International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.3, No.6, June 2014 DOI:10.15693/ijaist/2014.v3i6.52-57

Image Steganography Using Edge Adaptive technique with LSB Matching

J. R. Kharche
Lecturer/Department of E&TC
Pune University, Pune, India

V.N. Ghodke
Assistant Professor/Department of E&TC
Pune University, Pune

***Abstract*—The least-significant-bit (LSB)-based approach is a popular type of steganographic algorithms in the spatial domain. It is shown that the length of hidden messages embedded in the least significant bits of signal samples can be estimated with relatively high precision. The new steganalytic approach is based on some statistical measures of sample pairs that are highly sensitive to LSB embedding operations. The choice of embedding positions within a cover image mainly depends on a pseudorandom number generator without considering the relationship between the image content itself and the size of the secret message. Thus the smooth/flat regions in the cover images will inevitably be contaminated after data hiding even at a low embedding rate, and this will lead to poor visual quality and low security based on our analysis. In this paper, we expand the LSB matching revisited image steganography and propose an edge adaptive scheme which can select the embedding regions according to the size of secret message and the difference between two consecutive pixels in the cover image. For lower embedding rates, only sharper edge regions are used while keeping the other smoother regions as they are. When the embedding rate increases, more edge regions can be released adaptively for data hiding by adjusting just a few parameters. The experimental results evaluated on 6000 natural images show that the new scheme can enhance the security, while preserving higher visual quality of stego images at the same time.**

***Keywords*- LSB, LSBMR, Secret, Message,**

I. INTRODUCTION

Steganography is a technique for information hiding. It aims to embed secret data into a digital cover media, such as digital audio, image, video, etc., without being suspicious. On the other side, steganalysis aims to expose the presence of hidden secret messages in those stego media. If there exists a steganalytic algorithm which can guess whether a given media is a cover or not with a higher probability than random guessing, the steganographic system is considered broken. In practice, two properties, undetectability and embedding capacity, should be carefully considered when designing a steganographic algorithm. Hiding information by embedding secret data into an innocuous medium is often referred to as steganography. Steganography can be applied electronically by taking a message (a binary file) and some sort of cover (image file) and combining both to obtain a “stego-object”. The stego-object is essentially the cover with its redundant information replaced with the message. Unfortunately, replacing the redundant information with structured message data often results in the stego-object being an augmented version of the cover “marked” by the hidden data and this makes it easy to suspect the presence of steganography. Most of these marks can be attributed to the hiding algorithm’s lack of concern for the cover’s content. If the cover’s original content were taken into account then the message could be concealed in areas of the cover where it would be less likely to leave marks. Previous attempts at achieving this goal have required the user to provide the original cover as well as the stego-object. The best areas to hide are first identified in the original cover, then these areas are mapped across to the stego-object and the hidden information is retrieved. The original cover must be provided because the information overwritten in the message hiding process may have been used to identify the best hiding areas. However, to provide the original object is not secure, because taking the differences between the two objects would be enough to suspect the existence of (and in some cases, recover) the hidden information. This paper investigates an approach that eliminates the need for providing the original object. We use images as a cover medium and introduce new algorithms to determine effective hiding places.

II. RELATED WORK

Digital image is defined as a two dimensional function f(x, y), where x and y are spatial (plane) coordinates, and the amplitude of f at any pair of coordinates (x, y) is called intensity or grey level of the image at that point. The field of digital image processing refers to processing digital images by means of a digital computer. The digital image is composed of a finite number of elements, each of which has a particular location and value. The elements are referred to as picture elements, image elements, pels, and pixels.

1. Image Compression

Digital Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is removal of redundant data. From the mathematical viewpoint, this amounts to transforming a 2D pixel array into a statically uncorrelated data set. The data redundancy is not an abstract concept but a mathematically quantifiable entity. If n1 and n2 denote the number of information-carrying units in two data sets that represent the same information, the relative data redundancy R_D [2] of the first data set (the one characterized by n1) can be defined as,

$$R_D = 1 - \frac{1}{C_R}$$

Where C_R called as compression ratio [2]. It is defined as

$$C_R = \frac{n1}{n2}$$

In image compression, three basic data redundancies can be identified and exploited: Coding redundancy, interpixel redundancy, and phychovisal redundancy. Image compression is achieved when one or more of these redundancies are reduced or eliminated.

The image compression is mainly used for image transmission and storage. Image transmission applications are in broadcast television, remote sensing via satellite, air-craft, radar, or sonar, teleconferencing, computer communications & facsimile transmission. Image storage is required most commonly for educational and business documents, medical images that arise in computer tomography (CT), magnetic resonance imaging (MRI) and digital radiology, motion pictures, satellite images, weather maps, geological surveys, and so on.

Image Compression Types

A. Lossy Image Compression:

Lossy compression provides higher levels of data reduction but result in a less than perfect reproduction of the original image. It provides high compression ratio. Lossy image compression is useful in applications such as broadcast television, videoconferencing & facsimile transmission, in which a certain amount of error is an acceptable trade-off for increased compression performance.

Originally, PGF has been designed to quickly and progressively decode lossy compressed aerial images. A lossy compression mode has been preferred, because in an application like a terrain explorer texture data (e.g., aerial orthophotos) is usually mid-mapped filtered and therefore lossy mapped onto the terrain surface. In addition, decoding lossy compressed images is usually faster than decoding lossless compressed images.

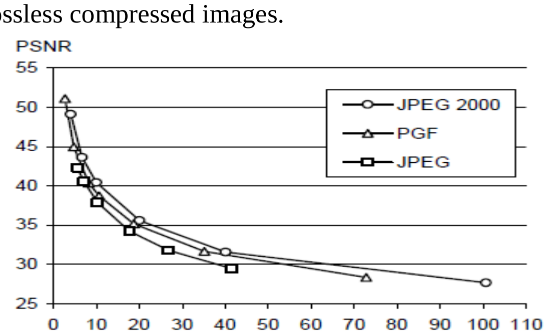

Fig.1. PSNR of lossy compression in relation to compression ratio

In the next test set we evaluate the lossy compression efficiency of PGF. One of the best competitors in this area is for sure JPEG 2000. Since JPEG 2000 has two different filters, we used the one with the better trade-off between compression efficiency and runtime. On our machine the 5/3 filter set has a better trade-off than the other. However, JPEG 2000 has in both cases remarkable good compression efficiency for very high compression ratios but also a very poor encoding and decoding speed.

The other competitor is JPEG. JPEG is one of the most popular image file formats. It is very fast and has reasonably good compression efficiency for a wide range of compression ratios. The drawbacks of JPEG are the missing lossless compression and the often missing progressive decoding. Fig. 1 depicts the average rate-distortion behavior for the images in the Kodak test set when fixed (i.e., nonprogressive) lossy compression is used. The PSNR of PGF is on average 3% smaller than the PSNR of JPEG 2000, but 3% better than JPEG. These results are also qualitative valid for our PGF test set and they are characteristic for aerial ortho-photos and natural images. Because of the design of PGF we already know that PGF does not reach the compression efficiency of JPEG 2000. However, we are interested in the trade-off between compression efficiency and runtime. To report this trade-off we show in Table 1 a comparison between JPEG 2000 and PGF and in Fig. 1.we show for the same test series as in Fig. 2 the corresponding average decoding times in relation to compression ratios.

Table 1 contains for seven different compression ratios (mean values over the compression ratios of the eight images of the Kodak test set) the corresponding average encoding and decoding times in relation to the average PSNR values. In case of PGF the encoding time is always slightly longer than the corresponding decoding time. The reason for that is that the actual encoding phase takes slightly longer than the corresponding decoding phase.

For six of seven ratios the PSNR difference between JPEG 2000 and PGF is within 3% of the PSNR of JPEG 2000. Only in the first row is the difference larger (21%), but because a PSNR of 50 corresponds to an almost perfect image quality the large PSNR difference corresponds with an almost undiscoverable visual difference. The price they pay in JPEG 2000 for the 3% more PSNR is very high. The creation of a PGF is five to twenty times faster than the creation of a corresponding JPEG 2000 file, and the decoding of the created PGF is still five to ten times faster than the decoding of the JPEG 2000 file. This gain in speed is remarkable, especially in areas where time is more important than quality, maybe for instance in real-time computation.

	JPEG 2000 5/3			
Ratio	**Encoding**	**Decoding**	**PSNR**	**Enc**
2.7	**1.86**	**1.35**	**64.07**	**0.34**
4.8	**1.75**	**1.14**	**47.08**	**0.27**
8.3	**1.68**	**1.02**	**41.98**	**0.22**
10.7	**1.68**	**0.98**	**39.95**	**0.14**
18.7	**1.61**	**0.92**	**36.05**	**0.12**
35.1	**1.57**	**0.85**	**28.86**	**0.10**
72.9	**1.54**	**0.85**	**28.86**	**0.08**

Table 1. Trade-off between quality and speed for the Kodak test set

In Fig.1 we see that the price we pay in PGF for the 3% more PSNR than JPEG is low: for small compression ratios (< 9) decoding in PGF takes two times longer than JPEG and for higher compression ratios (> 30) it takes only ten percent longer than JPEG. These test results are characteristic for both natural images and aerial ortho-photos. Again, in the third test series we only use the 'Lena' image. We run our lossy coder with six different quantization parameters and measure the PSNR in relation to the resulting compression ratios. The results (ratio:PSNR) is shown in fig. 1.

B. Lossless Image compression:

Lossless Image compression is the only acceptable amount of data reduction. It provides low compression ratio while compared to lossy. In Lossless Image compression techniques are composed of two relatively independent operations: (1) devising an alternative representation of the image in which its interpixel redundancies are reduced and (2) coding the representation to eliminate coding redundancies.

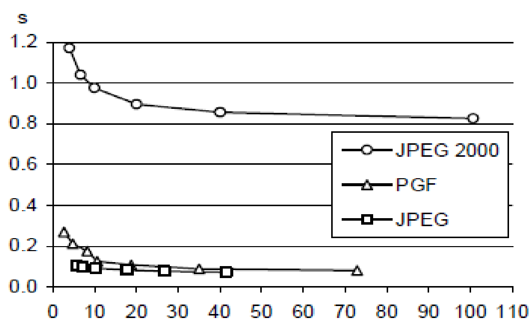

Fig. 2 Decoding time in relation to compression ratio

Lossless Image compression is useful in applications such as medical imaginary, business documents and satellite images.

Table 2 summarizes the lossless compression efficiency and Table 3 the coding times of the PGF test set. For WinZip we only provide average runtime values, because of missing source code we have to use an interactive testing procedure with runtimes measured by hand. All other values are measured in batch mode.

	WinZip	JPEG-LS	JPEG 2000
Aerial	**1.352**	**2.073**	**2.383**
Compound	**12.451**	**6.802**	**6.068**
Hibiscus	**1.816**	**2.200**	**2.822**
Houses	**1.241**	**1.518**	**2.155**
Logo	**47.128**	**16.280**	**12.959**
Redbrush	**2.433**	**4.041**	**4.494**
Woman	**1.577**	**1.920**	**2.564**
Average	**9.71**	**4.98**	**4.78**

Table 2: Lossless compression ratios of the PGF test set

In Table 2 it can be seen that in almost all cases the best compression ratio is obtained by JPEG 2000, followed by PGF, JPEG-LS, and PNG. This result is different to the result in [SEA+00], where the best performance for a similar test set has been reported for JPEG-LS. PGF performs between 0.5% (woman) and 21.3% (logo) worse than JPEG 2000. On average it is almost 15% worse. The two exceptions to the general trend are the 'compound' and the 'logo' images. Both images contain for the most part black text on a white background. For this type of images, JPEG-LS and in particular WinZip and PNG provide much larger compression ratios. However, in average PNG performs the best, which is also reported in [SEA+00].

These results show, that as far as lossless compression is concerned, PGF performs reasonably well on natural and aerial images. In specific types of images such as 'compound' and 'logo' PGF is outperformed by far in PNG.

WinZip and PNG decode even more faster than JPEG-LS, but their encoding times are also worse. PGF seems to be the best compromise between encoding and decoding times. Our PGF test set clearly shows that PGF in lossless mode is best suited for natural images and aerial orthophotos. PGF is the only algorithm that encodes the three MByte large aerial ortho-photo in less than second without a real loss of compression efficiency. For this particular image the efficiency loss is less than three percent compared to the best. These results should be underlined with our second test set, the Kodak test set.

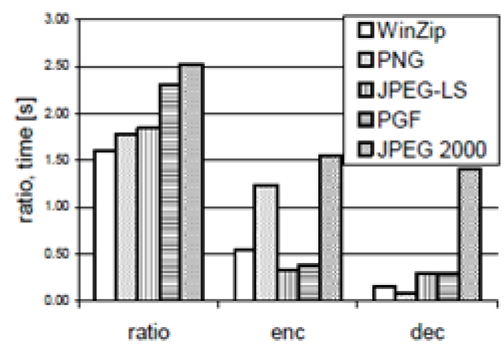

Fig. 3: Lossless compression of the Kodak test set

Fig. 3 shows the averages of the compression ratios (ratio), encoding (enc), and decoding (dec) times over all eight images. JPEG 2000 shows in this test set the best compression efficiency followed by PGF, JPEG-LS, PNG, and WinZip. In average PGF is eight percent worse than JPEG 2000. The fact that JPEG 2000 has a better lossless compression ratio than PGF does not surprise, because JPEG 2000 is more quality driven than PGF.

However, it is remarkable that PGF is clearly better than JPEG-LS (+21%) and PNG (+23%) for natural images. JPEG-LS shows in the Kodak test set also a symmetric encoding and decoding time behavior. Its encoding and decoding times are almost equal to PGF. Only PNG and WinZip can faster decode than PGF, but they also take longer than PGF to encode.

If both compression efficiency and runtime is important, then PGF is clearly the best of the tested algorithms for lossless compression of natural images and aerial orthophotos. In the third test we perform our lossless coder on the 'Lena' image. The compression ratio is 1.68 and the encoding and decoding takes 0.25 and 0.19 seconds, respectively.

1.4.4. Image Compression Standards

There are many methods available for lossy and lossless, image compression. The efficiency of these coding standardized by some Organizations. The International Standardization Organization (ISO) and Consultative Committee of the International Telephone and Telegraph (CCITT) are defined the image compression standards for both binary and continuous tone (monochrome and Colour) images. Some of the Image Compression Standards are

1. JBIG1
2. JBIG2
3. JPEG-LS
4. DCT based JPEG
5. Wavelet based JPEG2000

Currently, JPEG2000 [3] is widely used because; the JPEG-2000 standard supports lossy and lossless compression of single-component (e.g., grayscale) and multicomponent (e.g., color) imagery. In addition to this basic compression functionality, however, numerous other features are provided, including: 1. Progressive recovery of an image by fidelity or resolution; 2. Region of interest coding, whereby different parts of an image can be coded with differing fidelity; 3. Random access to particular regions of an image without the needed to decode the entire code stream; 4. A flexible file format with provisions for specifying opacity information and image sequences; and 5. Good error resilience. Due to its excellent coding performance and many attractive features, JPEG 2000 has a very large potential application base. Some possible application areas include: image archiving, Internet, web browsing, document imaging, digital photography, medical imaging, remote sensing, and desktop publishing.

III. DATA HIDING BY SIMPLE LSB SUBSTITUTION:

In this section, the general operations of data hiding by simple LSB substitution method are described.

Let C be the original 8-bit grayscale cover-image of Mc × Nc pixels represented as

$$C = \{x_{ij} | 0 \leqslant i < M_c, 0 \leqslant j < N_c,$$
$$x_{ij} \in \{0, 1, \ldots, 255\}\}.$$

M be the n-bit secret message represented as

$$M = \{m_i | 0 \leqslant i < n, m_i \in \{0, 1\}\}.$$

Suppose that the n-bit secret message M is to be embedded into the k-rightmost LSBs of the cover-image C. Firstly, the secret message M is rearranged to form a conceptually k-bit virtual image M_ represented as

$$M' = \{m'_i | 0 \leqslant i < n', m'_i \in \{0, 1, \ldots, 2^k - 1\}\},$$

Where $n' < M_c \times N_c$.

The mapping between the n-bit secret message.

$M = \{m_i\}$ And the embedded message $M' = \{m'_i\}$ can be defined as follows:

$$m'_i = \sum_{j=0}^{k-1} m_{i \times k + j} \times 2^{k-1-j}.$$

Secondly, a subset of n_ pixels {xl1; xl2 ,…….. xln_ } is chosen from the cover-image C in a predefined sequence. The embedding process is completed by replacing the k LSBs of xli by mi' . Mathematically, the pixel value xli of the chosen pixel for storing the k-bit message mi' is modified to form the stego-pixel xli' as follows:

$$x'_{l_i} = x_{l_i} - x_{l_i} \bmod 2^k + m'_i.$$

In the extraction process, given the stego-image S, the embedded messages can be readily extracted without referring to the original cover-image. Using the same sequence as in the embedding process, the set of pixels{xl1', xl2',…. Xln'} storing the secret message bits are selected from the stego-image. The k LSBs of the selected pixels are extracted and lined up to reconstruct the secret message bits. Mathematically, the embedded message bits mi' can be recovered by

$$m'_i = x'_{l_i} \bmod 2^k.$$

Suppose that all the pixels in the cover-image are used for the embedding of secret message by the simple LSB substitution method. Theoretically, in the worst case, the PSNR of the obtained stego-image can be computed by

$$\mathrm{PSNR}_{worst} = 10 \times \log_{10} \frac{255^2}{\mathrm{WMSE}}$$
$$= 10 \times \log_{10} \frac{255^2}{(2^k - 1)^2} \text{ dB}.$$

Table 1 tabulates the worst PSNR for some k = 1–5. It could be seen that the image quality of the stego-image is degraded drastically when k>4.

Case 4 $(-2^k < \delta_i < -2^{k-1}$ and $p'_i < 256 - 2^k)$

$$\delta'_i = p''_i - p_i = p'_i + 2^k - p_i = \delta_i + 2^k$$
$$\Rightarrow -2^k + 2^k < \delta'_i < -2^{k-1} + 2^k$$
$$\Rightarrow 0 < \delta'_i < 2^{k-1}.$$

Case 5 $(-2^k < \delta_i < -2^{k-1}$ and $p'_i \geqslant 256 - 2^k)$

$$\delta'_i = p''_i - p_i = p'_i - p_i = \delta_i$$
$$\Rightarrow -2^k < \delta'_i < -2^{k-1}.$$

IV. EXPERIMENTAL RESULT

We can use uncompressed color image as input and after converting it in grayscale image we apply our algorithm on it. Then find out high frequency component pixels which show the edge areas of images for hiding the text data in it.

One of the important properties of our steganographic method is that it can first choose the sharper edge regions for data hiding according to the size of the secret message by adjusting a threshold T. When T is 0, all the embedding units within the cover become available. In such a case, our method can achieve the maximum embedding capacity of 100% (100% means 1 bpp on average for all the methods in this paper).It can also be observed that most secret bits are hidden within the edge regions when the embedding rate is low, while keeping those smooth regions such as they are. The subjective quality of our stegos would be improved & based on the human visual system (HVS) characteristics.

Now we will see all the stages according with the results. In that first step is Histogram Modifications. The histogram of an image represents the relative frequency of occurrence of the various gray levels in the image. Modify an image so that its histogram has desired shape. In histogram equalization, the goal is to obtain a uniform histogram for the output image. The input gray level is first transformed nonlinearly and output is uniformly quantized.

Fig 4. : LENA Image in JPEG-2000 LS

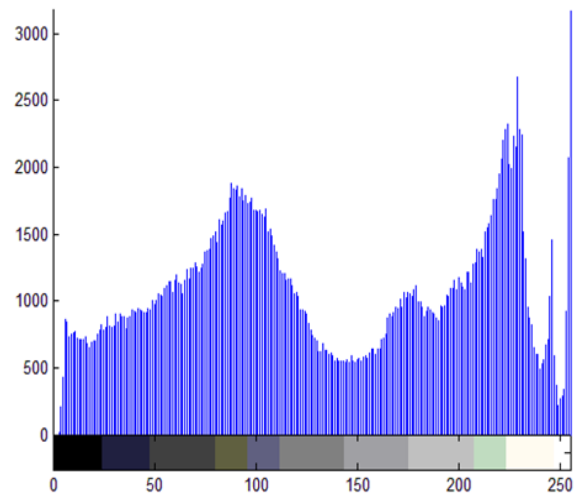

Fig 5. : Histogram of LENA Image

Fig.6 : After Histogram Modification

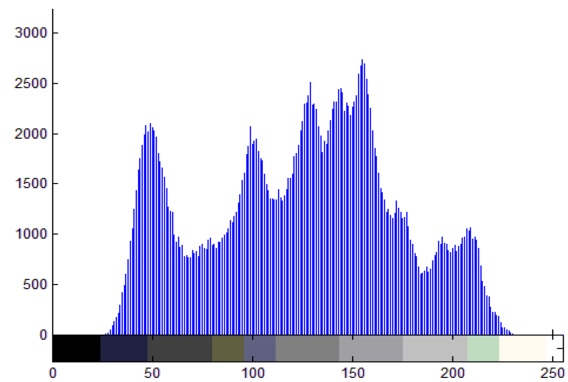

Fig.7 : Histogram of modified LENA Image.

Fig.8: After embedding the secret message into image

V. CONCLUSION

In this paper we proposed a novel data hiding scheme that hides data into an image. The system combines an adaptive data hiding technique to increase the hiding capacity of the system compared to other systems. The proposed system embeds secret data in a random order using a secret key only known to both sender and receiver.

It is an adaptive system which embeds different number of bits in each coefficient according to a hiding capacity function in order to maximize the hiding capacity without sacrificing the visual quality of resulting stego image. The proposed system also minimizes the difference between original coefficients values and modified values.

The proposed scheme was classified into three cases of hiding capacity according to different applications required by the user. Each case has different visual quality of the stego-image. Any data type can be used as the secret message since our experiments was made on a binary stream of data. There was no error in the recovered message (perfect recovery) at any hiding rate. From the experiments and the obtained results the proposed system proved to achieve high hiding capacity up to 48% of the cover image size with reasonable image quality and high security because of using random insertion of the secret message. On the other hand the system suffers from low robustness against various attacks such as histogram equalization and JPEG compression.

The proposed system can be further developed to increase its robustness by using some sort of error correction code which increases the probability of retrieving the message after attacks, also investigating methods to increase visual quality of the stego-image (PSNR) with the obtained hiding capacity.

VI. REFERENCES

[1] G. J. Simmons, "The prisoners' problern and the subliminal channel," in Proceedings of Crypto' 83, pp. 51-67, 1984.

[2] N. Wu and M. Hwang. "Data Hiding: Current Status and Key Issues,"International Journal of Network Security, Vol.4, No.1, pp. 1-9, Jan. 2007.

[3] W. Chen, "A Comparative Study of Information Hiding Schernes Using Amplitude, Frequency and Phase Embedding," PhD Thesis, National Cheng Kung University, Tainan, Taiwan, May 2003.

[4] C. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognition, pp. 469-474, Mar. 2004.

[5] Changa, C. Changa, P. S. Huangb, and T. Tua, "A Novel bnage Steganographic Method Using Tri-way Pixel-Value Differencing," Journal of Multimedia, Vol. 3, No.2, June 2008.

[6] H. H. Zayed, "A High-Hiding Capacity Technique for Hiding Data in images Based on K-Bit LSB Substitution," The 30th International Conference on Artificial Intelligence Applications (ICAIA – 2005) Cairo, Feb. 2005.

[7] A. Westfeld, "F5a steganographic algorithm: High capacity despite better steganalysis," 4th International Workshop on Information Hiding, pp.289-302, April 25-27, 2001.

Authors Profile

J.R.Kharche received the **B.E.** degree in electronics and Telecommunication engineering from the V.Y.W.S College of Engineering, Amravati University, Amravati, India, in 2004.Currently doing **M.E.** in VLSI & Embedded System) in G.S.M.College of Engineering,Pune University,Pune, India. S he is working as Head of Department in Abhinav Education Society,COET(Polytechnic) from last 5 years.Her research interest includes Image Processing, Advance communication System.

Venkat.N.Ghodke is Assistant professor in Electronics and Telecommunication Department of AISSMS'S Institute of information technology Pune .He has worked in various institute as UG and PG guide for image and embedded system design related area. He has published books and papers in journals