represented using 25 bits. The bit rate is 25/5=5 bits per pixel in lieu of 8 bits per pixel accounts to saving of 37.5 percentage. Let us scale this concept to a small image matrix: For example consider a 4 bit depth of 10 X 10 image show in Table1.

9	11	11	11	11	11	11	11	11	11
10	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	11	11	11	11
11	11	11	11	11	11	10	11	11	11
11	11	11	11	11	11	10	10	11	11

Table1: Four bit representation of 10 x 10 image

It's binary representation (across the column) is given as follows:

1 0 0 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1
1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1
0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1
0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1
0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1
0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1
0 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1
0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1

Run length encoding of the above binary representation is as follows:

0,1,2,2,1,1,1,1,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3
,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,
3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,1,1,1,1,1,1,1,3,1,3,1,3,1
,3,1,3,1,3,1,3,1,3,1,3,1,1,1,1,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,1,3,
1,3,1and 2

If the above run lengths are encoded using variable bit Huffman encoding it requires 401 bits in place of 400 bits raw image (4 x 10 x 10). Applying Huffman encoding does not stimulated the compression. But this can be improved, if the raw image binary pattern is bit pane coded, run length encoded and finally compressed by Huffman encoding. In general runlength encoding result cannot be directly embedded in the cover image pixels. It should be properly coded through any variable bit embedding technique, such as Huffman, arithmetic coding, lossless predictive coding [13] etc. Here in the proposed method it is coded using Huffman encoding.

2.2 Bit Plane Coding

Bit plane coding [13] is breaking the image in to bit planes and applying run length coding to each plane. Most probably the bits in most significant part seldom changes. In the above example we had considered (4 bit depth image) 3^{rd} bit is the most significant plane and 0^{th} bit is the least significant plane. Decomposing the 4 bit depth image intensity in to 4 individual bit planes and encoding separately every plane by run length technique will give higher magnitude in the runs. The run length magnitude may progressively decline from lower plane to higher plane. This is maximum for 0^{th} plane and minimum for 3^{rd} plane. This is due to the most significant part is responsible for visually significant data and least significant part show the finer details of an image. The performance of run length coding will be far better if the magnitude of every run length is high. For example in the following binary pattern: 10000000 10000000 10000000 10000000 the run length is 1,7,1,7,1,7,1,7. Bits required to encode is 16 bits (1 bit require for 1 and 3 bits require for 7). For this binary pattern 10111011 10111011 10111011 10111011 the run length is 1,1,3,1, 3,1,3,1,3,1,3,1,3,1,3,1,2. Bits required to encode is 25 bits (1 bit require for 1, 2 bits for 2 and 2 bits for 3). From the former it is apparent that the run length performance can be improved if the magnitude of every run length entry is high. This is explained from the earlier example (4 bit depth 10 x 10 image) we had considered. The bit plane coded binary data for the 4 bit depth 10 x 10 image is as follows:

1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

The bit plane coded binary data given above are arranged as per the following:
3rd bit plane followed by 2nd bit plane followed by 1st bit plane and finally 0th plane.

Run length encoding of the above bit planed binary data is as follows:
0,1,1,66,2,9,1,20,1,99,100 and 100

The binary pattern has a lesser mix of both 0's and 1's in 3rd plane. In the 2nd plane one zero and 99 ones. 1st plane has all zeroes and 0th plane has all ones. The runlength magnitude is higher which results in higher compression. If the above run lengths are encoded using variable bit Huffman encoding it requires just 195 bits in lieu of 400 bits raw image (4 x 10 x 10).This can be further improved if the raw image binary pattern is gray coded prior to bit pane coding, run length encoding and finally compressed by Huffman encoding.

2.3 Gray Coding

Gray code [13][16][17] are non-weighted code used in encoding a sequence of integers {0, 1, 2, …, n} so that the binary representation of adjacent integers differs by only 1 bit. For example, the Gray encoding for the sequence {0, 1, 2, 3, 4, 5, 6, 7} is {000, 001, 011, 010, 110, 111, 101, 100}. It is not true in the case of binary weighted codes i.e. 8421 code. Consider a grayscale image, whose intensity value is represented by 8 bit binary weighted code. The inherent disadvantage of this approach is that a minor change (adjacent value) in gray scale pixel intensity has a major impact on all or some of the bit planes. For example, the 8 bit binary weighted code of pixel intensity value 127 is represented by 01111111 whereas for 128 it is 10000000. Here the entire 8 bit planes are get modified to represent 128. Even if the binary bits are bit plane encoded it does not yield higher compression through run length coding. A mechanism should be devised, such that the effect of minor gray level variation should not influence much the bit planes. So, an alternative approach is to decompose and represent the image by k-bit gray code. For the above said gray level (intensity) 127 and 128 the gray codes are 01000000 and 11000000 respectively. There is a minor change in 7th bit position alone, a transition of 0 to 1. A k-bit graycode $g_{k-1} \ldots g_2 g_1 g_0$ that corresponds to binary weighted code equation1 is given in equation2.

$$P_i = (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2 = \sum_{n=0}^{7} b_n \, X \, 2^n \qquad \ldots\ldots\ldots\ldots (1)$$

$$g_i = a_i \oplus a_{i+1} \qquad 0 \le i \le k-2 \qquad \ldots\ldots\ldots\ldots (2)$$
$$g_{k-1} = a_{k-1}$$

.

Here, ⊕ denotes exclusive OR operation. Gray code has the unique property that consecutive code words differ in only 1 bit position which mitigates in affecting all the bit planes. Thus, run length encoding find greater run length magnitude which ends up with higher compression. This is explained from the earlier example (4 bit depth 10 x 10 image) we had considered. Binary representation for the 4 bit depth 10 x 10 image is given above. The gray coded bit planed binary data is as follows:

1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

The bit plane coded binary data given above are arranged as per the following:
3^{rd} bit plane followed by 2^{nd} bit plane followed by 1^{st} bit plane and finally 0^{th} plane.

Run length encoding of the above gray code bit planed binary representation is as follows:
0, 2, 66, 2, 9,1,21 and 299

The binary pattern of 2^{nd}, 1^{st} and 0^{th} plane has almost one. A lower mix of both 0's and 1's occurring in 3^{rd} plane which founds to comparatively better than bit plane coding. If the above run lengths are encoded using variable bit Huffman encoding it requires 164 bit depths in lieu of 400 bits raw image (4 x 10 x 10). The inference is that in bit plane coding the bigger run length magnitude is 299 which combines 2^{nd}, 1^{st} and 0^{th} plane. This bigger run magnitude has dragged the Huffman encoding to attain the compression of the 4 bit depth 10 x 10 image in just 164 bits. This concept has been scaled and used in the proposed research.

2.4 Huffman Encoding

There are several redundancies in an image such as [13] coding redundancy, interpixel redundancy and psychovisual redundancy. Among the redundancy, coding redundancy is the one which can be confidently used to attain error-free or lossless compression. Huffman encoding [14] is a variable length lossless compression technique can be applied to any entity which can be digitally represented. The pixel intensities of an image were treated as different symbols. The symbols vary from 0- 255. 0 denotes minimum and 255 denotes maximum intensity. The intensity of the pixels across the image is spatially correlated [13]. Information is pointlessly repeated in the representation of the correlated pixels. These repetitive pixels should also be represented by fixed number of bits in unencoded Huffman format. Actually these values are the best source for exploiting compression. Frequently occurred intensity value can be represented by variable numbers of bits (i.e. shorter bits) in contrast to the fixed number of bits for representing the pixel intensity used in unencoded Huffman technique. This is the core concept of Huffman encoding technique.

Huffman table has prefix free codes and corresponding symbols. Huffman table is used during Huffman decoding. Both the Huffman encoded prefix free binary code and Huffman table should be treated as payload and embedded in cover medium. The compression ratio depends on the count of the unique symbols and their frequencies. Lesser the symbol and higher the frequency results in higher compression, in contrast higher the symbols with less frequency results in lower compression. This is common in both text and image compression. Generally if the symbol count increases, this in turn influences the size of Huffman table. As a result the Huffman table entries get increased, which has impact on the compression ratio. Huffman table is an additional overhead in Huffman encoding technique. To summarize, the performance of Huffman compression depends on the count of unique symbols and their frequencies. Huffman compression on various information coding method for the 4 bit depth 10 x 10 image is presented in Table 2.

Image of 4 bit depth / Method	Huffman encoding (size in bits)	Huffman table (size in bits)	Huffman compression (size in bits)
Run length	312	89	401
Bit Plane and Run length	34	161	195
Gray code ,Bit Plane and Run length	22	142	164

TABLE 2: RESULTS OF HUFFMAN COMPRESSION ON VARIOUS CODING METHODS

3. RELATED WORKS

A stenographic technique proposed in [18] which is based on LSB replacement technique. Varying lengths of secret bits get embedded in every pixel. In method1 green and blue are embedding channels keeping red channel as indicator channel. In method2 an option is provided for choosing the indicator channel among the three channels. Once chosen, the remaining two channel act as embedding channel. In method3 the indicator channel is chosen by rotation scheme across all the pixels. In the first pixel red channel is indicator; green channel is the indicator in second pixel and in third channel blue act as indicator. Once indicator is finalized the remaining two channels will be used for embedding. This scheme is repeated for the consecutive pixels. The Mean Square Error (MSE) and PSNR is calculated for all channel and the average number of bits get embedded in every pixel is shown in their results.

A stenographic technique proposed in [7] is based on edge adaptive scheme. The absolute difference between two adjacent pixels is the primary criteria in identifying the region for embedding secret message. They use LSBMR (Least Significant Bit Matching Revisited) as their data hiding algorithm. Only 2 secret bits can be embedded in each embedding unit and threshold T is used in identifying the embedding region. Region selection and cover image hiding capacity is determined through trial and error process. The sharper edge regions of cover image alone are used for embedding. Even though the embedding capacity is lesser it withstand against statistical attack and they had proved that RS steganalysis is ineffective in detecting stego work.

In the steganography scheme adopted in [19], the embedding efficiency is improved by adopting matrix embedding method. ME-RA (matrix embedding repeat accumulate) is the data hiding algorithm used to hide the secret data. The reason to choose matrix embedding is to less adulterate the cover image, at the same time the secret data bits should get embedded. Here, a hamming code matrix is employed in attaining the goal. In the proposed work, instead of hamming code (for matrix embedding) a simple XOR operation is performed on the host image bits to check its coincidence against the secret bits. The host image bit is adjusted accordingly to suit the secret bits.

A novel image steganography technique [20] was discussed in which the cover image's spatial value is transformed in to Discrete Cosine Transformation (DCT); its LSB is modified to match the secret message. The secret message is Huffman encoded prior to the embedding scheme which achieves a significant compression rate. A higher embedding capacity and PSNR is obtained using this technique. This technique is superior to the method proposed in [23].

A stenographic technique [21] based on wavelet transformation on the images is proposed. Discrete Wavelet Transformation (DWT) converts the spatial domain of cover image into frequency domain. Huffman compression is applied for the stream of secret bits before overlaying them on the cover image. A high PSNR and very high embedding capacity is achieved. A higher level of security is obtained because the Huffman table and encoding rules is black box to the intruder.

A Least Significant Bit [22] steganographic scheme technique in which the secret image is bit plane coded, runlength encoded and finally Huffman compressed yielded a high embedding capacity. The payload (secret image) get embedded is just 4 bits out of every 8 bit pixels. Remaining 4 bits of every pixel is artificially constructed in the destination during the process of making the secret image from the stego image. The secret image retrieved in the destination is a lossy one, but sill PSNR above 30dB is acceptable.

4. PROPOSED METHOD

Applying coding technique on the payload will compress its size and further Huffman encoding reduce the coding redundancy present in an image. This is the central theme behind the proposed work. In the proposed method of [22], coding technique such as bitplane,runlength and Huffman encoding were applied. Gray coding can be attempted in addition prior to all the above mentioned coding technique which will improve the performance. This is the central theme behind the proposed work.

4.1 Least Significant Embedding

Least Bit (LSB) embedding [4][6][22][24][25][26][30] is one of the renowned spatial domain steganography techniques. The process of LSB embedding carried on a cover medium is explained below. Suppose we want to encode the letter A (**ASCII 65 or binary 01000001**) in the following 8 bytes (pixel intensities) of an image cover medium, it can be done as follows:

93	**208**	**28**	**172**	**231**	**135**	**107**	**227**
01011101	11010000	00011100	10101100	11100111	10000111	01101011	11100011

becomes

92	**209**	**28**	**172**	**230**	**134**	**106**	**227**
0101110**0**	1101000**1**	0001110**0**	1010110**0**	1110011**0**	1000011**0**	0110101**0**	1110001**1**

4.2 Matrix Embedding

In matrix embedding technique, the LSB's of all the channels of every pixels are taken to embed two bits of secret image.

4.2.1 Embedding

To encode bit b_1 and b_2 in the LSB of three channels of a pixel: say X_1, X2 and X3. Equation3 must be satisfied.

$$\begin{aligned} b_1 &= LSB(x_1) \oplus LSB(x_2) \\ b_2 &= LSB(x_2) \oplus LSB(x_3) \end{aligned} \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

If equation3 is not satisfied, a minor modification will be done on X_1, or X_2 or X_3 to satisfy the same. If the first one is satisfied but not the second one, simply flip the LSB of x3. If the second one is satisfied but not the first one, flip the LSB of x1. If neither is satisfied, flip LSB of x2.

4.2.2 Extraction

Every consecutive 3 bytes of stego image are extracted to construct 2 bits of secret image. This process is repeated until the required secret image bits are constructed. To extract bit b_1 and b_2

from LSB of three channels of a pixel: say X_1, X_2 and X_3 equation4 is applied on the stego image pixels.

$$LSB(x_1) \oplus LSB(x_2) = b_1$$
$$LSB(x_2) \oplus LSB(x_3) = b_2 \quad \ldots\ldots\ldots\ldots..\ldots..(4)$$

These collected bit streams are grouped and submitted for further operation of constructing the secret image.

4.3 Embedding Process

Fig.1 shows the embedding process carried on the sender side. After embedding the sender will send the cover image which has secret image embedded in it. The steps carried on the sender side are given below.

Step1: From each pixel of the secret image, remove the last four LSB's. Apply gray coding followed by bit plane slicing in the remaining four MSB's.
Step2: Apply run length encoding for the output obtained from Step1.
Step3: Apply Huffman encoding for the output obtained from Step2 which results in Huffman table and Huffman encoded secret image bit streams.
Step4: Embed dimensions of secret image and the resultant component (Huffman table and Huffman encoded secret image bit streams) obtained from Step3 into the cover image using matrix/LSB embedding technique.
Step5: Send the stego image obtained from Step5 to the receiver.

4.4 Extraction Process

Fig.2 shows the extraction process carried on the receiver side. Upon receiving the stego image, the receiver should extract the Huffman table, Huffman encoded bit streams, the four LSBs equivalent decimal value (which was computed during embedding on the sender side) and secret image dimension from the stego image. The steps carried on the receiver side are given below.

Step 1: For retrieving the secret image from the stego image, extract the LSB's from every pixel, apply matrix/LSB extraction technique and construct Huffman table and Huffman encodings using the LSB's bit streams.
Step 2: The Huffman table and Huffman encodings obtained in Step1 is used in Huffman decoding process. As a result a run length pattern is produced.
Step 3: Apply run length decoding to the result obtained in Step2 which results in producing corresponding binary runs consist of 0's and 1's.
Step 4: Finally, the image is bit plane, gray decoded and later the extracted decimal value of LSB four bits are converted to binary and appended to the existing 4 MSB's to make an 8 bit gray scale value. At last the image is constructed using all the pixels which will reveal the secret image.

The four bit decimal value which is substituted during secret image construction process in the receiver side is, precomputed and stored in the cover image by the sender. This precomputation is explained in detail. Three types of decimal value are computed for the 4 bit LSB, such as maximum value, minimum value and random value and any one among them is get substituted.

4.4.1 Maximum value: The 4 LSB's of all the pixels are separated and its decimal value is determined. The decimal value range will be from 0- 15, since it is just 4 bits. The individual count of every decimal value is identified and the decimal value which has higher count become the substitution value for the 4 LSB in the destination side. This value is stored in the stego image in addition to its size,Huffman encodings and Huffman table.

4.4.2 Minimum value: The individual count of every decimal value is identified and the decimal value which has lower count become the substitution value for the 4 LSB in the destination side.

4.4.3 Random value: A random value ranges from 0 -15 is computed for every 4 MSB and this random 4 bit value is padded to get 8 bit gray pixel value.

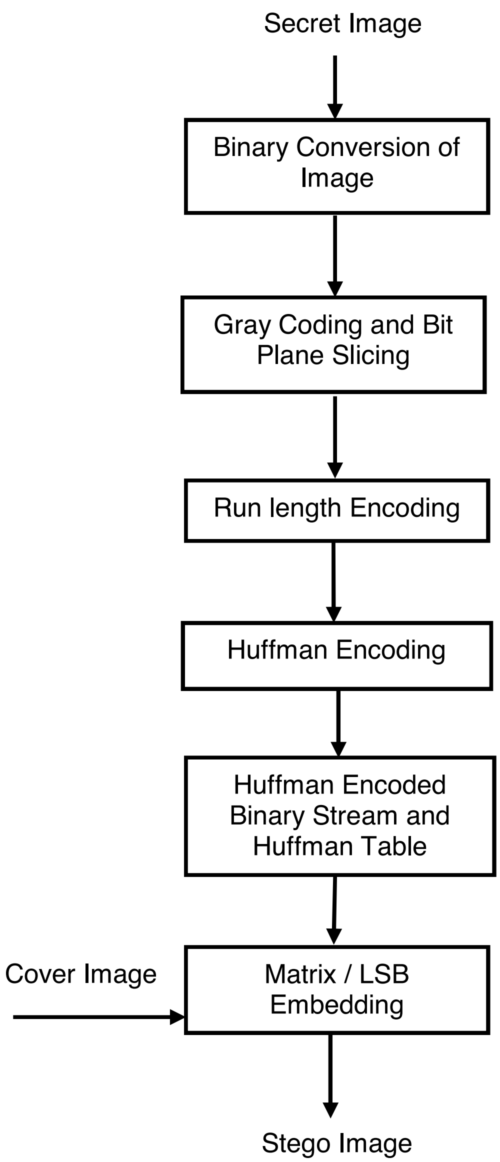


Figure 1: Embedding Process

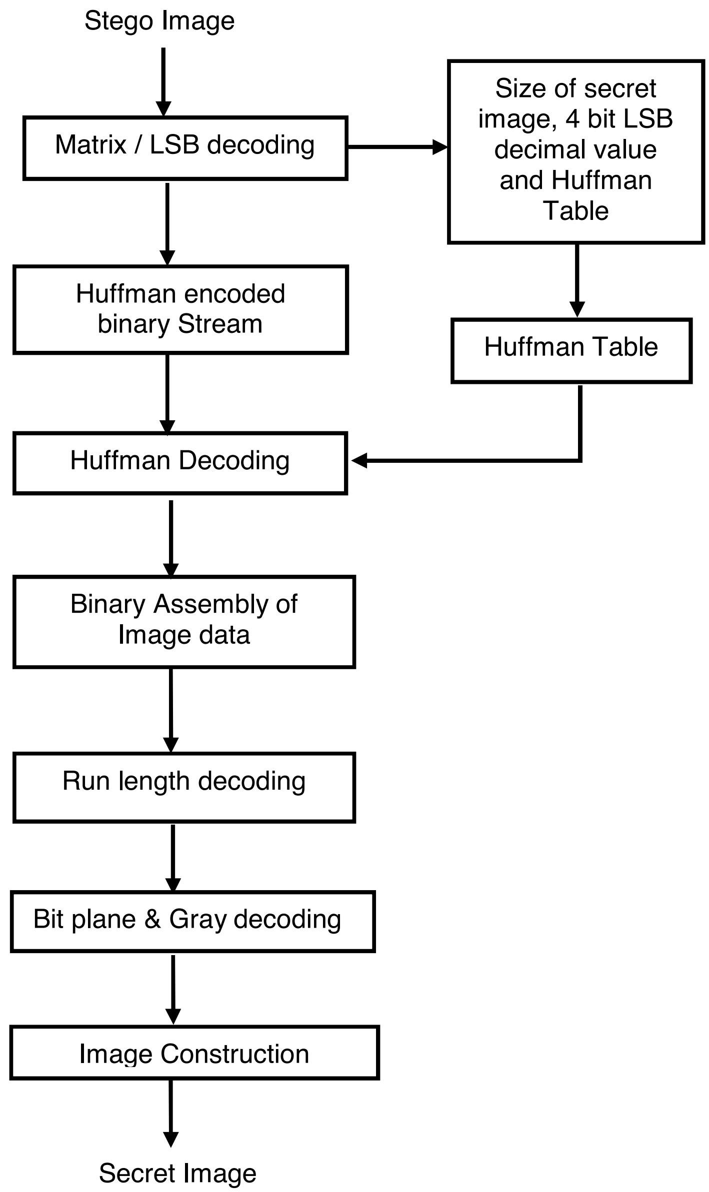


Figure 2: Extraction Process

$$MSE = \frac{1}{m*n}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(A_{ij} - B_{ij}\right)^2 \qquad \ldots\ldots\ldots\ldots(5)$$

All the above three substitution mechanism has attempted and their result (PSNR) while constructing the secret image is presented in Table5.The rate of adulteration between the cover matrix and stego matrix should be measured. To assess this, mean square error and Peak Signal to Noise ratio between the matrixes should be determined. The equation for MSE and PSNR are followed:
where A_{ij} represents pixel in the cover image and B_{ij} represents pixel in the stego image; m, n represents the height and width of the image respectively.

$$PSNR = 10 * log_{10}\left(\frac{Max^2}{MSE}\right) \quad \ldots\ldots\ldots\ldots.(6)$$

here max denote maximum color intensity of grayscale (255).PSNR is measured in decibels (dB).

5. EXPERIMENTAL RESULTS

Java 2.0 and MATLAB R2009a are the programming tools used to implement the proposed method. Standard 24 bit cover images of size 512 x 512 such as lena, baboon, flight and boat were used. Fig. 3a-3c shows the secret image and Fig 4a-4d shows the cover images used in the experiments. Fig.5a-5d shows the stego image obtained for LSB embedding. Fig. 6a-6d and 7a-7d shows the histogram of the cover and stego image respectively for LSB embedding technique. Similarly Fig. 8a-8d and 9a-9d shows the histogram of the cover and stego image respectively for Matrix embedding technique. PSNR and embedding Capacity are the metrics taken here to consolidate the strength of proposed method. Histogram of the stego image shows that the distortion between cover and stego image is minimum. Gray coding is used in the proposed to improve the performance of Huffman coding. The performance of Huffman encoding depends upon unique symbol present in the secret image. Here symbol refers to various unique pixel intensity or grayscale value. Table3 shows the experimental results compared against [21],[22] and [27]. There is a significant improvement in PSNR and embedding capacity through the proposed method. The improvement obtained for LSB and matrix embedding is exhibited in both Table3 and Table4. The maximum embedding capacity of [22] is experimented in the proposed method and the revised result using the proposed method is given in Table4. The maximum embedding capacity of the proposed method given in Table4. The secret image constructed in the destination is not as exact as the original secret image. The 4 LSB decimal value required for every pixel of secret image are unanimously replaced by either one of the following three entities: a maximum value, minimum value or random value which is explained in detail section 4.4. The quality of the secret image which is retrieved from the stego image is assessed. The experimental result is exhibited for various secret images in Table5. It is inferred from Table5 that PSNR of secret image is influenced by the substitution value get replaced while constructing the secret image in the destination. For Fig.3a, random value substitution gives good result whereas in the case of Fig.3b it is maximum and finally for Fig.3c it is minimum value. PSNR above 30dB is acceptable [28] but high quality recovering (secret) image should strive for 40dB and above.

The last 64 pixel in cover image is reserved for storing the technical details, which will be used in the receiver side to extract the secret data from the stego image. This 64 pixel (64x3=192 bytes) should be excluded while computing the maximum hiding capacity of cover image. Secret data of more unique symbol with any size can be hidden through our proposed method, provided it meets the above said condition.

5.1 Discussion

In the proposed method, gray coding is done prior to other coding technique which has enhanced the embedding capacity when comparing [22]. The reason is due to the principle that any two adjacent [13][16] decimal value when represented in binary pattern may have a huge difference between each other in their patterns. In the case of gray coding the binary pattern between any two adjacent gray coding value there will be a maximum of 1 bit change between each other. This nature is utilized to get a high throughput.

We quite often found that a secret image which is richer and whose dimension is lesser than Cameraman 256 X 248 shown in Fig. 3 (image used as secret image in the proposed method) cannot be embedded in this 512 x 512 cover image Lena. In contrast, a secret image which is not richer whose dimension is higher than cameraman 256 X 248 can be embedded in the cover image. This makes us to finalize that the embedding capacity of our proposed technique depends on Huffman encoding. Any image, whose Huffman compression is less, fits in the cover image irrespective of its size and richness. To discuss on security side, the proposed technique is robust enough; because extracting a data without knowing the architecture of the proposed technique is difficult, moreover data is Huffman encoded. Based on how the symbols are treated, the implementation of Huffman may vary. But however the basic rule that should be adhered is that Huffman implementation should have prefix free coding. Even if the intruder collects the LSBs from the cover image, the intruder should separate the Huffman table and actual Huffman encodings. And also, only four bits are transmitted, remaining four bits are constructed. Working-out this idea is another level of difficulty. Moreover, the Huffman output should be run length decoded, biplane decoded and finally gray decoded to construct the actual image. It is cumbersome to attack all the stages of our proposed algorithm.

Cover Image 512×512	Capacity (bits)	PSNR(dB)							
		DWT base [9]	DWTand Huffman Method [11]	Bitplane, Runlenth and Huffman Method [22]		Proposed Method			
				LSB	Matrix	LSB	Improved PSNR	Matrix	Improved PSNR
Lena	507856	46.08	54.93	53.54	54.79	54.14	0.60	55.39	0.60
Airplane	507856	45.99	54.67	53.54	54.77	54.14	0.60	55.37	0.60
Baboon	507670	46.19	55.11	53.52	54.78	54.13	0.61	55.40	0.62
Boat	507867	46.13	54.80	53.51	54.77	54.15	0.64	55.40	0.63

TABLE 3: Comparison of PSNR results for the Proposed Method against [9] , [11] and [22]

Cover Image 512×512	Capacity (bits)	PSNR(dB)					
		Bitplane, Runlenth and Huffman method [22]		Proposed Method			
		LSB	Matrix	LSB	Improved PSNR	Matrix	Improved PSNR
Barbora 270 x270	583200 (71 KB)	52.96	54.22	54.03	1.07	55.26	1.04
Barbora 380 x 380	1155200 (141 KB)	---		52.94	-----	54.19	----

TABLE 4: PSNR results for Maximum Embedding Capacity for Barbora in LSB and Matrix embedding

Methods / Obtained Secret Images' PSNR	Minimum	Maximum	Random
256 x 248 Cameraman	33.20	31.79	**33.20**
270 x 270 Elaine	**32.80**	29.20	31.79
380 x 380 Barbora	30.07	**33.89**	30.07

TABLE 5: Different Secret Images' PSNR, embedded on Lena cover image of size 512 x 512

CONCLUSION

We had proposed an image steganography algorithm which brings a better PSNR than [22]. Histogram of stego image and cover image are almost equal which emphasize on the result that distortion between cover and stego image is minimum. Capacity improvement and distortion reduction has been addressed in this proposed technique. In the proposed work, the embedding capacity of the cover image is increased, at the same time the PSNR also controlled. The proposed technique is not robust against any geometrical distortion such as rotation, translation, scaling, cropping etc., induced on the stego image.

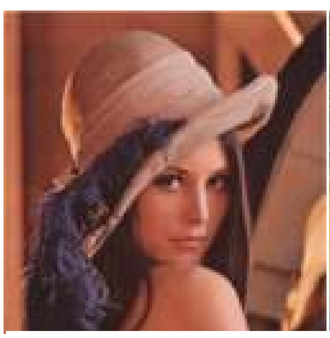
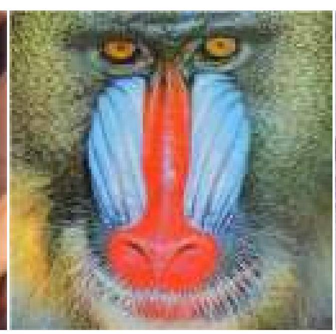
FUTURE WORK

The secret images which is constructed at the destination is resembling to be exactly the original image sent from sender side; but still it is lossy picture generated by constructing the 4 LSB (max,min and random) and 4 MSB received. The proposed work should be refined such that the PSNR of secret image should go beyond 40dB. The proposed algorithm should be customized to support embedding in the frequency domain. It should be enhanced to support color images and withstand geometrical distortion induced on the image.



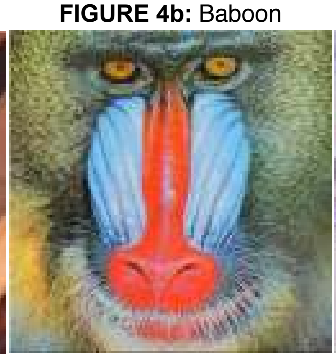
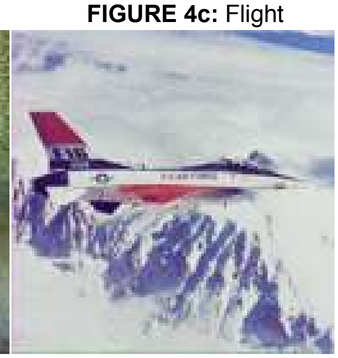
Secret Images

FIGURE 3a: Cameraman **FIGURE 3b:** Barbora **FIGURE 3c:** Elaine

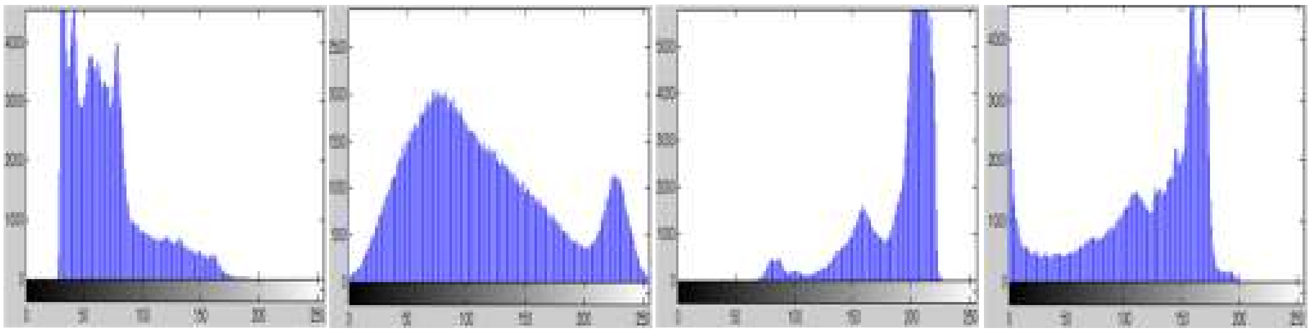
Cover Images

FIGURE 4a: Lena **FIGURE 4b:** Baboon **FIGURE 4c:** Flight **FIGURE 4d:** Boat

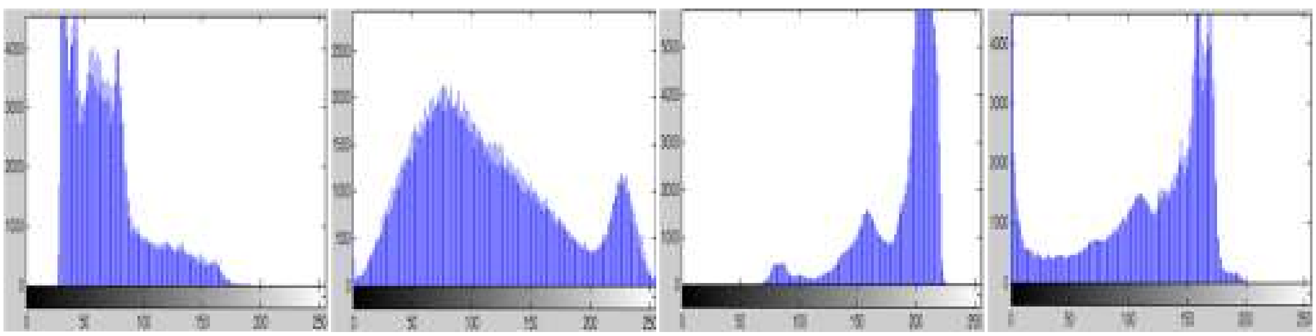

Stego Images: Least Significant Embedding

FIGURE 5a: Lena **FIGURE 5b:** Baboon **FIGURE 5c:** Flight **FIGURE 5d:** Boat



Histogram Images: Cover Images

FIGURE 6a: Lena **FIGURE 6b:** Baboon **FIGURE 6c:** Flight **FIGURE 6d:** Boat

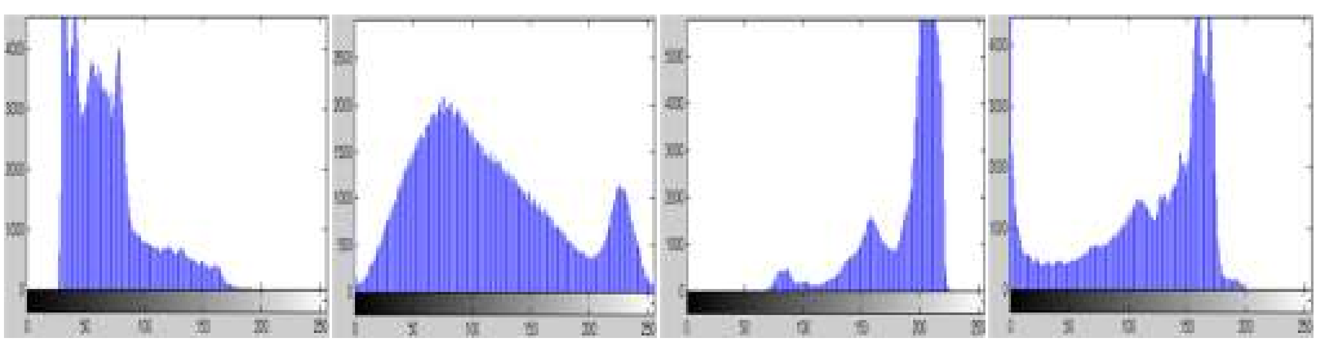


Histogram Images: Least Significant Bit Embedding

FIGURE 7a: Lena **FIGURE 7b:** Baboon **FIGURE 7c:** Flight **FIGURE 7d:** Boat

Stego Images: Matrix Embedding

FIGURE 8a: Lena **FIGURE 8b:** Baboon **FIGURE 8c:** Flight **FIGURE 8d:** Boat



Histogram Images: Matrix Embedding

FIGURE 9a: Lena **FIGURE 9b:** Baboon **FIGURE 9c:** Flight **FIGURE 9d:** Boat

REFERENCES

[1] Mao,J-F. Ru,Z.,Niu,X-X.,Yang,Y-X.,and Zhou,L-N "Research of Spatial Domain Image Digital Watermarking Payload", *EURASIP Journal on Information Security*, pp.1-12, Mar. 2011,DOI:10.1155/2011/502748.

[2] Tayana Morkel, Jan H P Eloff and Martin S Olivier "An Overview of Image teganography", *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, Sandton, South Africa, Jul. 2005.

[3] Injemar J. Cox,Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich and Ton Kalker. "*Digital Watermarking and Steganography",* Burlington, MA 01803, USA: Morgan Kaufmann, Second Edition,2008.

[4] Sutaone, M. S., and Khandare, M.V. "Image Based Steganography Using LSB Insertion Technique" in *IET International Conference on Wireless, Mobile and Multimedia Networks*(Beijing,China) IET,2008, pp.146-151.

[5] Amirtharajan, R., Behera, S.K., Swarup, M.A., Ashfaaq, M.K. and Rayappan, J.B.B. "Colour guided colour image steganography", *Universal Journal of Computer Science and Engineering Technology*, 1(1),16–23,Oct.2010.

[6] Chan,C-K., and Cheng, L.M. 2004. "Hiding data in images by simple LSB substitution",*Pattern Recogn*. 37(3),pp.469 – 474.DOI:10.1016/j.patcog.2003.08.007.

[7] Luo,W.,Huang,F.,and Huang,J. "Edge Adaptive Image Steganography Based on LSB Matching Revisited" *IEEE T INF FOREN SEC*, 5(2), pp.201-214,2010, DOI:10.1109/TIFS.2010.2041812.

[8] Wu,D-C.,and Tsai,W-H. "A steganographic method for images by pixel-value differencing".*Pattern Recogn.Lett.* 24(9),pp.1613–1626,2003. DOI:10.1016/S0167-8655(02)00402-6.

[9] Wu,H. C.,Wu,N.I.,Tsai,C.S.,and Hwang,M.S. "Image steganographic scheme based on pixel-value differencing and LSB replacement methods". *IEE P-VIS IMAGE SIGN*, 152(5),pp.611–615,2005. DOI:10.1049/ip-vis:20059022.

[10] Fridrich,J.,and Soukal, D. "Matrix Embedding for Large Payloads", *IEEE T INF FOREN SEC*,1(3),pp.390-395,2006. DOI: 10.1109/TIFS.2006.879281.

[11] Kipper,G., "*Investigator's Guide to Steganography"*, Virginia:Auerbach Publications,2003.

[12] Kharrazil,M.,Sencar,H.T.,and Memon,N. 2004. "W*SPC Lecture Notes Series:ImageSteganography:concepts and practice",* [On-line] http://iwearshorts.com/Mike/uploads/2011/06/10.1.1.62.8194.pdf [Jun. 1, 2010]

[13] Gonzalez, R.C. and Woods, R.E. "*Digital Image Processing"*,3rd Edition , New Delhi, India:PHI, 2008.

[14] David A. Huffman "A method for the construction of minimum-redundancy codes". *Proceedings of the Institute of Radio Engineers,* 40(9):1098–1101, Sep. 1952.

[15] Data Compression Project. [Online] Available at http://www.binaryessence.com [Feb. 20, 2011]

[16] Darko Dimitrov, Tom´aˇs Dvoˇr´ak, Petr Gregor, and Riste ˇ Skrekovski , "Gray Code Compression" *Combinatorial Algorithms Lecture Notes in Computer Science, Springer, Volume 5874, pp 183-193 2009.*

[17] Anil K. Maini ,"*Digital Electronics, Principal, Devices and Applications*" , *Hoboken, New Jersey: John wiley & Sons*, 2007.

[18] Amirtharajan R, Sandeep Kumar Beher, Motamarri Abhilash Swarup, Mohamed Ashfaaq K and John Bosco Balaguru Rayappan. **"**Colour Guided Colour Image Steganography", *Universal Journal of Computer Science and Engineering Technology*, Volume 1, pp.16 – 23, Oct . 2010.

[19] Sarkar,A.,Madhow,U.,and Manjunath B.S. "Matrix Embedding With Pseudorandom Coefficient Selection and Error Correction for Robust and Secure Steganography",*IEEE T INF FOREN SEC*,5(2),pp.225-239,2010,DOI:10.1109/TIFS.2010.2046218.

[20] Amitava Nag, S. Biswas, D. Sarkar, P.P. Sarkar. "A Novel Technique for Image Steganography Based on Block-DCT and Huffman Encoding" , *International Journal of Computer Science and Information Technology*, Volume 2, Number 3, pp. 103-112,June 2010.

[21] Amitava Nag, S. Biswas, D. Sarkar, P.P. Sarkar. "A Novel Technique for Image Steganography Based on DWT and Huffman Encoding" , *International Journal of Computer Science and Security*, Volume 4, Issue 6, pp. 561-570, Feb. 2011.

[22] P. Nithyanandam , T. Ravichandran, E. Priyadharshini and N.M. Santron."A Image Steganography Technique on Spatial Domain Using Matrix and LSB Embedding based on Huffman Encoding", *Journal of Future Engineering and Technology*, vol 6, no.3, Feb 2011.

[23] Chang, C.C and Tseng, H.W. "A Steganographic method for digital images using side match",*Pattern Recognition Letters*, 25: pp.1431 – 1437, Jun. 2004.

[24] Nan-I Wu and Min-Shiang Hwang, "Data Hiding: Current Status and Key Issues", *International Journal of Network Security*, Vol.4, No.1, PP.1–9, Jan. 2007

[25] Nithyanandam, P., Ravichandran, T.,Priyadharshini, E. and Santron, N.M. (2012) "An image steganography for colour images using lossless compression technique", *Int. J. Computational Science and Engineering*, Vol. 7, No. 3,pp.194–205.

[26] P.Nithyanandam, T.Ravichandran, N.M.Santron and E.Priyadarshini.(2011) 'A Spatial Domain Image Steganography Technique based on Matrix Embedding and Huffman Encoding' ,*International Journal of Computer Science and Security*, Volume 5, Issue 5,pp.456-468, Dec.2011.

[27] Chen, P.Y. and Wu, W.E. "A DWT Based Approach for Image Steganography", *International Journal of Applied Science and Engineering*, Volume 4, Issue 3: pp. 275 – 290, 2006.

[28] Abbas Cheddad, Joan Condell, Kevin Curran and Paul Mc Kevitt "Digital Image steganography:Survey and analysis of current methods" *Signal Processing 90* pp. 727-752, 2010.

[29] Abbas Cheddad. "Steganoflage: A New Image Steganography Algorithm",Ph.D. Thesis, School of Computing & Intelligent Systems Faculty of Computing & Engineering University of Ulster, United Kingdom,2009.

[30] Nithyanandam, P., Ravichandran, T. (2012) "Geometrical Rotation Addressed Steganograhy Technique using Matrix Embedding and Huffman Encoding", *European Journal of Scientific Research*, Vol. 87, No. 1,pp.31–45.