# A Spatial Domain Image Steganography Technique Based on Matrix Embedding and Huffman Encoding

**P.Nithyanandam** nithyanandamp@ssn.edu.in
*Department of Computer Application*
*SSN College of Engineering,*
*Anna University of Technology, Chennai*
*Kanchipuram Dt, Tamilnadu , 603110,India*

**T.Ravichandran** *dr.t.ravichandran@gmail.com*
*Principal*
*Hindustan Institute of Technology,*
*Anna University of Technology, Coimbatore*
*Coimbatore Dt,Tamilnadu, 641032,India*

**N.M.Santron** nmsantron@gmail.com
*III Year M.C.A.*
*Department of Computer Application*
*SSN College of Engineering,*
*Anna University of Technology, Chennai*
*Kanchipuram Dt, Tamilnadu , 603110,India*

**E.Priyadharshini** indrapriyadharshini.e@gmail.com
*III Year M.C.A.*
*Department of Computer Application*
*SSN College of Engineering,*
*Anna University of Technology, Chennai*
*Kanchipuram Dt, Tamilnadu , 603110,India*

## Abstract

This paper presents an algorithm in spatial domain which gives less distortion to the cover image during embedding process. Minimizing embedding impact and maximizing embedding capacity are the key factors of any steganography algorithm. Peak Signal to Noise Ratio (PSNR) is the familiar metric used in discriminating the distorted image (stego image) and cover image. Here matrix embedding technique is chosen to embed the secret image which is initially Huffman encoded. The Huffman encoded image is overlaid on the selected bits of all the channels of pixels of cover image through matrix embedding. As a result, the stego image is constructed with very less distortion when compared to the cover image ends up with higher PSNR value. A secret image which cannot be embedded in a normal LSB embedding technique can be overlaid in this proposed technique since the secret image is Huffman encoded. Experimental results for standard cover images, which obtained higher PSNR value during the operation is shown in this paper.

**Keywords:** Steganography, Imperceptibility, Payload, Stego Image, Least Significant Bit (LSB), Huffman Encoding, Matrix Embedding, Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Discrete Wavelet Transformation (DWT).

## 1. INTRODUCTION

Steganography is the art of secret communication. It has apparent difference with cryptography; because cryptography hides information content whereas steganography hides information existence. Steganography is broadly classified in to spatial and frequency domain technique. Least Significant Bit (LSB) replacement, LSB matching, Matrix embedding and Pixel value differencing are some of the spatial domain techniques. Frequency domain techniques include Outguess, F5, JP Hide and Seek. Fundamentally, a steganography algorithm or embedding function can influence the cover work in three different ways, namely cover lookup, cover synthesis and cover modification. Naturally, changes of larger scale will be more obvious than changes of smaller scale. As a result, most steganographic schemes try to minimize the distortion on cover work. The location of changes is controlled by the selection rule [1]. There are three types of rule namely sequential, random and adaptive.

The primary goal of steganography is to design embedding function that should be statistically undetectable and capable of communicating large payloads. There exists a tradeoff between embedding capacity and proportion of distortion. There are many algorithms evolving to accomplish steganography goal in both spatial and frequency domain. Minimizing the embedding impact while constructing a stego image could be one of the ways; this may thwart in applying statistical analysis over a stego image. The notion of this paper is to apply one such embedding technique and to produce a less distorted cover image. Supporting a higher payload on a cover image depends upon embedding technique; but it also can be viewed in another direction of compressing the payload before overlaying. A lossless Huffman [2] [3] [4] [5] compression prior to overlaying results in fewer distortion in the cover image.

Cachin's [1] description of steganography security calls for the Kullback-Leibler distance which says, the probability distance between the cover and stego work to be as little as possible. In our technique it is achieved by minimizing the distortion between the cover and stego work. This will make it harder for the warden to detect embedding. The embedding procedure can encode the message bits in many ways. For example in LSB embedding the LSB is replaced to match the secret message bits. On average, one can embed, 2 bits per embedding change. It can be substantially improved if we adopt a clever embedding scheme. In particular, if the payload is shorter than the embedding capacity, one can influence the location of changes to encode more bits per change. Let us take a look at the following simple example. Say, we have a group of three pixels with gray scale values *x*1, *x*2 and *x*3. We wish to embed 2 message bits, *b*1 and *b2*. It seems that a practical approach might be to simply replace *b*1 with x1 and *b*2 with x2 (i.e.) replacing the LSB of the pixels to match the corresponding message bits. Assuming the 2 bits are 0 or 1 with equal probability, the expected number of changes to the whole group of pixels to embed both bits is 1. Therefore, we embed at embedding efficiency of 2 or 2 bits per change. However, it can be improved. Let us encode *b*1 = *LSB* (*x*1) XOR *LSB* (*x*2) and *b*2 = *LSB* (*x*2) XOR *LSB* (*x*3). If the values of the cover work satisfy both equations with equality, no embedding changes are required. If the first one is satisfied but not the second one, simply flip the LSB of *x*3. If the second one is satisfied but not the first one, flip the LSB of *x*1. If neither one is satisfied, flip LSB of *x*2. Because all four cases are equally likely with probability 1/4, the expected number of changes is 3/4, which is less than what we had earlier. This embedding technique is called *matrix embedding* [1] which is further extended and used in the proposed method.

Huffman compression is a variable length coding whose performance depends on the input image bit stream. The compression is directly proportional to smoothness of the image. Higher the smoothness and higher the redundancy will give good compression. Subjective and objective measures [6] are the two techniques existing to test the distortion of the processed image. Subjective measure is not reliable because human vision is a metric in assessing the distortion of the stego objects. Human vision may vary from person to person; hence this approach is not suitable. In objective measure, the mean square error (MSE) represents the cumulative squared error between the stego image and cover image. A lower figure of MSE conveys lower error/ distortion between the cover and stego image.

The equation of MSE to assess the stego and cover object is given by:

$$MSE = \frac{1}{m*n}\sum_{i-1}^{m}\sum_{j-1}^{n}(A_{ij} - B_{ij})^2 \qquad \text{...........[1]}$$

Whereas $A_{ij}$ 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. It is measured in constant and the unit is decibel (dB).

Peak Signal to Noise Ratio (PSNR) is a metric which calculate the distortion in decibels, between two images. Higher the PSNR indicates a better reconstructed or stego image. The PSNR is represented by the following equation:

$$PSNR = 10*\log_{10}\frac{(Max)^2}{MSE} \qquad \text{..............[2]}$$

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

## 2. RELATED WORK

Chang, C.C et al., [7] has proposed an image steganography technique which offer high embedding capacity and bring less distortion to the stego image. The embedding process embed bits of secret bit stream on the stego image pixels. Instead of replacing the LSB of every pixel, this method replaces the pixel intensity with similar value. The range of modifiable pixel value is higher in edge areas than smooth areas to maintain good perceptual excellence. Various bit embedding methods are followed; which are decided by the correlation between the actual pixel and the neighboring pixels. The neighboring pixels may be a pixel left, right, top or bottom to the actual pixels. The different schemes are two sided, three sided and four sided one. Two sided scheme take upper and left pixels, three side scheme take upper, left and right whereas four sided take upper, left, and right and bottom pixels. The embedding capacity and PSNR are inversely proportional to the sides taken into account.

Po-Yueh Chen et al., [8] proposed an image steganography scheme which fixes the limitation of steganography technique proposed in [7]. The limitation of [7] is falling of boundary problem which means the pixel which is located for embedding will become unused; since it exceeds the maximum intensity level which is greater than 255 (maximum gray scale intensity). Fewer bits are added even on such pixels which improve the embedding capacity without compromising PSNR in this technique.

A. Nag et al., [9] proposed a stenographic technique which is based on wavelet transformation on the images. 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.

R.Amirtharajan et al., [10] proposed a stenographic technique 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 MSE and PSNR is calculated for all channel and the average number of bits get embedded in every pixel is shown in their results.

The rest of the paper is organized as follows. Section III discusses the proposed steganography technique. In Section IV experimental results are exhibited and discussed. Finally the conclusion and future direction are provided for the proposed work.

## 3. PROPOSED METHOD

### 3.1. System Architecture

Fig.1 shows the overall system architecture on which the proposed study stands on. The secret image pixel values are Huffman compressed which comprises of Huffman encodings and Huffman table. The size of Huffman table and Huffman encodings are measured in a 32 bit quantity each. These 64 bits are recorded across the last 64 byte's LSB of the stego image. Both the Huffman encodings and Huffman table binary content are embedded in the LSB of every byte using LSB replacement or Matrix embedding technique. The binary content of Huffman table is followed by Huffman encodings. The starting and the ending point of the corresponding binary component i.e. Huffman encodings or Huffman table is identified through the processed individual 32 bits entry stored at the end of the stego image. In the case of the secret image being sufficiently large, the stego image LSB may be fully utilized. Always, the last 64 byte is reserved for storing the size of Huffman table and Huffman encodings.

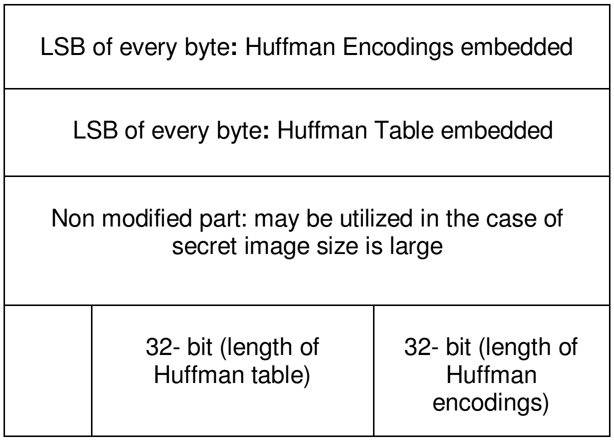
LSB of every byte: Huffman Encodings embedded
LSB of every byte: Huffman Table embedded
Non modified part: may be utilized in the case of secret image size is large
32- bit (length of Huffman table)
32- bit (length of Huffman encodings)

**FIGURE 1:** Stego Image Architecture

### 3.2. Huffman Compression on Image

The intensity of the pixels across the image is spatially correlated [3]. 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. A very frequent occurrence 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. The secret image is Huffman encoded prior to embedding process.

### 3.3. Extended Matrix Embedding

An extended matrix embedding technique is used in proposed method. Generally (1, n, k) matrix embedding [11] mechanism is used; which denotes k secret bits are embedded in n cover bits with at most 1 change. Here using three Least Significant Bits  of RGB channel 2 bits of secret bits might be embedded with at most one change, which is typically (1,3,2) in the above case. Here n is $2^k$-1.

It can be further expanded by considering; more secret bits can be embedded in a single go with at most 1 change. For example if k is 3, then n is $2^k$ -1.  K secret bit should be embedded in $2^k$ -1 cover bit with at most 1 change. It is denoted by (1,7,3), where 1 represent number of changes allowed,7 represent number of cover bit involved in the operation and 3 represent number of secret bit to be embedded. Now the cover bit selection and embedding mechanism to be designed in such a way that, k secret bits should be embedded in n cover bits with at most 1 change.

1) Cover bit Selection: Two types of cover bit selection are attempted in the above proposed technique and the results are shown for both the types.

Method1: In this method the LSB of every byte is chosen as cover bit. 7 bits of data are required to embed a 3 bit secret data. Those 7 bits are collected from seven consecutive bytes of the image. All 7 bytes' LSB is serving as cover bit.

Method2: In this method to collect 7 cover bit for the operation, on every pixel last two bits of red channel, last three bits of green channel and last two bits of blue channel are taken.

2) Secret bit Embedding: In order to embed and extract the 3 secret bit in the 7 cover bit with atmost 1 change, a reversible embedding and extraction algorithm should be designed. Equation 3 shown below will be used to meet the above goal. Assume $b_1$,$b_2$,$b_3$ are the secret bits, $x_1$,$x_2$,$x_3$,$x_4$,$x_5$,$x_6$,$x_7$ are cover bits. The cover bits are adjusted according to the secret bits $b_1$, $b_2$ and $b_3$ with atmost 1 change i.e. only one change is permitted out of all the 7 cover bits. At the same time the secret bit should be mapped inside the cover bit. The following equation is used in both embedding and extraction process.

$$\begin{aligned} b_1 &= x_1 \oplus x_4 \oplus x_6 \oplus x_7 \\ b_2 &= x_2 \oplus x_4 \oplus x_5 \oplus x_7 \\ b_3 &= x_3 \oplus x_5 \oplus x_6 \oplus x_7 \end{aligned} \quad . \qquad \textbf{....................[3]}$$

The above 3 expression in equation 3 is operated to check the coincidence of secret bit against cover bit. An exclusive OR operation is performed on the cover bit; if all the three expression is satisfied no adjustment is required on the cover bit. Sometimes the cover bit by itself, is suitable to fit the secret data. If any or more than one of the expressions in equation 3 is not satisfied then modification on the cover bit is followed according to Table 1. This slight modification on the cover bit enable the secret bit to be mapped on the cover bit with at most only one change. Since the cover bit are adjusted according to the secret bit; during extraction the same equation can be used in recovering the secret bit from the cover bit.

| **Secret bit Positions not matched ($b_1$,$b_2$,$b_3$)** | 1 | 2 | 3 | 1,2 | 2,3 | 1,3 | 1,2,3 |
|---|---|---|---|---|---|---|---|
| **Cover bit to be inverted** | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $x_7$ |

**TABLE 1:** Embedding/ Extraction Reference

Huffman coding technique is used in the proposed method to securely and imperceptibly hide the secret image in the cover image. The Huffman encoded bit stream and Huffman table bit stream is embedded in the cover image pixel either by method1 or method2 through Matrix embedding technique. Cover bit selection will differ in method1 and method2 whereas embedding process remain same.

### 3.4. Hashing

Finally to attain the integrity of the stego image; the secret image is hashed prior to embedding. This hash code should be send as a supplementary component in addition to stego image. In the receiving end, the retrieved secret image is hashed to cross check against the hash code received. If both the hash codes are same, it conveys no intruder has modified the stego image.

### 3.5. Embedding Process

Fig. 2a shows the embedding process carried on the sender side. The Hash code of secret image and stego images are sent to receiver.

The steps carried on the sender side are given below:

Step 1: Hash the secret image.

Step 2: The Secret image is converted into a Numerical matrix which contains the RGB value or intensity of each pixel.

Step 3: Apply Huffman encoding for the output obtained from Step 2 which results in Huffman table and Huffman encoded secret image bit streams.

Step 4: Group the above obtained binary bit stream (Huffman table and Huffman encoded) in chunk of three bits.

Step 5: M1: Method1:- Each color image pixel is represented by 3 bytes (RGB). Collect 7 consecutive bytes from the image. All 7 bytes' LSB is serving as cover bit.

Step 6: M1: Method1:- Using equation 3 adjusts the 7 bytes LSB to match the three secret bit chunk obtained in Step 4. (OR)

Step 5: M2: Method2:- Each color image pixel is represented by 3 bytes (RGB). In this method to collect 7 cover bit for the operation, on every pixel LSB and LSB -1 from Red channel, LSB, LSB -1 and LSB -2 from Green channel, LSB and LSB -1 from Blue channel; a total of 7 bits are chosen as cover bit.

Step 6: M2: Method2:- Using equation 3 adjusts the above 7 bits to match the three bit chunk obtained in Step 4.

Step 7: Repeat Step5 and Step6 until all the 3 secret bit chunks are mapped over the cover image pixels moving from left to right and top to bottom of the cover image.

Step 8: Send the Hash Code and stego image obtained from Step 7 to the receiver.

### 3.6. Extraction Process

Fig. 2b shows the extraction process carried on the receiver side. Upon receiving the stego image, and the Hash code, receiver should extract the Huffman table, Huffman encoded bit streams, and secret image dimension from the stego image.

The steps carried on the receiver side are given below:

Step 1: Apply the relevant bit collection on the stego image pixel depends on the method (method1/method2); the secret bit is embedded in the cover image as explained in embedding process.

Step 2: Size of secret image, Huffman Table and Huffman symbols are retrieved.

Step 3: The Binary Huffman table is then converted to the actual format that can be accepted by the Huffman decoding.

Step 4: The Huffman table and Huffman encodings obtained in Step 2 are used in Huffman decoding process. As a result RGB/intensity value, for every pixel of secret image is obtained.

Step 5: Finally, the image is constructed using all the pixels which is computed in Step 4 will reveal the secret image.

Step 6: To ensure the stego image integrity, the received hash code is compared against the Hash code of constructed secret image. If both are equal, cover image is free from intruder attack.

The intermediate results obtained in every stage of embedding and extraction process are redirected to a text file may be assumed for better understanding of the proposed method wherever required.

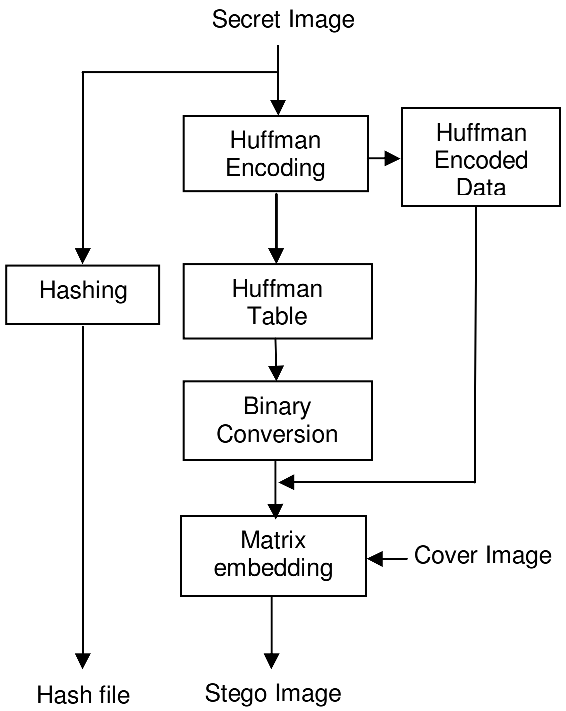

**FIGURE 2a:** Embedding Process

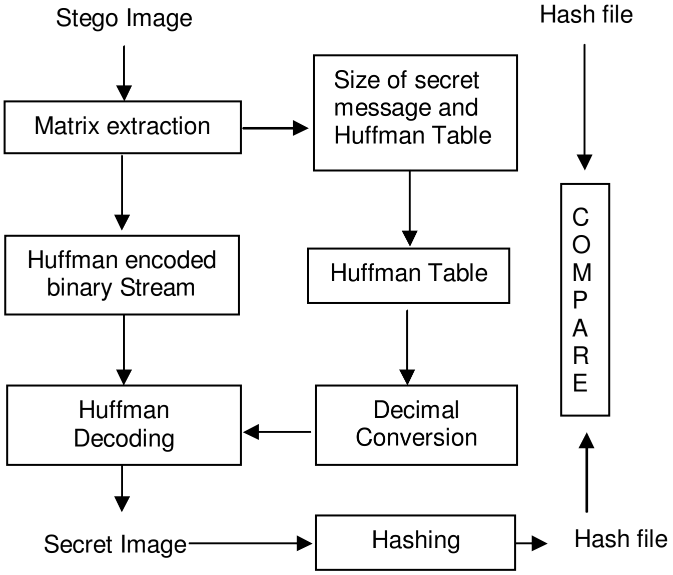

**FIGURE 2b:** Extraction Process

## 4. EXPERIMENTAL RESULTS

Java 2.0 and MATLAB 7.6 are the programming tools used to implement the proposed method. PSNR, Embedding Capacity and Mean Square Error are the three metrics taken here to consolidate the strength of proposed method. PSNR result is shown separately for all the channels. Two tables are used to present the performance of both the methods. The same cover image of size 256 X 256 is used in both the methods. The cover image and secret image taken here for experimentation is 24 bit color depth bmp (Bit Map Format) image.

A secret image Cameraman Fig. 3 of various sizes is embedded in the RGB cover images like Lena, Airplane, Baboon and Boat each of size 256 x 256. Fig. 4-7 shows the cover images, obtained stego images and histogram arrived in method1 and method2 of matrix embedding technique. Table2 and Table3 show the experimental results of method1 and method2 respectively. The PSNR and MSE arrived using the proposed method shows that the distortion occurred in stego image are very less. In method1 secret image of different sizes such as 85x85, 90x90 and 95x95 with 24 bit depth are embedded. The maximum capacity that the cover image can hold is 216,600 bits which is 26.5KB. The embedding capacity is 14% of the cover image using method1. The average PSNR and mean in method1 for 95x95 secret image is 58 and 0.12 respectively.

In method2, since the 7 cover bits are collected on a single pixel, the embedding capacity of the same cover image is better than method1. In method2, the same secret image Cameraman Fig. 3 of different size such as 85x85, 90x90, 95x95, 140x140, 150x150, and 155x155. In method2 a higher capacity is achieved but PSNR and mean is compromised. The maximum capacity that the cover image can hold is 576,600 bits which is 70.38KB. The embedding capacity is 37% of the cover image using method2. The average PSNR and mean in method2 for 155x155 secret image is 50 and 0.6 respectively. The PSNR and mean has declined with an enhanced capacity; but still PSNR value with more than 40 is acceptable.

| Cover Image of size 256 X 256 | | Red Channel | | Green Channel | | Blue Channel | |
|---|---|---|---|---|---|---|---|
| | | PSNR | MSE | PSNR | MSE | PSNR | MSE |
| Lena | 85 x 85 | 57.94 | 0.1044 | 57.90 | 0.1052 | 57.89 | 0.1057 |
| | 90 x 90 | 57.63 | 0.1120 | 57.45 | 0.1169 | 57.51 | 0.1151 |
| | 95x 95 | 57.18 | 0.1243 | 57.22 | 0.1232 | 57.11 | 0.1263 |
| Airplane | 85 x 85 | 57.94 | 0.1044 | 57.89 | 0.1057 | 57.82 | 0.1072 |
| | 90 x 90 | 57.51 | 0.1151 | 57.61 | 0.1125 | 57.46 | 0.1164 |
| | 95x 95 | 57.23 | 0.1227 | 57.12 | 0.1259 | 57.19 | 0.1242 |
| Baboon | 85 x 85 | 57.87 | 0.1061 | 57.93 | 0.1046 | 57.87 | 0.1060 |
| | 90 x 90 | 57.54 | 0.1145 | 57.55 | 0.1141 | 57.49 | 0.1156 |
| | 95x 95 | 57.15 | 0.1252 | 57.22 | 0.1232 | 57.17 | 0.1246 |
| Boat | 85 x 85 | 57.95 | 0.1040 | 57.88 | 0.1059 | 57.82 | 0.1073 |
| | 90 x 90 | 57.56 | 0.1139 | 57.55 | 0.1141 | 57.46 | 0.1167 |
| | 95x 95 | 57.15 | 0.1251 | 57.14 | 0.1254 | 57.14 | 0.1255 |

**TABLE 2:** 7 COVER BIT ON 7 BYTE (METHOD 1)

| Cover Image of size 256 X 256 | | Red Channel | | Green Channel | | Blue Channel | |
|---|---|---|---|---|---|---|---|
| | | PSNR | MSE | PSNR | MSE | PSNR | MSE |
| Lena | 85 x 85 | 54.55 | 0.2277 | 48.40 | 0.9393 | 54.55 | 0.2278 |
| | 90 x 90 | 54.15 | 0.2497 | 47.96 | 1.0397 | 54.30 | 0.2411 |
| | 95x 95 | 53.81 | 0.2700 | 47.72 | 1.0969 | 53.87 | 0.2665 |
| | 140x140 | 51.02 | 0.5131 | 44.77 | 2.1646 | 50.97 | 0.5200 |
| | 150x150 | 50.50 | 0.5783 | 44.17 | 2.4874 | 50.37 | 0.5959 |
| | 155x155 | 50.25 | 0.6131 | 43.96 | 2.6100 | 50.22 | 0.6169 |
| Airplane | 85 x 85 | 54.56 | 0.2275 | 48.40 | 0.9379 | 54.57 | 0.2266 |
| | 90 x 90 | 54.28 | 0.2423 | 47.94 | 1.0445 | 54.20 | 0.2469 |
| | 95x 95 | 53.81 | 0.2700 | 47.57 | 1.1366 | 53.89 | 0.2652 |
| | 140x140 | 50.98 | 0.5180 | 44.80 | 2.1505 | 51.02 | 0.5134 |
| | 150x150 | 50.45 | 0.5856 | 44.16 | 2.4921 | 50.51 | 0.5781 |
| | 155x155 | 50.23 | 0.6155 | 44.04 | 2.5631 | 50.15 | 0.6275 |
| Baboon | 85 x 85 | 54.57 | 0.2267 | 48.34 | 0.9527 | 54.64 | 0.2229 |
| | 90 x 90 | 54.33 | 0.2397 | 47.97 | 1.0374 | 54.12 | 0.2517 |
| | 95x 95 | 53.89 | 0.2652 | 47.70 | 1.1028 | 53.75 | 0.2740 |
| | 140x140 | 51.01 | 0.5145 | 44.73 | 2.1881 | 50.97 | 0.5190 |
| | 150x150 | 50.47 | 0.5824 | 44.24 | 2.4460 | 50.43 | 0.5882 |
| | 155x155 | 50.21 | 0.6191 | 43.98 | 2.6004 | 50.16 | 0.6266 |
| Boat | 85 x 85 | 54.61 | 0.2248 | 48.27 | 0.9673 | 54.56 | 0.2272 |
| | 90 x 90 | 54.24 | 0.2448 | 47.89 | 1.0569 | 54.27 | 0.2427 |
| | 95x 95 | 53.86 | 0.2673 | 47.72 | 1.0984 | 53.81 | 0.2699 |
| | 140x140 | 51.04 | 0.5110 | 44.76 | 2.1710 | 51.01 | 0.5147 |
| | 150x150 | 50.43 | 0.5882 | 44.25 | 2.4384 | 50.48 | 0.5815 |
| | 155x155 | 50.30 | 0.6064 | 43.93 | 2.6298 | 50.23 | 0.6160 |

**TABLE 3:** 7 COVER BIT ON 1 PIXEL – 2, 3,2 (METHOD 2 )

The proposed method's hiding capacity depends upon the Huffman encoding output. The Huffman encoded result of a secret image (Huffman encoded bit stream and Huffman Table) size should be lesser than the total number of LSB spot available in the cover image. 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 image from the stego image. This 64 pixel (64x3=192 bytes) should be excluded while computing the hiding capacity of cover image. Image of any size/richness can be hidden through our proposed method, provided it meets the above said condition. Integrity of the stego image is verified by crosschecking the hash code received against the constructed secret image hash code. If both hash code are same, it conveys no intruder modified the stego image.

### 4.1. Discussion

In method2 the PSNR of green channel is less, compared to the other two channels. It is due to the reason that the cover bits are selected in the same pixel in this order (2, 3, and 2). Two bits from red channel, three bits from green channel and two bits from blue channel are taken. Out of 7 bits, 3 bits are taken from green channel; hence this channel is highly vulnerable to distortion. So, as a result the PSNR of green channel has declined in method2.

We quite often found that a secret image which is richer and whose dimension is lesser than Cameraman,(shown in Fig. 3) say 100 X 100 cannot be embedded in this 256 X 256 cover image shown in figure 4. In contrast, a secret image which is not richer and whose dimension is higher than 100 X 100 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.

The embedding capacity of the cover image can be improved further, if a pixel adjustment process technique is adapted. The number of bits get embedded in the proposed technique is just 3 bit per pixel in method1 or 3 bit using LSB's of seven consecutive bytes in method2. Pixel adjustment process technique is just substituting the intensity of the every cover pixel with an equivalent resembling pixels. This could exploit the cover pixels in embedding greater than 3 bits (9 bits/pixel). But, it will be on the cost of compromising a little bit distortion gets introduced on the cover image.

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. Stego image integrity is validated through hashing which give confidence to the receiver. Thus, the privacy and security issues are addressed in this proposed technique to a reasonable extent.

## CONCLUSION

We had proposed an image steganography algorithm which brings a better PSNR and MSE. The experimental results show that distortion between cover and stego image is minimum. Capacity improvement and distortion reduction has been addressed in this proposed technique. In the proposed method, the embedding capacity of the cover image is increased which results in slight decline in both PSNR and MSE parameters. The veracity of the stego image is verified and then progressed for their usage on receiver side. The proposed technique is not robust against any geometrical distortion such as rotation, translation, scaling, cropping etc., induced on the stego image. Improving this parameter is still under research and not matured yet.

## FUTURE WORK

The proposed algorithm should be customized to support embedding in the frequency domain. It should be enhanced to withstand geometrical distortion induced on the image.

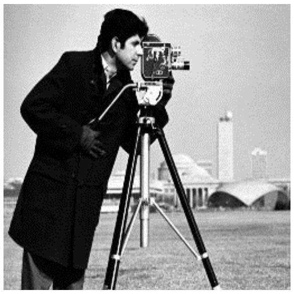

**FIGURE 3:** Cameraman

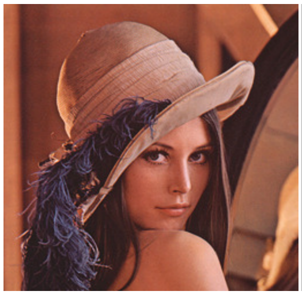

**FIGURE 4a:** Lena Cover

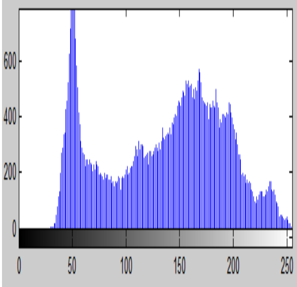

**FIGURE 4b:** Red Channel

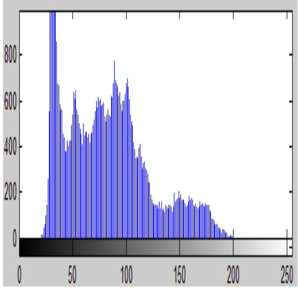

**FIGURE 4c:** Green Channel

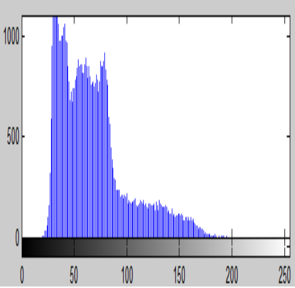

**FIGURE 4d:** Blue Channel

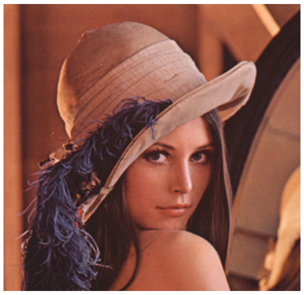

**FIGURE 4e:** Lena Stego M1

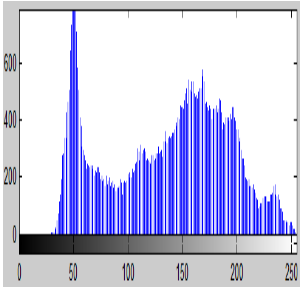

**FIGURE 4f:** Red Channel

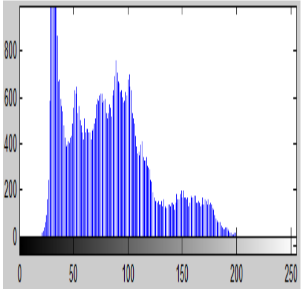

**FIGURE 4g:** Green Channel

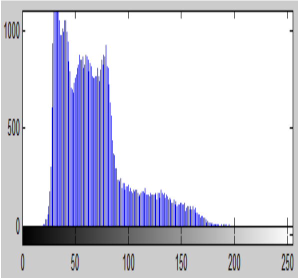

**FIGURE 4h:** Blue Channel

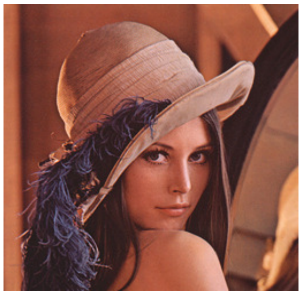

**FIGURE 4i**: Lena Stego M2

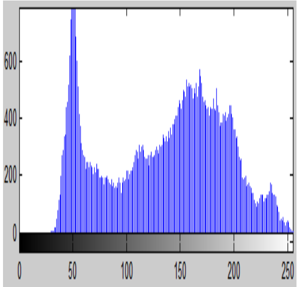

**FIGURE 4j:** Red Channel

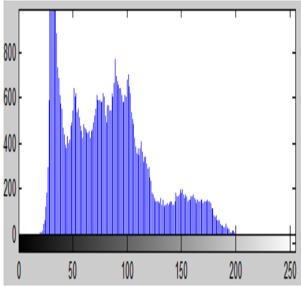

**FIGURE 4k:** Green Channel

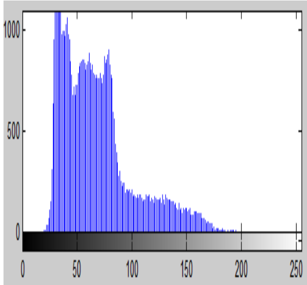

**FIGURE 4l:** Blue Channel

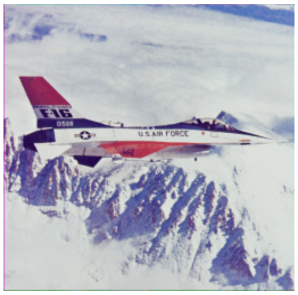

**FIGURE 5a:** Airplane Cover

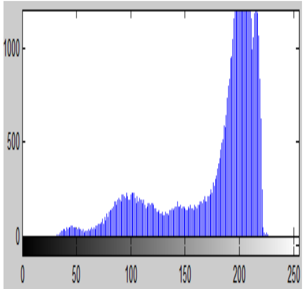

**FIGURE 5b:** Red Channel

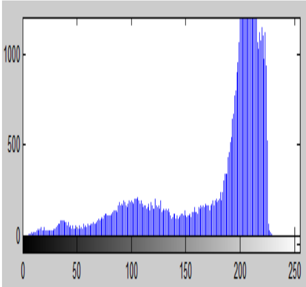

**FIGURE 5c:** Green Channel

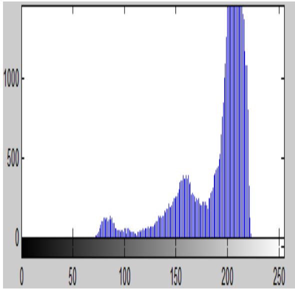

**FIGURE 5d:** Blue Channel

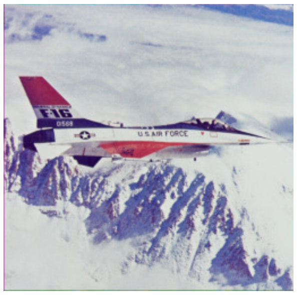

**FIGURE** 5e: Airplane Stego M1

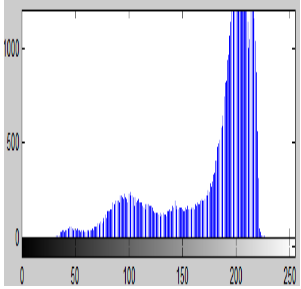

**FIGURE** 5f: Red Channel

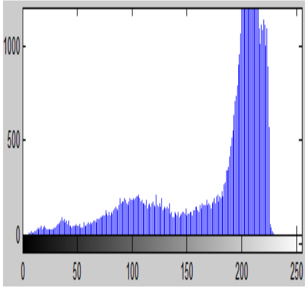

**FIGURE 5g:** Green Channel

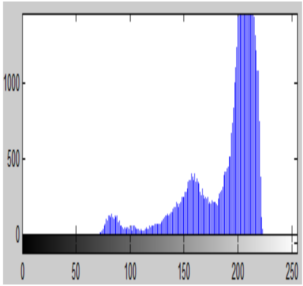

**FIGURE 5h:** Blue Channel

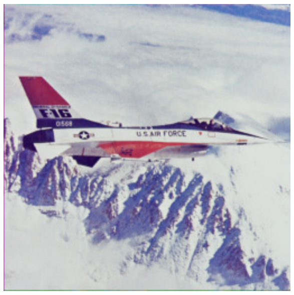

**FIGURE 5i:** Airplane Stego M2

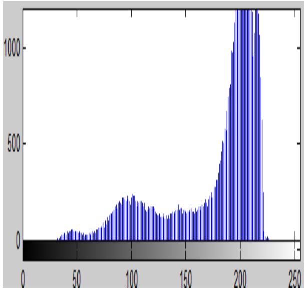

**FIGURE 5j:** Red Channel

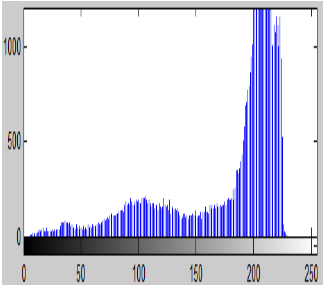

**FIGURE 5k:** Green Channel

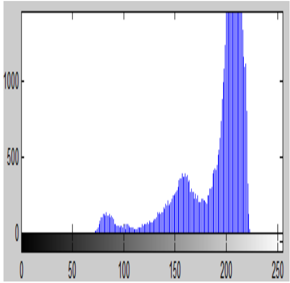

**FIGURE 5l:** Blue Channel

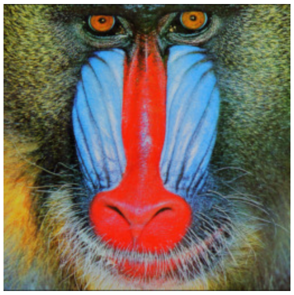

**FIGURE 6a:** Baboon Cover

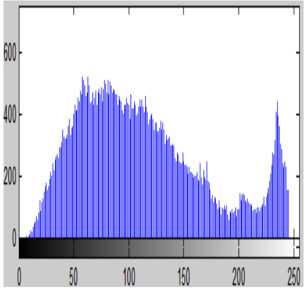

**FIGURE 6b:** Red Channel

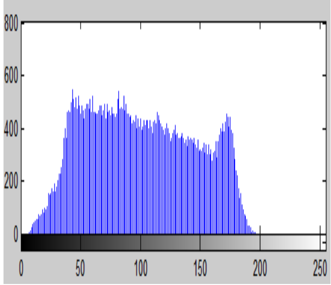

**FIGURE 6c:** Green Channel

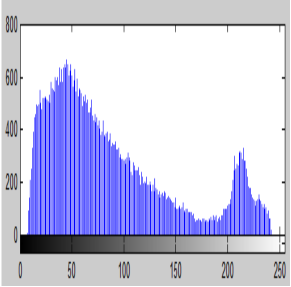

**FIGURE 6d:** Blue Channel

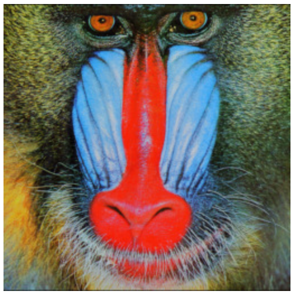

**FIGURE 6e:** Baboon Stego M1

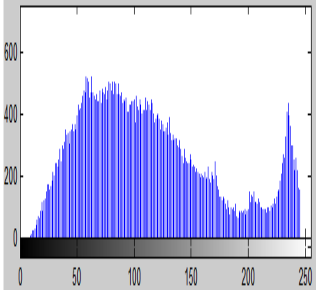

**FIGURE 6f:** Red Channel

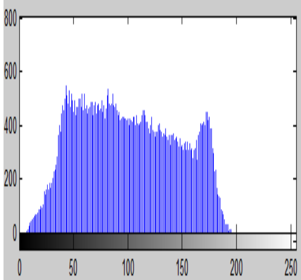

**FIGURE 6g:** Green Channel

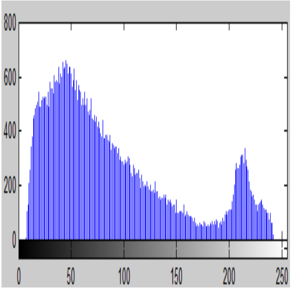

**FIGURE 6h:** Blue Channel

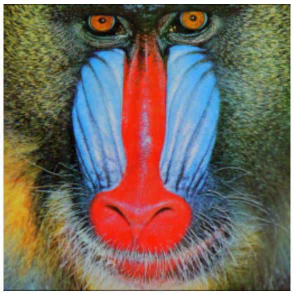

**FIGURE 6i:** Baboon Stego M2

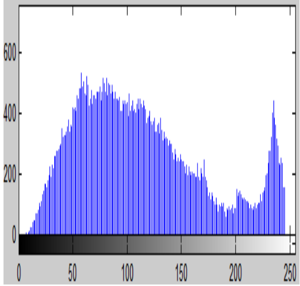

**FIGURE 6j:** Red Channel

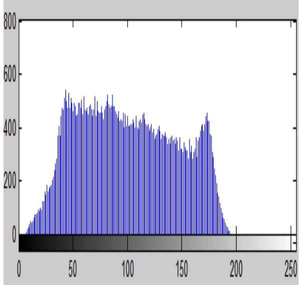

**FIGURE 6k:** Green Channel

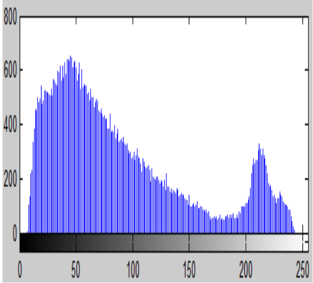

**FIGURE 6l:** Blue Channel

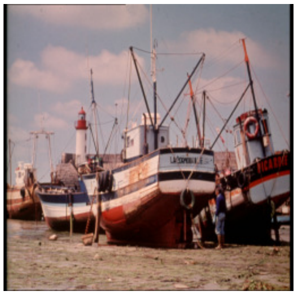

**FIGURE** 7a: Boat Cover

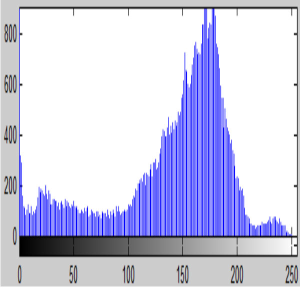

**FIGURE** 7b: Red Channel

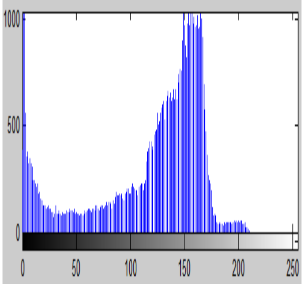

**FIGURE** 7c: Green Channel

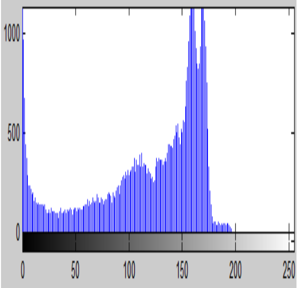

**FIGURE** 7d: Blue Channel

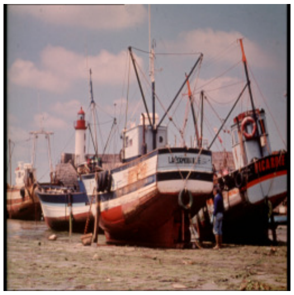

**FIGURE 7e:** Boat Stego M1

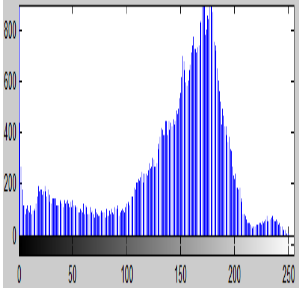

**FIGURE 7f:** Red Channel

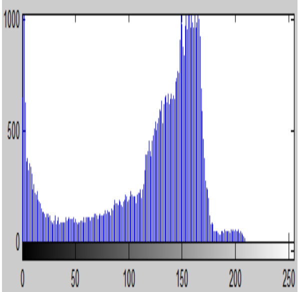

**FIGURE 7g:** Green Channel

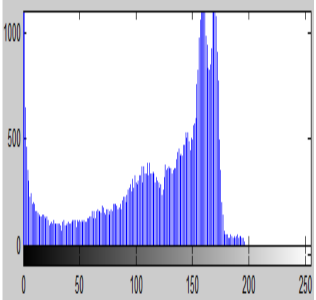

**FIGURE 7h**: Blue Channel

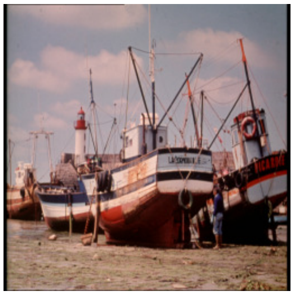

**FIGURE 7i:** Boat Stego M2

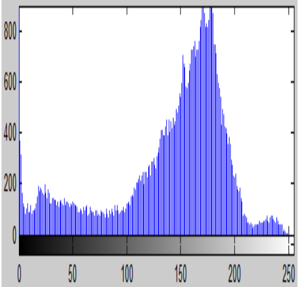

**FIGURE 7j:** Red Channel

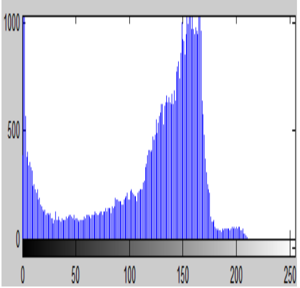

**FIGURE 7k:** Green Channel

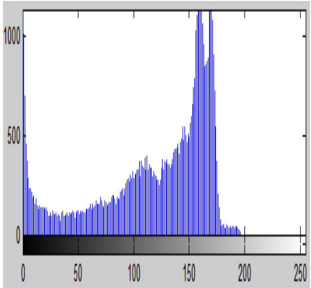

**FIGURE 7l:** Blue Channel

## REFERENCES

[1] Injemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, Ton Kalker. *Digital Watermarking and Steganography.* Morgan Kaufmann, Second Edition,2008.

[2] Professor Luca Trevisan, 2001, "*Lecture notes on Intro. To CS Theory.*" Online. Available: http:// ww.cs.berkeley.edu/~luca/cs170/notes/lecture15.ps

[3] Rafael C. Gonzalez Richard E. Woods. *Digital Image Processing.* ,PHI, Third Edition, 2008.

[4] Alasdair McAndrew. Introduction *to Digital Image Processing with MATLAB,* CENGAGE Learning, 2004

[5] [Online].Available : http://www.binaryessence.com

[6] Ali K. Hmood, Z. M. Kasirun, Hamid A. Jalab,Gaz Mahabubul Alam, A. A. Zaidan, B. B. Zaidan. "On the accuracy of hiding information metrics: Counterfeit protection for education and important certificates." *International Journal of the Physical Sciences*, Volume. 5, Issue 7, pp. 1054-1062, August,2010.

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

[8] Po-Yueh Chen,Wei-En Wu. "A Modified Side Match Scheme for Image Steganography" , *International Journal of Applied Science and Engineering,* Volume 7, Issue 1, pp. 53-60, October 2009.

[9] A. Nag, S. Biwa's, 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.

[10] R.Amirtharajan, 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.

[11] Santosh Arjun, Atul Negi, Chaitanya Kranti, and Divya Keerthi. "An Approach to Adaptive Steganography Based on Matrix Embedding" *TENCON 2007 - 2007 IEEE Region 10* , Volume 1,pp.1-4, Oct . 2007.