International Journal of Computer Applications Technology and Research
Volume 4– Issue 4, 318 - 321, 2015, ISSN:- 2319–8656

Data hiding using Pixel Value Differencing

Shruti
Deptt. Of Computer Science
Guru Nank Dev University
Gurdaspur, India

Abstract:-This paper presents embedding of data in an image using pixel-value differencing technique. This scheme is used to embed large amount of data by changing the difference between two pixels so that we are able to increase the embedding capacity. There is another technique that is pixel value shifting which also increase the embedding capacity but according to this scheme capacity will increase at edge areas of image.

Keywords: - Capacity; Pixel-value differencing; Image quality

1. INTRODUCTION

Nowadays, the Internet has become a common communication channel. Communicating in public system means that some problems need to be faced, such as data security, copyright protection, etc. Ciphering is a well-known method for security protection but it has the disadvantage of making a message unreadable thereby attracting the attention of eavesdroppers. This makes steganography which hides data within data a good choice for secret communications. One of the best-known steganographic methods is the least significant- bit (LSB) substitution[1]. The simple LSB substitution method replaces the length-fixed LSB with the fixed length bits. Although the technique is efficient, it is rather easy to create a noticeable distortion for the human eye or can be detected by some programs. Therefore, several adaptive methods have been proposed for steganography in order to decrease the distortion caused by the LSB substitution[3]. In addition, some methods use the concept of human vision to avoid the detection of Programs. Now a technique "pixel-value differencing" steganographic method that used the difference value between two adjacent pixels in a block in order to determine the number of embeddable secret bits . This difference value is adjusted so as to embed the secret bits, and the difference between the original and new difference values is adjusted between the two pixels. To check the proposed method, author applies the dual statistics method , called as RS-diagram, to detect the function of embedding method. In RS-diagram, first of all, the discrimination and flipping functions are applied to define pixel groups: Regular (R), Singular (S), and Unusable (U). Then, the percentages of all groups of Regular and Singular with masks m= [0110] and −m= [0−1−1 0] are computed, in which they are represented as Rm, R−m, Sm, and S−m, respectively. Finally, the RSdiagram applied hypotheses of Rm ~= R−m and S ~= S−m S−m to present the detected resultant. There is a method which combines the pixel-value differencing and LSB replacement method. This approach provides higher capacity[2] than pixel-value differencing, but it does not pass the detection of RS-diagram. But pixel-value differencing method not only provides high capacity but also passes the detection of RS-diagram:-

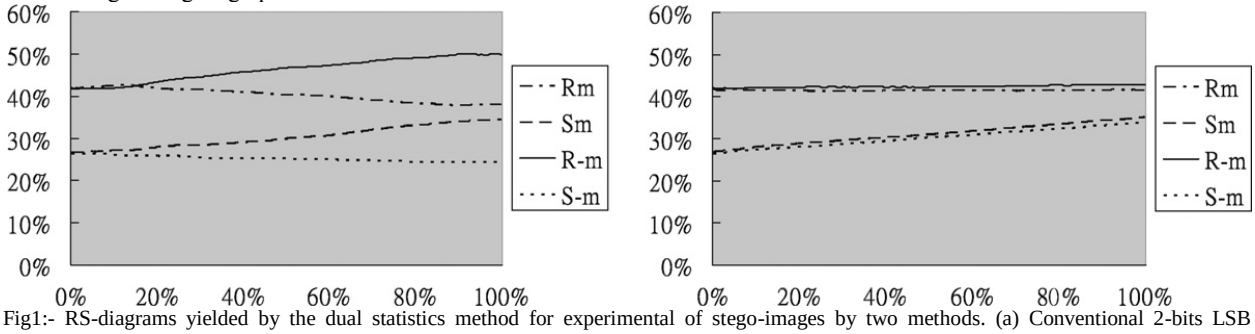

Fig1:- RS-diagrams yielded by the dual statistics method for experimental of stego-images by two methods. (a) Conventional 2-bits LSB substitution; (b) pixel-value difference method.

In this paper, we propose an efficient steganographic scheme to hide data imperceptibly in gray-level images. This scheme is based on the property of human eye, which is more sensitive to the change in the smooth area than the edge area. In this paper, we process a block of four neighboring pixels simultaneously. The number of secret bits to be embedded in a block depends on the degree of smoothness or sharpness. Each four-pixel block is divided into two two-pixel groups, and each group is processed using the approach of pixel-value differencing. In order to extract the embedded data correctly, some schemes are designed cleverly to differentiate which pixels belonging to different groups.

Some conditions will cause a block to be abandoned without embedding. To overcome this obvious drawback, a new technique known as pixel-value shifting is proposed.

2. LITERATURE REVIEWS

Steganographic method hides secret data in graylevel images by pixel-value differencing[4]. First, the host image is partitioned into non-overlapping consecutive two-pixel blocks by scanning all the rows of the host image in a zigzag manner. A difference value d is calculated from the two pixels, say pi and pi+1, of each block. By symmetry, only the possible absolute values of d (0 through 255) are considered and they are classified into a number of contiguous ranges, called as Ri, where i=1, 2, 3, . . ., n. The width of Ri is ui −li + 1, where ui is the upper bound of Ri and li is the lower bound of Ri. The width of each range is taken as a power of 2. This restriction of width facilitates the embedding of binary data. If d falls in smooth area, less secret data will be hidden in the block. On the other hand, if d falls in sharp area, then the block has higher tolerance and thus more secret data can be embedded inside it. Suppose that d falls into the range Rk. The number of embedding bits is determined by the width of Rk. Therefore, the embedding operation is to replace d with a new difference value d_, which is the sum of the embedded value and the lower bound of Rk. Finally, an inverse calculation from d_ is performed to generate the new gray values of the two pixels in the block. Note that the new gray values of the two pixels must lie in between the range [0, 255]. Therefore, if the new gray values are created by value uk, which are the maximally possible value of d_, falling outside the range [0,255], the block must be abandoned for embedding data. In the extracting phase, the secret data are extracted from the blocks of the stego-image in the same order as the embedding phase. The number of secret bits to be embedded in a two-pixel block is determined by the range Rk, which is the range of the difference value between two pixels. In addition, the value of the embedded data in the block is calculated by subtracting the lower bound of Rk from the difference value of the block. Therefore, the embedded bits in the block can be reconstructed. To verify the security of the proposed method, authors apply statistic steganalytic technique which is called RS-diagram, in order to prove that the method is undetected. The results are shown in Fig. 1.

3. OUR APPROACH

In this section, steganographic scheme based on blockwise embedding. We use the idea of pixel-value differencing, but we process four pixels simultaneously in spite of two at a time. In the pixel-value differencing approach[5], each time two pixels are grouped for embedding secret data. Fig. 2(a) shows the only grouping result of Wu and Tsai's method for a four pixel block. However, as shown in Fig. 2(b), there are three kinds of possible grouping results. In order to embed data more efficiently, we have considered different grouping results in our approach. Moreover, new techniques are proposed in order to avoid the additional information needed for recording the selected grouping data. The grouping, embedding, and extracting procedures of our approach are described in the following subections.

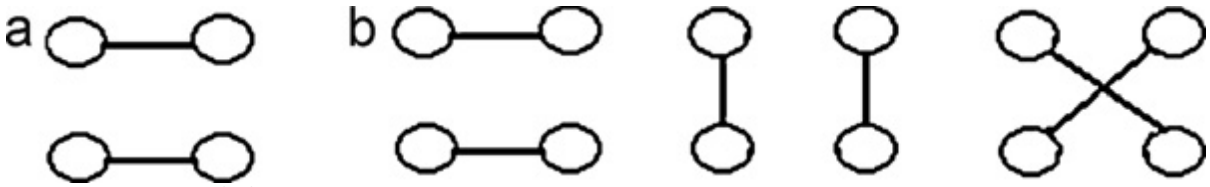

Fig:-2 Grouping results of a four-pixel block: (a) the only grouping result (b) all possible grouping results.

3.1 Pairwise grouping of a block

The host images used in our scheme are 256 gray value. Two difference values d1 and d2 are computed from each non-overlapping block with four neighbor pixels, say pi,j, pi,j+1, pi+1,j+1, and pi+1,j, of a given host image. The strategy of partitioning the host image into four-pixel blocks is to run through all the rows in a raster scan. The four pixels pi,j, pi,j+1, pi+1,j+1, and pi+1,j are then renamed as p1, p2, p3, and p4, and their corresponding gray values g1, g2, g3, and g4 satisfy the condition g1 ≤g2 ≤g3 ≤g4. The four-pixel block is partitioned into two two-pixel groups (g1, g4) and (g2, g3). The group which belongs to pi,j is defined as group1, and the other is defined as group2. In our scheme, the two group differences are computed as (g4 −g1) and (g3 −g2). The group difference of group1 is denoted as d1, and the group difference of group2 is denoted as d2. The difference value di (where i = 1 or 2) may be in the range from 0 to 255.

3.2 Data embedding

The secret message can be seen as a long bit stream. The task is to embed this stream into the four pixels block. The number of bits which can be embedded varies and is decided by the width of the range to which the difference values of the block belongs. Let the group difference di (where i = 1, 2) falls into the range of index ki. Then, the number of bits, say ni, to be embedded in ith group is calculated by ni = log2(uki – lki + 1). We embed n1 bits and n2 bits into group1 and group2, respectively. Let S1 and S2 be the bit streams of the secret message to be embedded, where |S1| = n1 and |S2| = n2. The two new differences d_ 1 and d_ 2 can be computed by:-

d_ I = lki + bi, where i = 1, 2.

In the above equation, b1 and b2 are the decimal values of S1 and S2, and are embedded into group1 and group2, respectively. By replacing d1 and d2 with d_ 1 and d_ 2, respectively, an inverse calculation from d_ 1 and d_ 2 generates the new pixel values g_ 1, g_ 2, g_ 3, and g_ 4 of the pixels p1, p2, p3, and p4. For the following two reasons, the pixel values g_ 1, g_ 2, g_ 3, and g_ 4 need to be modified further. One is that all values g_ 1, g_ 2, g_ 3, and

g_ 4 must fall into the range [0, 255]. The other is that the two groups need to be distinguished after embedding. The key point in order to successfully distinguish between two groups is to maintain the two intervals [g_ 1, g_ 4] and [g_ 2, g_ 3] such that one of them contains the other.

3.3. Data extracting

The process of extracting the embedded message is similar to the embedding process with the same traversing order of visiting all blocks. First,were name the pixels of a block as (p1, p2, p3, p4) and distinguish group1 and group2. Then, all values of di (i = 1, 2), ki (i=1, 2), and ni (i = 1, 2) of this block are found. Note that the calculation of these values is the same as the embedding process, except that the block now comes from stego-images. The bit-stream values b1 and b2, which are embedded in this block are then extracted using the following equations:

bi = di – lki , where i = 1, 2

Finally, each value bi is transformed into a bit stream with ni bits.

4. EXPERIMENTAL RESULTS

(a)

(b)

(c)

Fig:-3. Two of the experimental host images with size 512×512: (a) Peppers; (b) Baboon (c) Leena.

(a) (b)

Fig:-4. Two secret images with size 256×256: (a) Boat; (b) Airplane.

In our experiments, we used two host images with size 512×512. These are shown in Fig. 3. Two sets of widths, which partition the range [0, 255], are used in the experiments. The first experiment selects the widths of 8, 8, 16, 32, 64, and 128, which partitions the range [0, 255] into ranges [0, 7], [16, 31], [32, 63], [64, 127], and [128, 255]. The second experiment is based on the widths of 16, 16, 32, 64, and 128. We used a random bit stream, images "Boat" and "Sailboat" as separate secret messages in the experiments. Images "Boat" and "Airplane" are 256×256 as shown in Fig. 4. If a random bit stream is used as secret data, then the values of PSNR and capacities are the averages of the results obtained by 100 times executing the different random bit streams.

Table 1:- The capacities and PSNRs for embedding random bit stream, image Boat, and image Sailboat by our approach.

Cover images (512 × 512)		Widths of 8, 8, 16, 32, 64, and 128			Widths of 16, 16, 32, 64, and 128		
		Random bits	Boat	Airplane	Random bits	Boat	Airplane
Lena	Capacity	410,854	410,854	410,854	528,966	528,966	528,966
	PSNR	40.54	41.24	41.21	36.75	37.38	37.23
Baboon	Capacity	482,515	482,515	482,515	559,222	559,222	559,222
	PSNR	34.67	35.31	35.21	34.30	34.46	34.32
Peppers	Capacity	408,281	408,281	408,281	528,791	528,791	528,791
	PSNR	40.47	41.19	41.05	36.83	37.38	37.27

Table 2:- The results of our approach using the widths of 8, 16, 32, 64, 128, and 8.

Our scheme		Cover images		
		Lena	Baboon	Peppers
Ranges	Capacity	432,333	524,785	428,909
8–16–32–64–128–8	PSNR	38.86	33.31	39.15

The capacities and PSNR of embedding the results of two sets of widths are shown in Table 1. The capacities of the second experiments are higher than that of the first experiment. This is due to the fact that the second experiment uses larger widths. Consequently, the values of PSNR of the second experiment are worse.

5. ANALYSES AND DISCUSSIONS

There are three conclusions shown in this section. First, our approach finds out more edge areas for various images. In other words, more secret could be embedded into the edge areas by means of our scheme. Second, our approach can avoid the conditions of falling out of the boundary. We propose the skill of pixel-value shifting to shift the pixel values. It can solve the conditions of falling out of the boundary, and hence increase the embedded messages shows the amounts of pair wise pixels falling out the boundary . Although the amounts of falling out of the boundary are not large in this method, however, our approach guarantees that all pair wise pixels are usable. Finally, there is a partition scheme of range, but it is not practical. In their partition, the later ranges are larger for the reason that edge areas can tolerate larger distortions. Therefore, their partition of range is reasonable. We suggest a more practical partition with widths 8, 16, 32, 64, 128, and 8. The experimental results are shown in Table 2 hence capacities increase 17,344–42,270 bits and the PSNR values are still accepted by human eyes.

6. CONCLUSION

This Paper presents the technique for embedding the data keeping human vision in mind. Here I use blockwise approach in which I process four pixel block simultaneously. According to pixel value differncing, if a block in a sharp area then embedding capacity will increases. Therefore, by this approach large amount of secrete data can be embedded in host image.

7. References

[1]. Ker, A.D., 2007. Steganalysis of embedding in two least-significant bits. IEEE Transactions on Information Forensics and Security 2 (1), 46–54.

[2] .Lee, Y.K., Chen, L.H., 2000. High capacity image steganography. IEE Proceedings on Vision Image and Signal Processing 147 (3), 288–294.

[3].Li, X., Yang, B., Cheng, D.F., Zeng, T.Y., 2009. A generalization of LSB matching. IEEE Signal Processing Letter 16 (2), 69–72.

[4].Liu, J.C., Shih, M.H., 2008. Generalizations of pixel-value differencing steganography for data hiding in images. Fundamenta Informaticae 83 (3), 319–335.

[5].Wang, C.M., Wu, N.I., Tsai, C.S., Hwang, M.S., 2008. A high quality steganographic method with pixel-value differencing and modulus function. Journal of Systems and Software 81 (1), 150–158.