Histogram Shifting Based Reversible Data Hiding

Lincy Rachel Mathews[1], Arathy C. Haran V.[2],

[1]PG Scholar, [2] Assistant Professor, Dept. of Electronics and Communication, Kerala University

Mar Baselios College of Engineering and Technology, Trivandrum, Kerala, India

***Abstract*— This paper presents a reversible data hiding scheme based on histogram modification. Distribution of pixel difference is used to achieve high hiding capacity. In order to solve the issue of communicating the multiple peak points to the recipients, a binary tree structure is adopted. Data embedding performed after block division facilitates the marked image quality. Histogram shifting technique prevents overflow and underflow problems.**

***Keywords*— Histogram shifting(HS), Least significant bit(LSB)**

I. INTRODUCTION

In the recent years, the development of multimedia technologies and internet has provided the consumers the access to multimedia data. Hence the data may be tampered or attacked during transmission. So it is essential to protect the digital data in applications like military or medical applications [1]. A possible solution to prevent such forgery is data hiding techniques.

Data hiding refers to hide data within a digital media. Media can be anything like audio, image and video. Hiding is done by modifying the contents of the digital media. Hiding process is done in such a way that modification of pixel values should be undetectable to the viewers. Image where the hiding is done is known as the cover media. Image after data hiding is known as watermarked image.

In most hiding techniques, the host image is distorted permanently and thus it cannot be recovered back completely from the marked content. However, in various applications like military and medical applications, degradation of the over media is not allowed. For medical applications, even the slight changes in the image are intolerable. So it is essential to introduce the data hiding in such a way that it is reversible and quality degradation after embedding is lowered. Such reversible data hiding techniques are also known as distortion less or lossless data hiding techniques. Reversible data hiding techniques hides information within the digital media in such a way that only the authorized person could decode the hidden data and restore back to the original state.

II. RELATED WORKS

Most of the data hiding techniques are not reversible completely. The well known LSB technique is not completely reversible due to bit replacement without "memory". Reversible data hiding techniques are mainly classified into three main categories, first is based on data compression[2]-[4],second based on pixel value difference expansion[5]-[7] and third based on histogram shifting[8]-[10]. All these techniques aim at improving the embedding capacity and lowering the distortion.

Earliest technique is based on LSB replacement. In Data compression technique, the data to be embedded as well as related information's of the image used for data recovery is compressed. Compressed data is embedded directly into the cover media using LSB replacement. Celik et al. [4] proposed a data hiding technique in which each image pixel is quantized using L-level scalar technique. The residues yielded after quantization is compressed using a lossless compression algorithm called CALIC. Compressed residues along with the to-be embedded bits are embedded into the quantised image using LSB replacement. Distortions introduced on the watermarked image by this method are comparatively high.

Technique based on pixel value difference expansion discovers the redundancy of pixel values in the images. Arithmetic operations are performed on the pixel pair's in order to explore the space for data embedding. Whether the pixel values can be expanded or not is indicated by the location map. This technique achieves higher embedding capacity and keeps the distortion low. Tian et al[5] proposed the data hiding technique in which data bits are embedded by computing the difference between the pixel pairs. Location map which is essential for image restoration is embedded along with the data bits. Coltuc et al[13]proposed a data hiding technique in which embedding of data bits is done by taking the RCM transform of the image. Reversible contrast mapping is an integer transform which is invertible even if some LSB bits are lost. Data bits are embedded in the LSB bits of such transformed pixel pairs. Complexity of the technique based on pixel value difference expansion is less, thus hacking becomes easier.

Histogram based data hiding technique embeds the data in the cover media by shifting the histogram of the image. Histogram technique finds peak or zero points in the histogram and data embedding is done by shifting these peak and zero points. This technique yields higher data hiding capacity with low distortion. Histogram based reversible data hiding method was introduced by Ni *et al*. in [8], where message is embedded within the histogram. Embedding is done by shifting the peak and zero points of the histogram. Also histogram shifting technique prevents overflow and underflow problem. Overflow is the condition that the gray value exceeds above 255. Underflow is the condition that the gray value falls below 0.

Embedding based on histogram modification has been presented in [11], [12]. One of the major issues associated with all these techniques is that the peak and zero points needs to be embedded along with the data during image embedding for the complete restoration of the image.

The proposed algorithm solves this problem of communicating the peak and zero points by introducing a binary tree structure. Proposed approach also adopts the histogram shifting technique in order to avoid overflow and underflow problem.

III. PROPOSED APPROACH

In the proposed approach, the input image is divided into blocks and then histogram shifting is done on each block which enhances the data hiding capacity and visual quality. Amount of information that can be embedded within image blocks is more as compared with embedding within a single image.

This reversible data hiding technique mainly consists of three main stages:1)Dividing image into two blocks 2)Processing stage and 3)Embedding stage.

First stage consists of dividing the image into two main blocks. Processing stage includes generating the histogram of each block and taking the difference of histogram after histogram modification.

The proposed approach presents a binary tree structure to overcome the drawback of communicating the multiple peak points to the receiver. Also data embedding is done after dividing the image into blocks. There are so many reimbursements while considering the histogram of image blocks than a single image. It is possible to distribute the embedded bits along the whole image. Image blocks have narrower histogram and thus it helps in selecting the suitable peak zero points which may increase the quality of watermarked image. Using the binary tree structure, the number of peak points used for data embedding is assumed to be 2^L, where L represents the level of binary tree. If the pixel difference is less than 2^L, the left child of node d_i is visited if the message bit to be embedded is 0. If the message to be embedded is 1, the right child of node d_i is visited. Binary tree level L that determines the multiple peak point needs to be shared with the recipient for image restoration. Distortion of image increases with increase in tree level L.

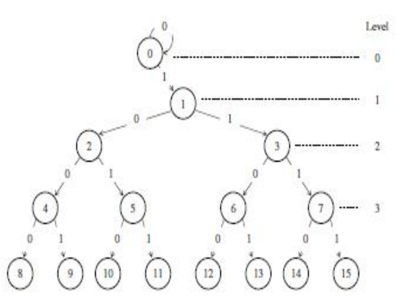

Fig 1 Binary tree structure

Pixel modification cannot be done if the pixel is saturated, that is if overflow or underflow occurs. Overflow means gray value of pixel rises above 255. Underflow means gray value below 0. In order to prevent this overflow or underflow problem, histogram shifting is done that narrows histogram from both sides. Histogram is narrowed down to range 2^L, $255\text{-}2^L$ by shifting the histogram from both sides by 2^L units. This histogram shifting information is embedded along with the message bits.

A. Embedding Process

Consider an N pixel 8 bit gray scale image with pixel value y_i (0-255)

Embedding process is done as follows

1) Divide the image into two blocks

2) Generate the histogram of each block

3) Find the tree level, L of the binary tree

4) For the first block, do the following steps

a) Narrow the histogram in the range 2^L,$255\text{-}2^L$ by shifting the histogram from both sides

b) Scan the image block in the inverse S order and find difference between adjacent pixel values. Let d_i be the difference value

c) Then scan the image block in the same order and if difference value d_i is greater than 2^L, then shifting is done by 2^L units

$$z_i = \{ y_i \text{ , if } i = 0 \text{ or } d_i < 2^L \text{ ,} \quad y_i + 2^L\text{, if } d_i > 2^L \text{ and } y_i \geq y_{i-1}, \quad y_i - 2^L\text{, if } d_i > 2^L \text{ and } y_i < y_{i-1} \}$$

z_i represents the pixels of the watermarked image.

d) If $d_i<2^L$, then message bits are embedded

$$z_i = \{ y_i + (d_i + b) \text{ if } y_i \geq y_{i-1} \quad y_i - (d_i + b) \text{ if } y_i < y_{i-1} \}$$

5) The above steps a)-d) is repeated for the second block.

B. Extraction process

Consider a N pixel 8 bit watermarked image with pixel value z_i. Message bits can be extracted from the watermarked image blocks using the following steps:

For the first image block, do the following steps:

a) Scan the watermarked image block in the inverse S order

b) if $|z_i \, y_{i-1}| < 2^{(L+1),}$ extract message bit *b* by

$$b = \{0, \text{ if } |z_i - y_{i-1}| \text{ is even} \quad 1, \text{ if } |z_i - y_{i-1}| \text{ is odd}\}$$

where y_{i-1} denotes the restored value of z_{i-1}.

c) Original pixel value of host image block is restored by

$$y_i = \{ z_i + \text{floor}(|z_i - y_{i-1}|/2) \quad \text{if } |z_i - y_{i-1}| < 2^{(L+1)} \text{ and } z_i < y_{i-1}$$
$$z_i - \text{floor}((|z_i - x_{iy-1}|/2) \quad \text{if } |z_i - y_{i-1}| < 2^{(L+1)} \text{ and } z_i > y_{i-1}$$
$$z_i + 2^L \quad \text{if } |z_i - y_{i-1}| \geq 2^{(L+1)} \text{ and } z_i < y_{i-1}$$
$$z_i - 2^L \quad \text{if } |z_i - y_{i-1}| \geq 2^{(L+1)} \text{ and } z_i > y_{i-1}$$
$$z_i \quad \text{otherwise} \}$$

d) This process is repeated until all the message bits are extracted from the two image blocks.

IV. RESULTS AND DISCUSSION

Performance of the proposed algorithm is tested for four different datasets of size 512*512 with 8 bit resolution (figure 2). Variation of PSNR for different values of L (0 to 5) is analyzed.

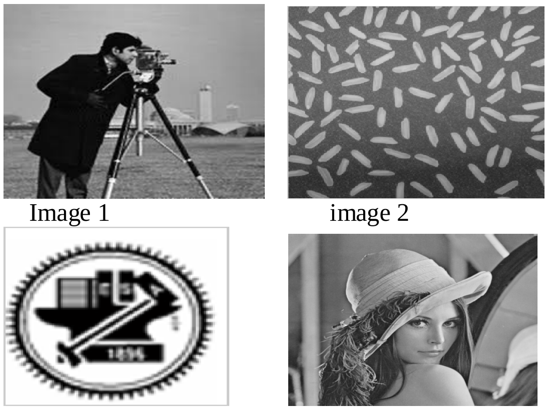
Image 1 image 2
Image 3 Image 4

Fig 2 Data sets

Firstly, image is divided into two blocks and histogram of each block is plotted and after histogram modification, data is embedded into each block using the proposed algorithm.

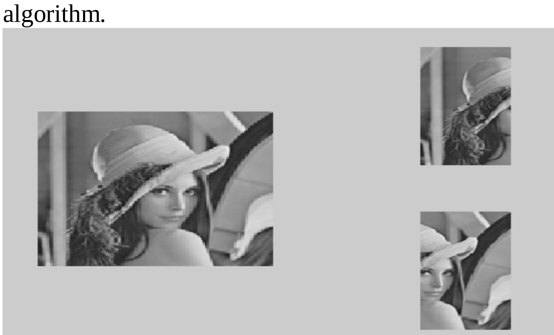

Fig 3 Image after block division

Figure 4 shows the histogram of image 4 after dividing it into two blocks and histogram of them after histogram modification. Histogram of individual image blocks makes it possible to distribute the message bits along the whole image and also improves the image quality.

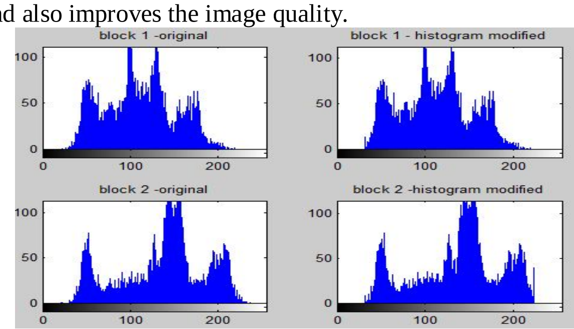

Fig 4:Modified histogram of the input image 4

Table 1 summarizes the variation of PSNR (dB) with tree level L for different images. As table shows, distortion of image increases with rise in the value of L.

TABLE 1 VARIATION OF PSNR(dB) FOR DIFFERENT VALUES OF L

Host image 512*512	Tree level, L					
	0	1	2	3	4	5
Image1	48.8603	43.7978	39.2369	35.0450	30.6390	25.2971
Image2	48.3787	42.7184	37.3611	32.6427	28.9897	26.2350
Image3	48.4895	42.9967	37.8086	33.0630	28.7254	24.9868
Image4	49.4926	43.3546	37.0854	30.9795	25.0216	19.2101

Table 2 shows that PSNR is more when embedding is performed after dividing the image into blocks when compared with the embedding performed in a single image. Thus marked image quality increases after block division

TABLE2: VARIATION OF PSNR(dB) WITH L FOR A SINGLE IMAGE AND IMAGE AFTER BLOCK DIVISION

Tree level	PSNR of Whole image	PSNR of two blocks	Average PSNR after Block Division
0	48.379	48.398	48.388
		48.378	
1	42.718	42.770	42.744
		42.718	
2	37.360	37.467	37.414
		37.361	
3	32.641	32.874	32.758
		32.642	
4	28.988	29.360	29.175
		28.989	
5	26.235	26.553	26.394
		26.235	

V CONCLUSION

This paper proposes a new algorithm for data hiding in which histogram modification technique is done by considering the pixel difference rather than a single pixel. One of the main drawbacks of all the histogram modification techniques is the issue of communicating the multiple peak and zero points. This drawback is overcome in this work using the binary tree structure. Number of peak points is determined by the tree level L. Number of bits that can be embedded is determined by the number of pixels associated with the peak points. Also, in this work data embedding is performed after dividing the image into blocks. This helps to distribute the message bits along the whole image and also improves the hiding capacity. This work can be extended to color images.

ACKNOWLEDGMENT

We would like to thank the almighty for giving us the strength to work on this subject and coming up with this research paper. We are grateful to our families for supporting us and praying for us. We are also thankful to all those who directly or indirectly helped us in making this paper a reality.

REFERENCES

[1] R.Norcen, M.Podesser, A.Pommer, H.Schmidt, and A.Uhl, "*Confidential Storage and Transmission of Medical Image Data*" Computers in Biology and Medicine 33, pp.277–292, 2003.

[2]J. Fridrich, M. Goljan, and R. Du, "Lossless data embedding-new paradigm in digital watermarking," *Eur. Assoc. Signal Process. J. Appl. Signal Process.*, vol. 2002, no. 2, pp. 185–196, Feb. 2002.

[3] T. Kalker and F. M. J. Willems, "Capacity bounds and constructions for reversible data hiding," *Security Watermarking Multimedia Contents V*, vol. 5020, pp. 604–611, Jun. 2003.

[4] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, Feb. 2005.

[5] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890–896, Aug. 2003.

[6] L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet techniques and sorting," *IEEE Trans. Image* Process., vol. 14, no. 12, pp. 2082–2090, Dec. 2005.

[7] D. M. Thodi and J. J. Rodriguez, "Expansion embedding techniques for reversible watermarking," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 721–730, Mar. 2007

[8] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 354–362, Mar. 2006.

[9] S. K. Lee, Y. H. Suh, and Y. S. Ho, "Reversible image authentication based on watermarking," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2006, pp. 1321–1324.

[10] W. Hong, T. S. Chen, and C. W. Shiu, "Reversible data hiding for high quality images using modification of prediction errors," *J. Syst. Softw.*, vol. 82, no. 11, pp. 1833–1842, Nov. 2009.

[11] M. Fallahpour and M. H. Sedaaghi, " High capacity lossless data hiding based on histogram modification," *IEICE Electron. Exp.*, vol. 4, no. 7, pp. 205–210, Apr. 2007.

[12] S. K. Lee, Y. H. Suh, and Y. S. Ho, "Reversible image authentication based on watermarking," in *Proc. IEEE Int. Conf. Multimedia Expo*, Toronto, ON, Canada, Jul. 2006, pp. 1321–1324.

[13]D. Coltuc and J. M. Chassery, "Very fast watermarking by reversible contrast mapping," *IEEE Signal Process. Lett.*, vol. 14, no. 4, pp. 255–258, Apr.2007