Improving Security in Tri Pixel Difference Value Method

Mubin Tamboli and Avinash Gulve

Abstract—**Pixel value differencing (PVD) is a steganography method which embeds secret data in images based on spatial information. And to improve the capacity of hiding data is increased by using Tripixel Difference Value Method. In which data is hidden in 2X2 Square block. Here proposed method improves the security of existing TPVD method by making certain modification to make it to more robust to histogram quantization. But it reduces certain capacity of original TPVD algorithm. And the stability of against histogram quantization is significantly improved.**

Index Terms—**Stegoimage, stegnalysis, segnography, PVD, TPVD, cryptography.**

I. Introduction

Steganography is the art and science of invisible communication over network. This is accomplished through hiding information in other information, thus hiding the existence of the communicated information. The strength of steganography can thus be amplified by combining it with cryptography [1]. Two other technologies that are closely related to steganography are watermarking and fingerprinting. Hiding information in text is historically the most important method of steganography. Text steganography using digital files is not used very often since text files have a very small amount of redundant data.

The main purpose of steganography is to hide a message in another one in a way that prevents any attacker to detect or notice the hidden message. The aim of this work is to develop a new method for hiding message in gray-scale images, mainly embedding text data in digital images.

Watermarking is one way for hiding the data in the manner of steganography which is used mainly for the copyright. Here the purpose of the stegnography is to hide data in large quantity. There are various schemes for hiding the data in images.

An image is a collection of numbers that constitute different light intensities in different areas of the image. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the Internet consists of a rectangular map of the image's pixels (represented as bits). These pixels are displayed horizontally row by row. The number of bits in a color scheme, called the bit depth, refers to the number of bits used for each pixel. The smallest bit depth in current color schemes is 8, meaning that there are 8 bits used to describe the color of each pixel. Monochrome and grey scale images use 8 bits for each pixel and are able to display 256 different colors or shades of grey.

In LSB [2] method of image steganography, data hidden in each pixel's byte at the LSB. However, since some pixels could not tolerate changes of substitution during the embedding process, then those pixels turn apparently different from their original values. This effect appears easily in the smooth area that they are noticeable for human eyes. So main approach is to find the adaptive method based on using variable amount of bits instead of fixed length.

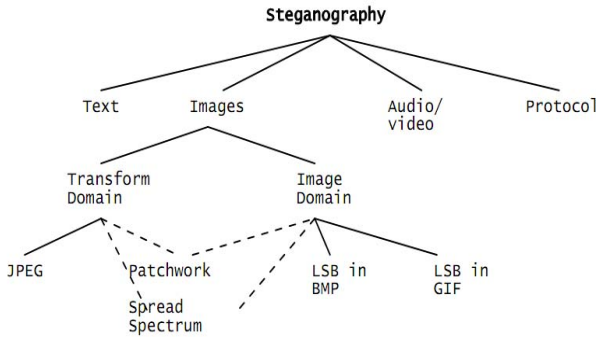

Fig. 1. Stegnographic techniques

Two things are considered in image steganography, first is capacity of hiding data and another is quality of stego-image. There is one method which was proposed for hiding large amount of data without degrading the performance of image. In this method the data is hidden in difference of two consecutive pixel-Values. In this method edge only in one direction is considered. Various other methods have been proposed on the basis of PVD method. However this steganography can be detected by difference histogram analysis.

In Tri-Pixel Value Difference method data is embedded in 2x2 blocks of 4 pixels which improves the quantity of data to embed. But data can be detected in histogram analysis.

II. Review of the PVD and TPVD Method

In PVD method [3], [4], [5] the secret data is assumed to be a long-bit stream and the cover image is a gray-level image. The image is partitioned into non overlapping blocks composed with two consecutive pixels (p_i, and pi_{+1}). From each block difference d is calculated which ranges from -255 to 255. Therefore |d| ranges from 0 to 255. The block with a small value |d| locates in the smooth area, whereas a block

Manuscript received March 8, 2012; revised May 5, 2012.

M. Tamboli is with the Department of computer engineering, faculty of engineering, Amrutvahini College of Engineering, Sangamner, Maharashtra, India. (e-mail: yesmubin@rediffmail.com).

A. Gulve is with the Department of MCA, faculty of MCA, Government college of engineering, Aurangabad, Maharashtra, India (e-mail: akgulve@yahoo.com).

with a large value |d| is considered as a block with sharp edges. According to the properties of human vision, eyes can tolerate more changes in sharp-edge blocks than in smooth blocks. That is, more data can be embedded into the edge areas than into smooth areas. Therefore, in the PVD method, the first step is to design a range table with n contiguous ranges (R_K where k = 1, 2,.... , n) and the table range is from 0 to 255.

The embedding algorithm is described as follows:

1) Calculate the difference value d_i between two consecutive pixels p_i and p_{i+1} for each block in the cover image. The value is given by $d_i = p_{i+1} - p_i$.
2) Using $|d_i|$ to locate a suitable R_k in designed range table, that is to compute j = min(u_i - $|d_i|$) where $u_k \geq |d_i|$ for all $1 \leq k \leq n$. then R_j is the located range.
3) Compute the amount of secret data bits t that can be embedded in each pair of two consecutive pixels by R_j The value t can be estimated from the width w_j of R_i , this can be defined by t = [$log_2 w_j$].
4) Read t bits from the binary secret data and transform the bit sequence into a decimal value b . For instance, if bit sequence =110, then the converted value b = 6. Calculate the new difference value d'_1 given by $d'_1 = l_j +$ b, if $d_i \geq 0$ or $d'_1 = -(l_j +$ b), if $d_i < 0$ to replace the original difference d_i .
5) Modify the values of p_i and p_{i+1} by the following formula: $(p'_i, p'_{i+1}) = (p_i - [m/2], p_{i+1} + [m/2])$ where m = $d'_i - d_i$. Until now, to embed the secret data into pixel pair (p'_i, p'_{i+1}) is done by changing values of p_i, p_{i+1}.

Repeat Step 1-5 until all secret data are embedded into the cover image, then the stego-image is obtained.

During the phase of secret message extraction, the original designed range table is required. In the beginning, the same method as in the embedding phase is used to partition the stego-image into pixel pairs (blocks). Then the difference value d_i for each pair of two consecutive pixels p_i and p_{i+1} in the stego-image is calculated. Next, $|d_i]$ is used to locate the sitable R_j during the embedding phase. Therefore, b is obtained by subtracting l_j from $|d_i]$. if the stego image is not altered, **b** is equal to b. Finally **b is** transformed from a decimal value into a binary sequence with t bits where t = $[\log_2 w_j]$.

In TPVD [6][7][8]method, the image is partitioned into 2x2 non-overlapped blocks. Data is hidden in each 2 x 2 block which includes four pixels of *P(x,y)*, *P(x+1,y)*, P(x,y+1), P(x+1,y+1) where x and y are the pixel location in I Image. we are considering 3 pairs $P_0 = (P(x,y), P(x+1,y))$, $P_1 = P(x,y),P(x,y+1))$ and $P_2 = P(x,y), P(x+1,y+1))$ respectively.

Same method of PVD is used for embedding and extracting data. The difference is that three pairs of pixels are considered. While embedding the secret data, each pair is modified *(P'i)* and a new difference value d'_i for $i = 0,1,2$ is calculated. The new pixel values in each pair are different from their original ones. Therefore, one of the *P'i* is selected as the reference point to offset the other two pixel values. Selecting different reference points results in varied distortion to the stego-image. An optimal selection approach is used to achieve minimum Mean-square-Error (MSE). Suppose the $m_i = d'_i - d_i$ where d'_i and d_i are difference value of pixel pair i before and after embedding procedure. Rule to find out optimal reference pair is as follows:

1) If m_i is greater than 1 or smaller than -1the optimal pixel pair is i' is the pair with greatest |m|.
2) If all mi have the same sign and only one mi belongs to {0, 1, 1}, then the optimal pixel pair optimal i is selected from the other two pairs with the smallest |m|.
3) If only one mi has a different sign from the other two pairs, the optimal pixel pair optimal i is selected from the other two pairs with the smallest |m|.
4) If only one mi= {0,1, 1} and the other two mi has different signs, the optimal pixel pair optimal i is the pair with {0,1, 1}.
5) If there exists more than one pair with mi= {0,1, 1}, the optimal pixel pair optimal i can be selected as any one pair with {0,1, 1}.

By considering optimal pair, calculate the new values of all three pixel in the block. Improved technique of pixel difference value method is described in [5].

III. Proposed Method

Original TPVD method provides very high capacity for data hiding but it is very sensitive in stegnalysis attack. The histogram difference is always in between -255< d < 255.

At first partition the image as usual in TPVD method i.e. into non-overlapping 2 × 2 blocks with 4 pixels. Consider three pairs P0 = (P(x,y), P(x+1,y)) , P1 = P(x,y),P(x,y+1)) and P2 = P(x,y), P(x+1,y+1)) respectively.

To avoid the distortion we follow following rules:

- Embedding_bit (P0) ≥ 5 and Embedding_bit (P1) ≥ 4
- Embedding_bit (P0) < 5 and Embedding_bit (P2) ≥ 6

Above rule is used to avoid distortion in stego images. Normally hidden data bits are chosen in such a way that its difference in image is not so much recognizable. In original TPVD data is hidden in very high quantity. So as to avoid distortion in the stego image, certain modifications are made in the original TPVD algorithm. The quantity of data is to be hidden is less than the TPVD method but greater than PVD method. Modification is made such that new difference will be less than the original difference. For the difference d ε R_i , the maximum number of secrete bits are selected such that new difference d is less than new difference value. Also boundaries of ranges are overlapped.

A. Embedding Algorithm

In the embedding algorithm, main task is to hide the data by following the steps given below. In embedding first task is partition the image in 2x2 non overlapping blocks. The next task is to find out the difference in each pair of pixels. Out of six different pair we are using 3 pairs in which first pixel is common which is called as reference pixel. Once difference is calculated range of each difference is find out and corresponding number of bits to be hidden is calculated in such a way that the new difference after embedding the bits is always less than the original difference. As the difference is less than original difference so the difference in original image is negligible and not able recognize with compare to original image. Algorithm for embedding is given as below:

1) First find out the difference of three pixel pair

i. $D_0(x,y) = P(x+1,y) – P(x,y)$
ii. $D_1(x,y) = P(x,y+1) – P(x,y)$
iii. $D_2(x,y) = P(x+1,y+1) – P(x,y)$

2) For each D (from D_0-D_2) belongs to R_i, the threshold is find out. Calculate threshold $T_i = U_i – l_{i-1}$ .For embedding the bits following rules are applied.
   i. If $d \geq T_i$ U $i=1$, No overlapping is required but b should be in the range of [0, D- l_i] so magnitude of new difference is less than D. Maximum number of secrete bits are selected so that $b \leq D$- l_i where b is decimal value. New difference is $l_i + b$ as D'
   ii. *If* $d < T_i$, b should be in the range of $[0, u_{i-1} – l_{i-1}]$. Maximum number of secrete bits are b ≤ D- l_i where b is decimal value. If b ≤ D- l_i , the new difference D' *is* $l_i + b$ otherwise D' *is* $l_{i-1} + b$.
3) Modify the values of pixels by using following formula:
   $(P_i , P_{i+1}) = (P_i – [m/2]$, and $P_{i+1} + [m/2])$, Where P_i aid P_{i+1} represented two pixels in P_i and $m = d'_i - d_n$.
4) Find out optimal reference point $P_{i(xy)}$ which is used to offset the other two pixel pairs.
5) Now new block is constructed from the three pixel pair, with embedding secrete data.

B. Extraction Algorithm

In extraction algorithm exactly the opposite procedure is carried out that is in embedding algorithms. Partition the image in blocks of 2x2 pixels. The three pixel pairs are formed again. To retrieve the embedded secret data from the stego-image, the extraction algorithm is described in the following steps.:

1) Partition the stego-image into 2×2 pixel blocks, and the partition order is the same as that in the embedding stage.
2) Calculate the difference values $\boldsymbol{d}_{i(xy)}$ separately for each block in the stego-image given by
   a. $\boldsymbol{D}_0(x,y) = P(x+1,y) – P(x,y)$
   b. $\boldsymbol{D}_1(x,y) = P(x,y+1) – P(x,y)$
   c. $\boldsymbol{D}_2(x,y) = P(x+1,y+1) – P(x,y)$
3) $|\boldsymbol{d}_{i(xy)}|$ is used to locate the suitable range R_i as introduced in the embedding phase. At the same time, the amount of embedding bits t_i, where $t_i = log_2 w_i$ is obtained.
4) After R_i is located , l_i is subtracted from the selected $|\boldsymbol{d}_{i(xy)}|$ and $\boldsymbol{b}_i$ is obtained. If image is to not altered $\boldsymbol{b}_i$ is equal to b_i. Finally, $\boldsymbol{b}_i$ is converted from a decimal value into a binary sequence with t_i bits where $t_i = log_2 w_{ji}$. Note that the t_i -bit stream is only one part of the secret data before embedding.

C. Examble

Consider the four pixel having gray values are (150, 176, 165, and 157). We consider first four pixel pair P_0= (150,176), P_1= (150,165), P_2= (150,157). Data bit stream is consider as 11010101001101.

Now difference calculatd from the pair is 26, 15 7 respectively. Consider the range table of range R_0 (0-7), R_1(8-15), R_2(16-31), R_3(32-63), R_4(64-127), R_5(128-255). Now corresponding range for each difference is R_{20} = [16-31], R_{11}=[8-15], R_{02}=[0-7] respectively(D_0-D_2). Now threshold for D_0 is (31-8)=23; D_1 is (15-0)=15; D_2 is (7-0) =7.

Now For D_0 let b should be in range of [0 – 10] and thus decimal value of bit stream is in the range of [0-10] so the no. of bits to be embedding are 3 bits (110) so new difference D'0 is 16+6=22.

Now for D_1 let b should be in range [0 – 7] and thus decimal value of bit stream is in the range of [0-7] so the no. of bits to be embedding are 3 bits so new difference D'_1=8+5=13.

Now for D_2 let b should be in range [0-7] and thus decimal value of bit stream in the range of [0-7] so the no. of bits to be embedding are 3 bits sow new difference is $D'2$= 0+2=2.

Thus new differences become (22, 13, and 2). For finding the new values we can find m_0=26-22=4, then $P_0+m_0/2$, $P_1-m_0/2$ P_0'=[152,174], similalrly P_1'=[151-164], P_2'= [153,174] and m1= 2, m2= 5. Thus P_1' has the optimal pair.

P_0'= [152+(151-152) 174+(151-152)] m0= 4
P_1'= [151 164] m1= 2
P_2'= [153+(151-153) 155+(151-153)] m2= 5

Now newly embedded block is obtained and given by (151,173, 164, 153).

In this way data is embedded in stego-image.

IV. Analysis of Algorithm

In this section results for the proposed data hiding algorithm are presented. In the proposed method the improved technique of data hiding is presented. In this method quantity of data to be hidden is some what reduced but greatly increases quality of stego image.

Our proposed mentioned approach produces the high capacity of embedding but still it is less than that of original TPVD method but in this approach security provided for data is very high. And cannot be easily recognized by the comparing histogram also.

Below given are the results for hiding a message of approximately 8 kb.

Cover image	PSNR
Lena (512 x 512)	39.57
Baboon (512 x 512)	34.48
Ship (512 x 512)	37.36
Elain (512 x 512)	39.41

(a) (b)

Fig. 3. (a) Before embedding (b) after embedding

V. Conclusion

In this paper the original TPVD algorithm is modified to improve security of the data to be hidden. The data hiding capacity of this algorithm is less than that of TPVD algorithm but quality of the stego image will be better than that of TPVD algorithm. The experimentation is carried out on 8 bit

gray scale images. The algorithm can be modified for 8 and 24 bit color images.

REFERENCES

[1] T. Morkel, J. H. P. Eloff, and M. S. Olivier, "An Overview of Image Steganography," *Information and Computer Security Architecture (ICSA) Research Group*.

[2] H. C. Wu, N. I. Wu, C. S. Tsai, and M S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods," *IEEE Proceedings on Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 611-615, 2005.

[3] K. C. Chang, P. S. Huang, T. M. Tu, and C. P. Chang, "Adaptive Image Steganographic Scheme Based on Tri-way Pixel-Value Differencing," *IEEE 1-4244-0991-8/07©2007 IEEE*.

[4] H. Zhang, G. Geng, and C. Xiong, "Image Steganography using Pixel-Value Differencing," *Second International Symposium on Electronic Commerce and Security*, 2009.

[5] R. R. Ahirwal, D. Ahirwal, and Y. K. Jain, "A High Capacitive and Confidentiality Based Image Steganography using Private stego-key," in *Proc. of the International Conference on Information Science and Applications*, chennai, India, 2010.

[6] K. C. Changa, C. P. Chang, P. S. Huang, and T. M. Tu, "A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing," *Journal of Multimedia*, vol. 3, no. 2, 2008.

[7] A. P. Sherly and P. P. Amritha, "A Compressed Video Steganography using TPVD," *International Journal of Database Management Systems (IJDMS)* vol.2, no.3, August 2010.

[8] N. Zaker, A. Hamzeh, S. D. Katebi, and S. Samavi, "Improving Security of Pixel Value Differencing Steganographic Method," *IEEE* 978-1-4244-6273-5/09/$26.00 ©2009

Mubin Tamboli was born in 1982, Pune, India. He obtained his graduate degree in Computer Engineering from Pune University and obtain masters of degree in Computer Science and Engineering, from BAMU University, Maharashtra, India. Assistance Professor, in Amrutvahini College of Engineering, Sangamner, Pune University, India. His main research interest include Network Security, Distributed Computing and. In these areas he has published above 5 papers in Journal and Conference Proceedings

Avinash Gulve, has obtained his masters degree in Computer Science and Engineering from Allahabad University. Currently he is working as Assistant professor in MCA, Government College of Engineering, Aurangabad. His research interest includes Network security, Steganopgraphy. In this area he has published above 10 papers in Journal and Conference Proceedings.