Volume 4, Issue 4, April 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Block level mod value based Pixel Difference Method for High Capacity Reversible Data Hiding in Images

**Jaimon Jacob*, Muralidharan K B, Sudheep Elayidom**
*Dept of Computer Science*
*Cochin University, India*

**Abstract—** ***This paper presents a novel scheme to hide data in image using Block level mod value based Pixel difference method. The image is first divided into equal sized blocks. The extra space required for embedding the secret data in the image is created by storing the image pixels in less number of bits using mod value based technique.. This data embedding scheme uses the spatial characteristics of the Host image and hence, data embedding capacity purely determined by the Host Image content, where the data embedding take place. Compared with other existing data hiding techniques, this proposed method outperforms in terms of data embedding capacity, and Peak Signal to Noise Ration (PSNR). Hence, this proposed scheme can also be used for embedding a set of images in place of data in a host image.***

**Keywords—** ***Block Level mod value , Contourlet, Histogram, Steganographic***

## I. INTRODUCTION

Many techniques have been devised to hide information inside an image such that the changes made to the cover image are imperceptible to human vision. These Steganographic techniques methods can be broadly classified in to pixel level manipulation methods and transform domain techniques. Common approaches include (a) Least Significant Bit (LSB) manipulation (b) Masking and Filtering (c) Transform techniques (d) Histogram Shifting Based. Out of which least significant bit insertion is the simplest method. But it has the limitation of being very weak in resisting even simple attacks such as transforms, compression, etc. The masking and filtering techniques analyse the image and hide information in significant areas so that the hidden message is more a part of the image than being added noise in the image. The transform technique involves modulating the coefficients of the cover data in the frequency domain. Image hiding techniques that are implemented in frequency domain take advantage of features in human visual system for image. It is based on dividing the image into blocks, intensity histogram of each block is generated and shifting the histograms of each image block between its minimum and maximum frequency. Data are then inserted at the pixel level with the largest frequency to maximize data hiding capacity. The peaks (maxima) of the histograms of the image tiles are then relocated to embed the data.

In this proposed data hiding method, it ensures very high data embedding capacity compared to all other existing methods. Hence, this method can be used for embedding large amount of data or a set of images itself in the host Image. Additional encryption algorithms can also be used for getting another level of security[7]. The embedding capacity is determined by the size of blocks we divide initially the Host Image and the spatial characteristics of the Host Image.

## II. RELATED WORK

Many Reversible Data Hiding (RDH) [1][2] methods have been proposed in recent years, Ni et al.s Histogram Shifting (HS) based algorithm is an important work of RDH, in which the peak value of the Host image histogram[3] is utilized to embed data. In this method, each pixel value is modified at most by 1, and thus the visual quality of marked image is guaranteed. In Lee et al.s proposed a method by using the histogram of difference image. This method outperforms Ni et al.s by improving both Embedding Capacity (EC) and the visual quality. The spatial correlation of natural images is exploited in Lee et al.s method and thus a more accurate histogram is obtained. In the Histogram shifting technique, present a general framework for designing shifting and embedding process[4]. Let S and T be a partition of $Z^n$ : S U T = $Z^n$ and S ∩ T = Ø. Suppose that three functions g : T →$Z^n$, $f_0$ : S →$Z^n$ and $f_1$ : S → $Z^n$ satisfy the following conditions: C1: The functions g, $f_0$ and $f_1$ are injective. C2: The sets g(T), $f_0$(S)and $f_1$(S) are disjointed with each other. Here, g is the function for shifting and used to shift pixel values, $f_0$ and $f_1$ are the functions for embedding data. More specifically, each block with value xϵT will be shifted to g(x), and the block with value xϵS will be expanded to either $f_0(x)$ or $f_1(x)$ to carry one data bit. The shifting and embedding functions will give a HS-based RDH algorithm where the reversibility can be guaranteed by the conditions C1 and C2.

The underflow/overflow is an inevitable problem of RDH, i.e., for gray-scale image, the shifted and expanded values should be restricted in the range of [0; 255]. To deal with this, the above defined sets T and S need be further processed.

In Histogram- Based Reversible Data Hiding Using Block Division scheme[6], dividing the image into blocks, intensity histogram of each block is generated[5] and shifting the histograms of each image block between its minimum

and maximum frequency. Data are then inserted at the pixel level with the largest frequency to maximize data hiding capacity[7][8]. The peaks (maxima) of the histograms of the image tiles are then relocated to embed the data. The gray values of some pixels are therefore modified. High capacity, high fidelity, reversibility and multiple data insertions are the key requirements of data hiding in images. It is shown how histograms of image blocks of images can be exploited to achieve these requirements

In the Data Hiding in Images using Contourlet Transform[9], the image is first contourlet transformed and then text is embedded. The embedding and extraction algorithms are presented in this paper. The text data is first converted to ASCII format and then an encryption algorithm is applied which provides additional security. A digit of data is embedded by modifying the least significant digit of a contourlet coefficient. A high frequency directional pass band from the contourlet transform is selected for data embedding. High capacity can be achieved using this method. The capacity depends on the number of levels in contourlet decomposition and how many subbands we have selected for embedding

## III. PROPOSED METHOD

The proposed method based on block level mod value, is used here to improve the data hiding capacity, by exploiting the spatial characteristics of the blocks in the host image. To make free space for storing the additional secret information, an indexing mechanism is used. The host image is divided into nxn non-overlapping blocks, where n= 8 in the following descriptions. In each block, the maximum frequent pixel is selected which is the mod of that block. Then the original pixel values are replaced with the difference of mod and actual pixel values. An additional data structure is required for storing the overhead information. That is, a table is constructed with the mod value and number of bits required for storing the pixel as the entries. The index to this table is the block numbers. The first entry in the table corresponds to mod value of block 0 and the number of bits required for storing the pixel difference in that block.

The number of bits required is set as maximum(cmin,cmax) + 1 where cmin = $\log_2$(mod - min), cmax = $\log_2$(max - mod), min = minimum gray level in the block, max = maximum gray level in the block. An additional bit is used for storing the sign bit, 0 represent positive, 1 represent negative. If 1 <=cbits <= 5, significant space can be saved. Here, cbits represent the number of bits required for storing the pixel difference. Hence, (8-cbits) x $n^2/8$ bytes available for storing secret information. In other cases where cbits>5, original pixel values are not modified since not much space can be saved. To represent such blocks, the cbits value in index table will set as $(FF)_h$. Prepare a mod index table in the format mod, cbits, with k entries where k represent number of n x n blocks in the host image. Prepare a table of pixel gray value difference from mod value for each pixel in each block using cbits. Signbit = 0 represent positive value, indicates mod value is less than pixel gray value, and Sign bit=1 represent negative value, indicates mode value is greater than pixel gray value. If all pixels in a block are having same gray value, set cbits=$(00)_h$. This entry in the mod index table is enough to represent that block. Hence, n x n bytes can be used for storing the secret information.

Initial Q no of blocks in the host image is used for storing the mod index table where Q = $(2*(M*N)/(n^2))/(n^2)$.

*totalbits=0*
*for i = 1 : k*
*pixeldiff=max(modvalue(i)-minimum(i),maximum(i)-modvalue(i))*
*differencebits = ceil($\log_2$(pixeldiff) + 1)*
*blockbits = differencebits *64*
*totalbits = totalbits + blockbits*
*reqbytes = totalbits/8*

The total number of bytes required for representing the image using pixel difference from mode value can be computed as *reqbytes* bytes as above in the proposed scheme. The total space saved is *spacesaved = M*N-(2*k + reqbytes).*

The host image is divided into three partitions, where each partition is a set of n x n blocks. First partition with Q blocks is used for mod index table. Second partition with *reqbytes/$n^2$* blocks is used for representing the image using pixel difference from mod value. Third partition with (MxN/$n^2$)-(Q+reqbytes/$n^2$ ) blocks is used for storing the secret information.

The proposed method is explained as an algorithm follows.

1) The host image [MxN] is divided into nxn non-overlapping k blocks.
2) Find the mod, min and max value for each block.
3) Let cmin = log2(mod-min) and cmax = log2(max-mod) .
4) cbits = max(cmin,cmax) +1, additional 1 bit is for storing the sign bit.
5) if cbits>5, cbits=(FF)h, represent no change in pixel value.
6) if cbits=0, cbits= (00)h, represent, all pixels in a block are having same gray value,
7) Initial Q no of blocks are used for the mod index table where Q = $(2*(M*N)/(n^2))/(n^2)$
8) Calculate the number of bytes required for representing the pixel difference values

totalbits=0
for i = 1 : k
pixeldiff=max(modvalue(i)-minimum(i),maximum(i)-modvalue(i))
differencebits = ceil(log2(pixeldiff) + 1)

blockbits = differencebits *64
totalbits = totalbits + blockbits
reqbytes = totalbits/8

9) Replace each pixel value with (modvalue-original pixel value) for each block. Store the difference in (cbits+1) bits computed for that block starting from Q + 1 block and ending in Q+1+reqbytes position. If the difference is positive, keep the MSB bit as 0 and if the difference is negative keep it as 1.

11) The space saved by this method is computed as overheadstorageforindextable = (2*k)bytes
spacefordifferencestorage = reqbytes
spacesaved = M*N-(2*k + reqbytes)

12) The space saved by this method is used for embedding secret information. .

## IV. RECONSTRUCTION

1. The received encrypted image will not be in readable format. It contains the image embedded with secret data.
2. The embedded secret information is stored from positions starting from Q + reqbytes +1 position onwards.
3. From the received image retrieve first Q blocks which will give the block-level mod values and cbits (the number of bits required for representing pixel difference in each block)
4. Starting_pos = Q+1
5. Repeat until Starting_pos <Q+1+reqbytes
   - i. Retrieve the cbits for that block from the block index based table where block number is the index to the table
   - ii. If cbits = $(FF)_h$
     It indicates the original pixel values are not changed by the algorithm and keep the block as it is
   - iii. Else if cbits = $(00)_h$
     It indicates all the pixel values are uniform in that block. Hence copy the mod value as the original pixel values in that block. This method can save the entire block sized space while encryption.
   - iv. Else if 1<=cbits<=5
     Take each block's mod value and extract bit patterns of length cbits. Check the sign bit(MSB) and if it is 0, add mod value to the difference (cbits-1) to get the original pixel value. If the sign bit is 1, subtract the difference from the mod value to get the original pixel values.

## V. EXPERIMENTAL RESULT

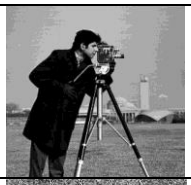
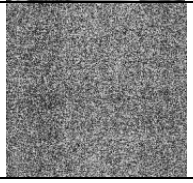
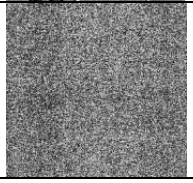
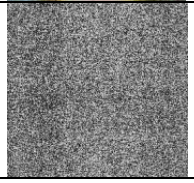
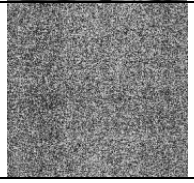
The efficiency of the algorithm is tested by embedding secret data in standard image set. Secret information is embedded in these images applying the 8x8 Block Level mod value based Pixel Difference Method. The result proves Block-level Index based Mode value method shows significant improvement in saving space compared to n-dimensional histogram shifting method which is one of the most promising embedding framework exist so far. The method proves to be reliable, secure and efficient.

TABLE I: COMPARISON BETWEEN PROPOSED SCHEME AND N DIMENSIONAL HISTOGRAM SHIFTING METHOD

| Sample Image | Space Saved | |
|---|---|---|
| | Using n Dimensional Histogram Shifting method | Using Block level mod value based method |
| Lena512.bmp | 4.1204 % | 21.2860 % |
| Cameraman512.jpg | 8.0089 % | 35.1715 % |
| Lion512.jpg | 5.2638 % | 25.7242 % |
| Barbra512.jpg | 6.1204 % | 32.1461 % |

From the table I, it is clear that the proposed block level mod value based method outperform the n-Dimensional shifting method by saving around 5 times more space for storing the secure data. The algorithm is reliable and secure.

TABLE III: OBSERVED RESULT IN STANDARD IMAGE SET

| | | | | |
|---|---|---|---|---|
| Input Image |  |  |  |  |
| Output Image after embedding Secret data |  |  |  |  |

## VI. CONCLUSION

The proposed method proves to be a better solution in saving space for storing secret information. The method uses a mod value based method and stores only the pixel difference from mod values to reduce the storage space. This space saved is used for embedding the information. Experimental results shows the proposed method outperforms the n dimensional histogram shifting method. The method also provides an additional level of encryption while embedding secret data.

**REFERENCES**

[1] Mehdi Fallahpour, D Megias, and Mohammed Ghanbari. *Reversible and high-capacity data hiding in medical images. IETimage processing*, 5(2):190–197, 2011.

[2] Jessica Fridrich, Miroslav Goljan, and Rui Du. *Lossless data embeddingnew paradigm in digital watermarking*. EURASIP Journal on Advances in Signal Processing, 2002(2):185–196, 1900.

[3] Rintu Jose and Gincy Abraham. *A separable reversible data hiding in encrypted image with improved performance.* In Emerging research areas and 2013 international conference on microelectronics, communications and renewable energy AICERA/ICMiCR), 2013 annual international conference on, pages 1–5. IEEE, 2013.

[4] *Xiaolong Li, Bin Li, Bin Yang, and Tieyong Zeng. A general framework to histogram-shifting-based* reversible data hiding.2013

[5] Chia-Chen Lin and Xiao-Long Liu. *A reversible data hiding scheme for block truncation compressions based on histogram modification*. In Genetic and Evolutionary Computing (ICGEC), 2012 Sixth International Conference on, pages 157–160. IEEE, 2012.

[6] PH Pawar and KC Jondhale. *Histogram-based reversible data hiding using block division*. In Advanced Communication Control and Computing Technologies (ICACCCT), 2012 IEEE International Conference on, pages 295–299. IEEE, 2012.

[7] T Sivakumar, G Chithira Rakshmi, and A Ummu Salma. *An approach to reduce the storage requirement for biometric data in aadhar project* (2002)

*[8] A. Xinpeng Zhang. Separable reversible data hiding in encrypted image. Information Forensics and Security, IEEE Transactions on, 7(2):826–832, 2012*

[9] Malini Mohan & Anurenjan P.R, *A New Algorithm for Data Hiding in Images using Contourlet Transform*, IEEE *Transactions on*, 978-1-4244-9477-4*:411-415, 2011*