An Adaptive Data Hiding Method Using Neighborhood Pixels Differencing Based On Modulus Function

Najme Maleki , Mehrdad Jalali , Majid Vafaei Jahan

Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran

Abstract- *This paper presents an adaptive data hiding method based on four-pixel differencing and modulus function. In our scheme, the average differencing value of a four-pixel block via a threshold secret key determines whether current block is located in edge or smooth areas. Pixels in the edge areas are embedded by Q-bit of secret data with a larger value of Q than that of pixels placed in smooth areas. The proposed scheme presents several advantages.1-the embedding capacity is scalable, 2-the high embedding capacity with minimal visual distortion can be achieved, 3-our method requires little memory space for secret data embedding and extracting phases, 4-five secret keys are used to protect of the embedded secret data, 5-the problem of overflow or underflow does not occur. Experimental results indicated that the proposed adaptive scheme significantly is superior to the currently existing scheme, in term of stego-image visual quality, embedding capacity and level of security.*

Keywords: data hiding, average differencing value, steganography, modulus function, embedding capacity, imperceptible.

1 Introduction

Nowadays message transmission on the internet still has to face some problems such as data security, copyright control, etc. Therefore, we need secret communication schemes for transmitting message on the internet. Encryption may provide a safe way, which transforms data into a cipher-text via cipher algorithms. However, encryption makes the message unreadable, but making message suspicious enough to attract eavesdropper's attention. To overcome this problem, via data hiding techniques can hide the secret data behind a cover media such as text, image, audio and video, and the result does not attract any special attention. Steganography is a data hiding technique. In the past decade, steganography in image extremely has been studied. The image into which a massage is hidden is called a cover image and the result a stego-image. Application of the data hiding can be used in military, commercial and anti-criminal depended application [5]. A well-known steganographic method is the Least Significant Bit (LSB) substitution, which embeds secret data by replacing k LSBs of a pixel with k secret bits directly [10] which the embedding capacity in each pixel of cover image is a fixed value e.g. a non-adaptive method without taking the image local texture into consideration. However, not all Pixels in a cover image can tolerate equal amount of changes without causing noticeable distortion. The changes occur in smooth areas can be easily noticed by human eyes. Therefore,adaptive methods for steganography are presented ([3],[4],[6],[7],[8],[9]) in which the amount of embedding data in pixels is variable and provides a more imperceptible result than those employed by simple LSB s substitution and non-adaptive schemes. Wu-Tsai proposed a novel steganographic method that uses the difference value between two neighboring pixels to estimate how many secret bits should be embedded [6]. Chang-Tseng used the side information of neighboring pixels for each input pixel to help the capacity estimation in edge and smooth areas [7]. In Zhang-Wang scheme, three neighbor pixels are employed to assess the size of secret message for each pixel in the original image [8]. Yang-Wang proposed a multi-pixel differencing method that to determine how many secret bits should be embedded, that used of three difference values in a four-pixel block [9]. For improving the stego-image quality in Wu-Tsai scheme, Wang et al. presented a steganographic method which instead of the difference value that utilizes the remainder of two consecutive pixels to record the information of secret data [4]. Yang et al. proposed an adaptive LSB steganographic method using the difference value of two consecutive pixels based on k-bit modified LSB substitution method to discriminate between edge and smooth areas [3]. Four criteria are used to evaluate the performance of data hiding schemes: the embedding capacity, the visual quality of the stego-image, the security, and the complexity of the data-embedding. However, existing data hiding schemes seldom consider all these factors in their methods. But in 2010, Lee and Chen [1] used of a simple modulus function to imply all the performance factor listed above. However, in Lee-Chen scheme the embedding capacity into each image pixel was fixed and thus that was a non-adaptive method. In order to provide better stego-image quality, larger embedding capacity and increasing level of security, an adaptive method using neighborhood pixels differencing based on modulus function [1] is presented in this paper. The average value of three difference values in four-pixel into a block is utilized to distinguish between edge areas and smooth areas and to estimate how many secret bits will be embedded into the block. Readjust procedure will be applied to extract secret data exactly in destination and to minimize the hiding effects resulted from embedding in the embedding phase.

The remainder of this paper is organized as follows. In Section 2, we represent three existing data hiding methods. In Section 3, the embedding and extracting algorithms of the proposed adaptive method is presented. In the next section, we compare the proposed scheme with Lee-Chen's. The experimental results and comparison will be in Section 5. Finally, conclusions are given in Section 6.

2 Related Work

We now describe three existing data hiding schemes, namely Wang et al.'s 2008[4], Yang et al.'s 2008[3] and Lee-Chen's 2010[1].

In order to produce a better stego-image quality than to Wu-Tsai's [6], in 2008, Wang et al. proposed a novel technique based on pixel-value difference and modulus function. For the embedding secret data, first a difference value from two consecutive pixels and next via a modulus function, the reminder of the two consecutive pixels is computed. By using of an original table rang, data can be embedded into the two pixels with altering their remainder. This method significantly decreased the hiding effects appeared into the stego-image of Wu-Tsai's [6].

In Yang et al.'s scheme the number of embedding bits is evaluated by the range which difference value of two consecutive pixels falls into. The range is divided into two levels, e.g. lower level and higher level. The embedding is performed by executing K-bit modified LSB substitution method, so that the value k is decided via the level which their difference value belong to. The higher level used a larger value of k.

In Lee-Chen's scheme, the values R1, R2, v1, v2 are secret keys. Two set-generation functions Hr(R1,v1) and Hc(R2,v2) are used to generate two sets Kr={Kri | i= $1,2,...,2^{v1}$} and Kc={Kcj | j $=1,2,...,2^{v2}$}, where $R1 \in [1,2^{v1}!]$,$R2 \in [1,2^{v2}!]$ and $v1,v2 \in \{0\} \cup N$. Each element Kri in Kr is unique and its numerical value falls within the range $[0, 2^{v1}-1]$. Similarly, each element Kcj in Kc is unique and its numerical value falls within the range $[0, 2^{v2}-1]$. Kr and Kc have $2^{v1}!$ and $2^{v2}!$ possible permutations, respectively. The bitstream secret data which denoted by S, divided into many secret pieces S_k, so that $S_k = S_{k1} \| S_{k2}$, where S_{k1} contains v1 bits and S_{k2} contains v2 bits and each pixel in cover image can carry (v1+v2)-bit secret data. Via sets Kr and Kc can form a variant of a Cartesian product denoted as $Kr \otimes Kc$. Position S_{k1} into Kr (e.g. Kri) and position S_{k2} into Kc (e.g. Kcj) is determined. Next, index of the bitstream kri||kcj (e.g. d) into $Kr \otimes Kc$ is exploited. Next, for each cover pixel a pixel group G = {g_t | t=1,2,...,n} using a modulus operation is created, where n $=2^{v1+v2}$. Than via index d, the secret piece S_k can be carried by the *dth* element in group G.

Before of the extracting process in the Lee-Chen's, first generate two sets Kr and Kc using Hr(R1,v1) and Hc(R2,v2), respectively. This step is the same with the embedding process. Next, for each stego pixel create the pixel group G and determine the position information d so that the value stego-pixel identical with g_d . Retrieve the *dth* element, which is the secret piece with (v1+v2) bits, from $Kr \otimes Kc$. Repeat before steps until all the stego-pixels have been processed. Finally, concatenate all the pieces of secret data and return the secret information.

The visual quality of the stego-image produced by Wang et al. is better than other schemes [6,7]. But, the embedding algorithm of Wang et al.'s [4] needs to extra steps for revising pixel values, when occurs the problem of overflow and underflow. Yang et al.'s [3] produces the smallest distortion in the LSB-related embedding approaches. But both methods couldn't consider sufficiently the features of edge. Besides, the level of security in other methods [6,7,9] was in low degree. Furthermore, level of security in Lee-Chen's scheme is high and detecting secret data because of existing very much permutations is difficult, but the embedding capacity into each image pixel is fixed and thus Lee-Chen's is a non-adaptive method. In order to provide better stego-image quality, larger embedding capacity and increasing level of security, an adaptive method using neighborhood pixels differencing based on modulus function [1] is presented in this paper.

3 Proposed scheme

We conducted our method based on Lee-Chen 2010 method [1]. The cover images selected 8-bit grayscale images because of, based on psycho visual redundancy in grayscale digital images, the pixels in edge areas than to smooth areas can tolerate much more changes without making perceptible distortion for human eyes and also a grayscale image needs lower space and time for transmission on the internet than to colored images. A cover image with size M×N is indicated as *I* and each cover pixel is denoted as y_i. The bitstream secret message is denoted by S. The stego-image is denoted as I', and y''_i represents each stego-pixel. There are five secret keys namely R1, R2, v1,v2,T and $1 \le v1$, $1 \le v2$, $(v1+v2) \le 5$. The average difference value of a four-pixel block is utilized to classify the block as a smooth area or an edge area. The range of average difference value is partitioned into two different levels, smooth level and edge level. Q-bit of secret data are embedded in Pixels located in the block, where Q is decided by the level which the average difference value belongs to. For embedding, according to the secret keys v1 and v2, smooth level will use a lower value v1 while edge level uses higher value v1+v2. The data embedding process are given in Section 3.1 and the extracting phase is described in Section 3.2.

3.1 Embedding phase

The cover image is partitioned into non overlapping four-pixel blocks. For each block, there are four neighboring pixels $P_{i,j}$,$P_{i,j+1}$,$P_{i+1,j}$,$P_{i+1,j+1}$ and their corresponding gray values are y_0, y_1,y_2,y_3, respectively .The detailed embedding steps are as follows.

Input: I, S and secret keys R1, R2, v1, v2, T.

Output: I′ .

Step 1: Identical with Lee-Chen's scheme[1] which has explained in before section, generate two sets Kr and Kc using Hr(R1,v1) and Hc(R2,v2), respectively. Via sets Kr and Kc form a variant of a Cartesian product e.g. $Kr \otimes Kc$. $Kr \otimes Kc$ generates an ordered set of combinations of Kr and Kc with $2^{v1}\times 2^{v2} = 2^{v1+v2}$ elements (Eq. 1). Each element of the variant cartesian product of the two sets Kr and Kc is a binary string concatenation that combines the two binary strings of Kri and Kcj together to form one bitstream: Kri || Kcj and each element Kri||Kcj has a length of (v1+v2) bits.

$$Kr \otimes Kc = \{Kri \parallel Kcj \mid Kri \in Kr, Kcj \in Kc, \\ i = 1,2,...,2^{v1}, j = 1,2,...,2^{v2}\} \quad (1)$$

Step 2: Calculate the average difference value *D*, which is determined by:

$$D = \frac{1}{3}\sum_{i=0}^{3}(y_i - y_{\min}) \\ y_{\min} = \min\{y_0, y_1, y_2, y_3\} \quad (2)$$

Step3: Our method using threshold key value T embeds secret data into two levels (smooth-level and edge-level). Addition to v1 and v2 keys, T stands for a predefined threshold that can be used to control image distortion and the embedding rate. If D≤T, D belongs to 'smooth-level' and the block belongs to a smooth area, then Q = v1. Otherwise, D belongs to 'edge-level' and the block belongs to an edge area, then Q = v1+v2. To success in the readjust procedure, we must satisfy the following conditions: $2^{v1} \le T \le 2^{v1+v2}$ and $1 \le v1, (v1+v2) \le 5$.

Step 4: Determine whether current block belongs to 'Error Block'. If is, restart from Step2. Otherwise, continue to next step.

Definition 1. Let y_{max}= max $\{y_0,y_1,y_2,y_3\}$, the block is called 'Error Block' if and only if: D ≤ T, $(y_{max}-y_{min}) > 2\times T+2$. 'Error Block' is not used to embed secret bits.

Step 5: For each pixel y_i in the block, separated Q-bit of secret data. For edge blocks, Q divide into two pieces S_{Q1} and S_{Q2}, where S_{Q1} contains v1 bits and S_{Q2} contains v2 bits. For smooth blocks, Q divide into one piece S_{Q1}, where contains the same v1 bits.

Step 6: For edge blocks obtain the indices i and j using the conditions S_{Q1} = Kri and S_{Q2} = Kcj and for smooth blocks determine index i using the condition S_{Q1} = Kri. (Kri and Kcj are *ith* and *jth* elements into Kr and Kc, respectively).

Step 7: For edge areas, bitstream Kri||Kcj into $Kr \otimes Kc$ can be indexed by Eq. (3) and for smooth blocks, bitstream Kri can be indexed by Eq. (4).

$$d = 2^{v2} \times (i-1) + j \quad (3)$$

$$d = i \quad (4)$$

Step 8: Create a pixel group G using the following equation. ($n = 2^Q$).

$$f(y_i) = y_i \bmod n \quad (5)$$

Then derive the corresponding stego-pixel y'_i from *dth* element of G: $y'_i = g_d$.

Step 9: This step is called '*Error reducing procedure*' for minimizing perceptual distortion between cover and stego images. Also this step called *'readjust procedure'* to guarantee the same level that the average differencing value belongs to before and after secret data embedding.

Let $y''_i = y'_i + L \times n$, $n = 2^Q$, $L \in \{0,1,-1\}$, $0 \le i \le 3$.

Search $(y''_0, y''_1, y''_2, y''_3)$ such that:

(1) D'' and *D* belong to the same level, where:

$$D'' = \frac{1}{3}\sum_{i=0}^{3}(y''_i - y''_{\min}) \\ y''_{\min} = \min\{y''_0, y''_1, y''_2, y''_3\}$$

(2) The value of is $\sum_{i=0}^{3}(y''_i - y_i)^2$ minimized.

(3)The final stego-block $(y''_0, y''_1, y''_2, y''_3)$ does not belong to 'Error Block'.

After the replacement of (y_0, y_1, y_2, y_3) by $(y''_0, y''_1, y''_2, y''_3)$ embedded 4×Q-bit of secret data in the block.

Step 10 **.** Repeat Steps 2-9 until the secret pieces have been embedded and obtain the stego-image *I′*.

128	129	130	131	132	133	134	135
1	2	3	4	5	6	7	8
136	137	138	139	140	141	142	143
9	10	11	12	13	14	15	16
144	145	146	147	148	149	150	*151*
17	18	19	20	21	22	23	*24*
152	153	154	155	156	157	158	159
25	26	27	28	29	30	31	32

Fig. 1. Pixel group created for value 155 for embedding secret data

*For instance***,** we present one example with high embedding capacity which that embeds 3-bit of secret data within each pixel in smooth area and 5-bit within each pixel in edge area. Suppose we have a block with four neighboring pixel values (155,99,184,140), and the secret data for embedding in cover image are '10001011000011011010'. Assume v1=3, v2=2, R1=30301, R2=20 and T=20. Before the embedding process, first Kr={001,110,101,010,111,100 ,011,000} can be generated using Hr(30301,3) and Kc={00, 10,11,01}can be created using Hc(20,2). We Calculate the average difference value D=(182 / 3) > T, thus current block has been placed in edge area and is embedded Q=5 bits of secret data in each y_i ,$0 \le i \le 3$ because v1+v2=5. Hence, sum total are embedded 4×5=20 bits in the block. Next we separated four pieces containing 5-bit of secret data. Each 5-bit piece is further separated into two substrings: the 3-bit and the 2-bit substring, respectively. For first pixel into the block, e.g. y_0=155, the first piece '10001' is separated into the two substrings '100' and '01'. Then, we achieve i=6 and j=4 because the sixth element of Kr is '100' and fourth element of Kc is '01'. According to Eq. (3), we compute d using 2^2×(6-1)+4=24. Next, the pixel group G is created for the pixel value y_0=155 with $n=2^{2+3}$=32 via Eq. (5), as shown in fig. 1, where g_{28} =155. Finally, the stego-pixel y'_0 can be obtained from the *dth* element of G, i.e. $y'_0= g_{24}$=151. In the same way, can be obtained reminder the stego-pixel y'_1=120,y'_2=161,y'_3=133 and stego-block (151,120,161,133). Readjust procedure is executed resulting in final stego-block (151,88,193,133). The average difference value for this block obtains using D=(213 / 3) > T. Hence final stego-block not only has the equal level to cover block level but also has been minimized differences between cover and stego pixels. As an another example, with the same given information of before example, suppose cover block is (70,79,109,106). We obtain D=28. Then stego-block is produced (87,88,97,101) and D=8. After executing readjust process, result (55,88,97,101) and D=40. Thus the final stego-block will have the equal level with cover block level.

3.2 Extracting phase

Input: a stego-image *I'* and secret keys v1, v2, R1, R2, T.
Output: a bitstream secret data.

The identical with the embedding process, Partition the stego-image into four-pixel blocks. The following steps are executed to extract the secret data.

Step 1: Generate two sets Kr and Kc using Hr(R1,v1) and Hc(R2,v2), respectively. We use of both Kr and Kc for blocks placed in edge areas and of Kr for blocks located in smooth areas.

Step 2: For each block ($P_{i,j}$,$P_{i,j+1}$,$P_{i+1,j}$,$P_{i+1,j+1}$), calculate the average difference value D by Eq. (2).

Step 3: Use the threshold value T to figure out the level which D belongs to. If D≤T (e.g. smooth area) then Q = v1, otherwise Q = v1+v2.

Step 4: Determine whether current block belongs to 'Error Block'. If not, continue to next step. Otherwise, restart from Step 2.

Step 5: For each pixel into the block create the pixel group G using Eq. (5) and determine the position information d because the stego-pixel $y''_i = g_d$, where $n = 2^Q$.

Step 6: Extract the *dth* element, which is the secret piece with Q=v1+v2 bits, from Kr⊗Kc for the blocks placed into edge areas and the secret piece with Q=v1 bits, from Kr for the blocks placed into smooth areas.

Step 7: Repeat Steps 2-6 until all the stego- blocks have been processed.

For instance, we extract the embedding example (151, 88,193,133), which is shown in the before subsection. Assume v1=3, v2=2, R1=30301,R2=20 and T=20. Kr={001, 110,101,010,111,100,011,000} using Hr(30301,3) and Kc= {00,10,11,01} by using Hc(20,2) are generated. We produce the variant Cartesian product Kr⊗Kc, which is {00100, 00110,00111,00101,11000,11010,11011,...,00011,00001}. Because D=(213 / 3) > T, this block is placed in edge area and hence Q=v1+v2=3+2=5 bits have hided into each Pixel in the block. Sum total, 4×5=20 bits have embedded in current block. Let us consider third pixel into the block (e.g. y''_2=193). The pixel group G is created for value 193 via Eq. (5) with $n=2^{2+3}$=32. The position of stego-pixel 193 in G is 2, because d=(193mod 32)+1=2. The piece of binary secret data '00110' can be extracted because '00110' is the second element of Kr⊗Kc. Similarly, has extracted the secret piece '10001' for y''_0, '01100' for y''_1 and '11010' for y''_3. Finally we achieve '100010110000110 11010' which that is the same secret data in the embedding example of before subsection.

4 Analysis and discussion

Generally, suppose a,b∈{0}∪N and (a mod b) = x. The following equation verified in each division:

$$(a-b) \bmod b = a \bmod b = (a+b) \bmod b \qquad (9)$$

Therefore, according to used modulus function and without loss of generality of extracting phase in Lee-Chen scheme, readjust procedure can work correctly. In this procedure for preventing of overflow or underflow problem, decrease of y'_i by n and increase y'_i by n may not be allowed if y'_i < n and y'_i > (255-n), respectively.

We now compare the proposed adaptive scheme in this scholar with Lee-Chen's [1]. Both these methods possess common factors: 1) need little memory space (Only ($v1\times2^{v1}+v2\times2^{v2}$) bits of memory space are required for storing Kr and Kc), 2) the problem of overflow or underflow does not occur, regardless of the nature of the cover pixels. Because, let us assume that the pixel intensity set is λ={0,1,2,3,...,255} and is an ordered set of pixel values which dominates the pixel values of a 8-bit gray-scale image. But G⊆λ and each element of λ falls into the rang [0-255]. Also in readjust process will be not occur overflow and underflow problem, 3) detecting secret data is difficult. Because existing very much permutations (totally, $2^{v1}!\times2^{v2}!$ for Kr⊗Kc), an unauthorized user will face extreme difficulty in guessing the secret data. But, our adaptive method is superior to Lee-Chen's scheme. Because, firstly the embedding capacity in Lee-Chen's scheme just via v1,v2 keys is scalable. A larger v1 or v2 can yield a greater embedding capacity and a smaller v1 or v2 can obtain a higher visual quality of stego-image. However, in lee-Chen non-adaptive scheme after of determining v1 and v2 keys only could achieve to fixed embedding capacity, e.g. M×N×(v1+v2). But, in our adaptive and flexible method after adjusting v1 and v2, by means of various values of key T, the embedding rate as well as the image quality can be adjusted depending on the requirements of the practical applications. Accordingly, a larger v1 or v2 and a lower T enhance embedding rate whereas a lower v1 or v2 and a higher T enhances stego-image quality. Hence our method is more scalable than to Lee-Chen's. Secondly, data hiding system security has provided via secret keys. In other words, extracting the secret data will be meaningless without knowing correct values of keys. The receiver must have the same set-generation functions Hr() and Hc() and appreciates the values of the secret keys v1,v2,R1,R2,T. Additionally, because of adding secret key T the security of our method has been increased than to Lee-Chen's scheme and hence detecting of secret data for eavesdropper will be more difficult. Thirdly, our adaptive method is better than Lee-Chen scheme in both the embedding capacity and PSNR value, as will be indicate in experimental results. In our adaptive scheme, has considered a block with 4×4 pixels due to a block of 4×4 pixels is neither too small nor too large to reflect the local complexity of an image. Also using a block with a larger size may increase the probability of degree revision and hence increases the distortion produced due to hidden data. In our method, 'Error Block' is not used to embed secret bits. Generally, there are significantly few error blocks in a cover image. So it will have a little effect on the capacity of our method, which can be almost ignored [2]. For example, let T=5, a block with four-pixel values (178,179,191,179) belongs to *'Error Block'* because, D= (15 / 3) ≤ 5 and *191-178=13> 2×5+2*.

5 Experimental results

Several experiments are preformed to evaluate our proposed methods. Eleven grayscale images with size 512×512 are used in the experiments as cover images, namely 'Lena', 'Baboon', 'Peppers', 'F16', 'Boat', 'Man', 'Tiffany', 'Barbara', 'Elaine', 'Couple', 'Splash', and are shown in Fig.2.The proposed scheme has been implemented using the MATLAB 7.8.0.347 (R2009a) program on Windows XP platform. We used a series of pseudo-random numbers as the secret data to be embedded into the cover images and also utilized the peak signal-to-noise ratio (PSNR) value to evaluate the stego-images quality. The PSNR is defined as follows.

$$PSNR = 10 \times \log_{10} \frac{255 \times 255}{\frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (p_{i,j} - q_{i,j})^2} (dB) \qquad (10)$$

here $p_{i,j}$ and $q_{i,j}$ denote the pixel values in row i and column j of the cover image with M×N size and the stego image, respectively. A high PSNR value indicates that the stego-image is very similar to the original image, whereas a low value indicates the opposite. Generally, distortion is indiscernible to the human eye when PSNR is higher than 30 dB.

Fig. 2. Eleven cover images used for the proposed scheme.

Table 1 : Experimental results with various parameters.

Cover images	2-3, T=7		3-4, T=15	
	Capacity	PSNR	Capacity	PSNR
Lena	591544	43.92	812484	39.50
Baboon	723808	41.43	916316	36.62
Peppers	573000	44.47	808024	39.69
F16	586244	44.12	817948	39.31
Boat	641828	42.71	830332	38.80
Man	641593	42.59	838032	38.17
Tiffany	583573	44.16	809228	39.62
Barbara	648980	42.71	873100	37.62
Elaine	653164	42.47	817068	39.18
Couple	617508	43.27	826428	38.89
Splash	578240	44.19	795172	40.28

Table 2 : Experimental results with various parameters.

Cover images	2-4, T=12		4-5, T=28	
	Capacity	PSNR	Capacity	PSNR
Lena	595120	41.63	1057480	34.30
Baboon	830432	36.75	1118320	31.93
Peppers	580208	42.34	1058764	34.20
F16	603256	41.43	1064080	34.00
Boat	646400	40.00	1064940	33.94
Man	660840	39.42	1068640	33.02
Tiffany	588056	41.95	1056668	34.43
Barbara	710760	38.64	1091152	32.85
Elaine	633656	40.16	1053116	34.53
Couple	631176	40.38	1061764	33.95
Splash	546080	44.39	1053912	34.52

We have experimented using a series of v1 and v2 division with various threshold values. For example, 3-4 division (v1=3,v2=1) with T=15 means that into each four pixels of the block with average difference value placing into smooth area and edge area, will be embedded the 3-bit (e.g. v1 bits) and 4-bit (e.g. v1+v2 bits) respectively. Tables 1 and 2 show the results of our method in terms of embedding capacity and PSNR values. The PSNR values and the embedding capacities (in bits) are average values of the results executed by random bit streams many times. Fig. 3 indicates the stego-images produced by proposed adaptive scheme. As the figures show, the distortions resulted from embedding are imperceptible to human vision. Tables 3 and 4 show the comparisons of results of the embedding rate and image quality between Lee-Chen's and ours. (For instance, in table 4 for Lee-Chen's the values of v1 and v2 have considered so that: v1+v2=4 and in our scheme has been determined: v1=4 and v2=1). Hence in Lee-Chen's scheme, each cover pixel embeds 4-bit of secret data but in our method, each pixel placed in smooth and edge block embeds 4-bit and 5-bit, respectively). Tables 5 and 6 show the comparisons of the results between Yang et al.'s, Wang et al.'s and ours, in terms of embedding capacity and PSNR value. As a matter of fact, our scheme is superior to Lee-Chen, Wang et al. and Yang et al. schemes in three respects, namely visual quality of stego-image, embedding capacity, level of security (because of existing five secret keys).

Table 3 : Comparisons of the results between Lee-Chen's and our method.

Cover images	Lee-Chen's method 2010 [1], 3-bit		Our method 3-4, T=16	
	Capacity	PSNR	Capacity	PSNR
Lena	786432	37.93	810052	39.58
Baboon	786432	37.94	909804	36.76
Peppers	786432	37.94	806576	39.75
F16	786432	37.98	816132	39.41
Boat	786432	37.95	826820	38.95
Man	786432	37.91	833912	38.32
Tiffany	786432	37.91	807040	39.74
Barbara	786432	37.93	861788	37.95
Elaine	786432	37.90	811760	39.40
Couple	786432	37.94	823096	39.01
Splash	786432	37.98	794680	40.32

Table 4 : Comparisons of the results between Lee-Chen's and our method.

Cover images	Lee-Chen's method 2010 [1], 4-bit		Our method 4-5, T=31	
	Capacity	PSNR	Capacity	PSNR
Lena	1048576	31.80	1055620	34.38
Baboon	1048576	31.85	1108708	32.21
Peppers	1048576	31.86	1057584	34.26
F16	1048576	31.85	1061748	34.12
Boat	1048576	31.84	1061852	34.08
Man	1048576	31.82	1065496	33.14
Tiffany	1048576	31.84	1055268	34.50
Barbara	1048576	31.89	1085072	33.07
Elaine	1048576	31.83	1051984	34.59
Couple	1048576	31.85	1058792	34.10
Splash	1048576	31.87	1053448	34.55

Table 5 : Comparisons of the results between Yang et al.'s and our method.

Cover images	Yang et al.'s2008, [3] 2-3 and 3-4		Our method 2-3 T=8 and 3-4 T=8	
	Capacity	PSNR	Capacity	PSNR
Lena	575188 837332	43.95 36.28	581852 843692	44.22 38.31
Baboon	695310 916010	41.15 33.01	712984 974264	41.60 35.61
Peppers	561236 823380	44.49 37.17	566836 828892	44.70 38.83
F16	568184 830328	44.38 37.80	579876 841856	44.32 38.46
Boat	624284 886028	42.69 35.53	625924 887656	43.10 37.09
Tiffany	566992 829136	44.23 37.10	575491 837288	44.40 38.53
Barbara	629976 892120	42.75 34.83	640664 902488	42.86 36.95
Elaine	621052 883196	42.57 33.52	634044 895252	42.88 36.82
Couple	581753 840470	43.70 36.87	606492 868408	43.54 37.55

Table 6 : Comparisons of the results between Wang et al.'s and our method.

Cover images	Wang et al.'s method 2008 [4]		Our method 1-2, T=3	
	Capacity	PSNR	Capacity	PSNR
Lena	409752	44.15	412416	47.74
Baboon	457168	40.32	506888	46.55
Peppers	407256	43.28	415480	47.56
F16	421080	42.14	468053	46.98
Boat	426007	42.42	457536	47.06
Man	426007	42.42	457536	47.06
Tiffany	407360	43.80	419212	47.61
Barbara	442560	42.34	446016	47.27
Elaine	408592	44.74	477664	46.87
Couple	412824	43.25	432520	47.44
Splash	389459	44.34	364397	48.51

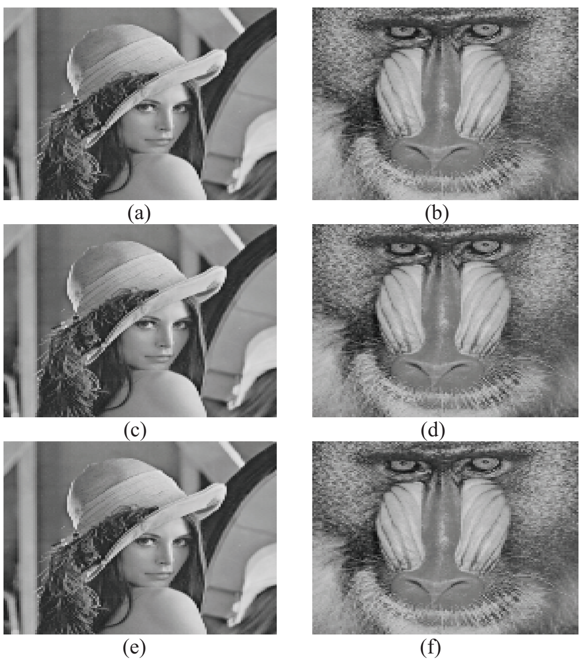

Fig. 3. (a) Original Lena image, (b) Original Baboon image ,(c) Stego-image of Lena, 2-3 T=8. (embedded 581852 bits, PSNR= 44.22dB)(d)Stego-image of Baboon, 2-3 T=8. (em bedded 712984 bits , PSNR = 41.60 dB) (e) Stego-image of Lena, 2-4 T=12. (embedded 595120 bits, PSNR =41.63 dB) (f) Stego-image of Baboon, 2-4 T=12. (embedded 830432 bits, PSNR= 36.75dB)

6 Conclusions

This study developed an adaptive data hiding scheme that uses of four-pixel differencing and modulus function. Our scheme is based on the concept of human vision sensitivity, so the pixels in edge areas than to smooth areas can tolerate much more changes without making visible distortion for human eyes. Accordingly, the number of bit to be embedded into each block is variable and determined by the correlation between neighborhood pixels into that block. Existing secret keys have enhanced the security of our method. Experimental results indicate that the proposed adaptive scheme significantly is superior to the currently existing scheme, in term of stego-image visual quality, embedding capacity, level of security. Because of level of security in ours is high and detecting secret data, because of existing very much permutations is extremely difficult, however our method products good result, it can be in future works to beside the metrics addressed in this scholar, weave other adaptive steganographic method to achieve a stego-image with higher quality.

7 References

[1] C.F. Lee ,H.L. Chen, A novel data hiding scheme based on modulus function .The Journal of Systems and Software 83, 832–843, 2010.

[2] X. Liao , Q.Y. Wen , J. Zhang, A steganographic method for digital images with four-pixel differencing and modified LSB substitution. J. Vis. Commun. Image R., 1-8, 2010.

[3] C.H. Yang, C.Y. Weng, S.J. Wang, H.M. Sun, Adaptive data hiding in edge areas of images with spatial LSB domain systems, IEEE Transaction Information Forensics Security.3 (3) , 488–497, 2008..

[4] C.M. Wang, N.I. Wu, C.S. Tsai, M.S. Hwang, A high quality steganography method with pixel-value differencing and modulus function, The Journal of Systems and Software 81, 150–158, 2008.

[5] Hartung, F., and Kutter, M. Information hiding – a survey, Proc. IEEE, 1999, 87, pp. 1062–1078

[6] D.C. Wu, W.H. Tsai, A steganographic method for images by pixel-value differencing. Pattern Recognition Letters 24, 1613–1626, 2003.

[7] C.C. Chang, H.W. Tseng, A steganographic method for digital images using side match. Pattern Recognition Letter 25 , 1431–1437, 2004.

[8] X. Zhang,,S. Wang, Steganography using multiple-base notational system and human vision sensitivity. IEEE Signal Processing Letters 12, 67–70, 2005 .

[9] C.H. Yang, C.Y. Weng, A steganographic method for digital images by multi pixel differencing, in: Proceedings of International Computer Symposium, Taipei, Taiwan, R.O.C., pp. 831–836, 2006.

[10] D.W. Bender, N.M. Gruhl, A. Lu, Techniques for data hiding, IBM Syst. J. 35, 313–316,1996.