Volume 3, No. 1, Jan-Feb 2012
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

# Data Hiding Based On Histogram Modification with Known Binary Tree Level

S.Chandramouli*
Department of ECE, QISCET, Ongole, AP.
chandramoulis4u@gmail.com

DR.CH.Balaswamy
Head of Dept of ECE, QISCET, Ongole, AP.
chbalaswamy7@gmail.com

***Abstract:*** In this paper we store data in a host image and retrieve the very same from the watermarked image. Data is embedded in and extracted from a 8 bit gray scale image. The algorithm we are using is much efficient in retrieving the data and reconstructing the image. In the previous method the peak points of the histogram of the image are used but here we use the peaks of the histogram of the differences of pixels. Hence there is a great improvement in capacity.

***Keywords:*** Host image, watermarked image, Image histogram, histogram peak.

## I. INTRODUCTION

Sending data in disguise of an image opens to wide range of applications such as military, intelligence etc,. But embedding data in the image results in loss of image quality. Such a loss is not acceptable in all cases though the loss is imperceptible to human vision. For example quality loss is unacceptable for medical images because it leads misinterpretation of anomalies. Similar is the case with military images too. Thus by using this scheme we can hide and retrieve data without any loss in the quality of the image along with that large hiding capacities are also possible.

Previously data had been hidden and recovered in audio [4], visible watermarking [6] and in images[7].In [7] the message is embedded into the histogram peak and zero points. The data hiding capacity is low in this process.[8],[9] though have large data hiding capacities suffers from the problem of sending the histogram peak points where the data is being stored. Here in this paper we attain large hiding capacities for data along with complete recovery of the host image. Here we need not send the peak points. We embed the data in the histogram peaks of the pixel differences. We need not send the peak point to the receiver, instead if we specify the level of the binary tree it is enough to recover the image and data.

## II. METHOD

In the previous paper data is embedded in the histogram peaks of the image. To store more peaks should be used and transmitted to the receiver. The bit per pixel (bpp) depends on the number of pixels associated with the peak points. Thus the average hiding capacity was not great for that method.

But here in this scheme we use the modified histogram i.e., the histogram of the differences of the adjacent pixels. As the neighbor pixels are highly correlated, the distribution of pixel differences has a prominent maximum. Here in this scheme we avoid the problem of communicating the peak points to the receiver side by using a binary tree structure and if we can specify the level of the binary tree it will be enough for good reconstruction of image and data.

<u>Procedure for embedding the data in the image with a single modified histogram peak point:</u>

Image information: IMAGE H

a) image type: gray scale image
b) Number of pixels: n
c) bit depth:8
d) Grayscale value of the i th pixel=x(i)
$0<=i<=N-1, X(i) \in Z, x(i) \in 0,255.$

i. Read the image H in the inverse S order and calculate the differences of the adjacent pixels
$d(i)=x(i)$ if $i=0$,
$d(i)=|x(i-1)-x(i)|$ else

ii. Find the peak point of the histogram of the difference matrix d(i) that is P.

iii. Scan the whole image in the same inverse S order. If the difference d(i) is greater than the peak P then shift the pel x(i) by 1 unit.

Here y(i) is the watermarked image which will be transmitted.

$y(i)=x(i)$ if $i=0$ or $d(i)<P$,
$x(i)+1$, if $d(i)>P$ and $x(i)>=x(i-1)$,
$x(i)-1$, if $d(i)>P$ and $x(i)<x(i-1)$

iv. if $d(i)=P$
$y(i)=x(i)+b(k)$, if $d(i)=P$ and $x(i)>=x(i-1)$
$=x(i)-b(k)$, if $d(i)=P$ and $x(i)<x(i-1)$

Where b(k) is the message to be embedded

### A. *At the Receiver end:*

Only the watermarked image y(i) is available
Scan y(i) in the same inverse S order.
$b(k)=0$ if $|y(i)-x(i-1)|=P$
$b(k)=1$ if $|y(i)-x(i-1)|=P+1$
where x(i) denotes the restored value of y(i).

#### a. *Recovering the image:*

$x(i)=y(i)+1$, if $|y(i)-x(i-1)|>P$ and $y(i)<x(i-1)$
$=y(i)-1$, if $|y(i)- x(i-1)|>P$ and $y(i)>x(i-1)$
$=y(i)$ otherwise

Here reconstruction is exact but we need to transmit the peak before.

To attain larger capacities and to prevent sending all the peaks before transmission we will be using a binary tree structure . Details of it are as follows

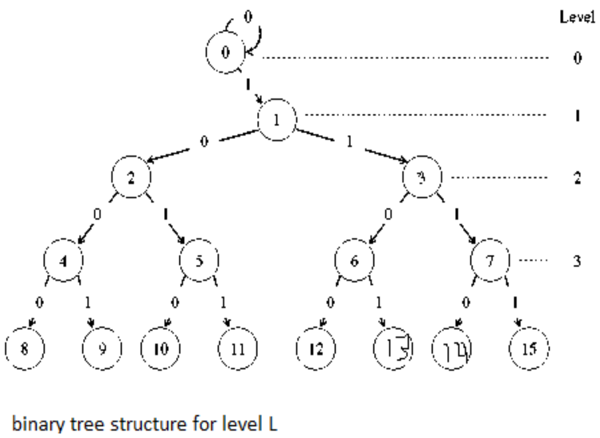


Figure 1 Binary Tree Structure

The figure above shows an auxiliary binary tree for solving the issue of communication of multiple peak points. Each element denotes a peak point. Let us assume that the number of peak points used to embed messages is $2^L$, where $L$ is the level of the binary tree. Once a pixel difference d(i) that satisfies d(i) < $2^L$ is encountered, if the message bit to be embedded is "0," the left child of the node d(i) is visited; otherwise, the right child of the node *d(i)* is visited. Higher payloads require the use of higher tree levels, thus quickly increasing the distortion in the image beyond acceptable levels. However, all the recipient needs to share with the sender is the tree level $L$, because we propose an auxiliary binary tree that predetermines multiple peak points used to embed messages. A detailed embedding algorithm with the auxiliary binary tree is given later in this paper.

## III. HISTORAM SHIFTING

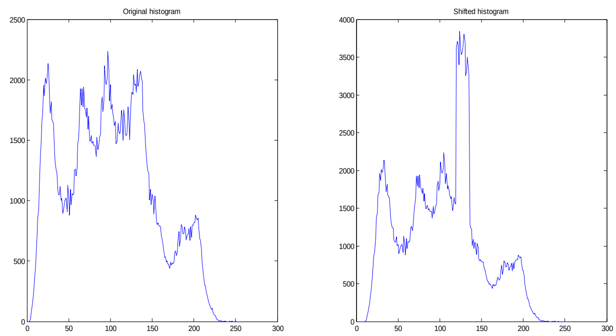


Figure 2 (a):Histogram before shifting (b):Histogram after shifting

Modification of a pixel may not be allowed if the pixel is saturated (0 or 255). To prevent overflow and underflow, we adopt a histogram shifting technique that narrows the histogram from both sides, as shown in Fig above. Let us assume that the number of peak points used to embed messages are having the value $< 2^L$, where $L$ is the level of the proposed binary tree structure. Thus, we shift the histogram from both sides by $2^L$ units to prevent overflow and underflow since the pixel x(i) that satisfies d(i)> $2^L$ will shift by $2^L$ units after embedding takes place. Note that the maximum modification to a pixel is limited to $2^L$ according to the proposed tree structure. As a result, shifting the histogram from both sides by $2^L$ units enables us to avoid the occurrence of overflow and underflow.

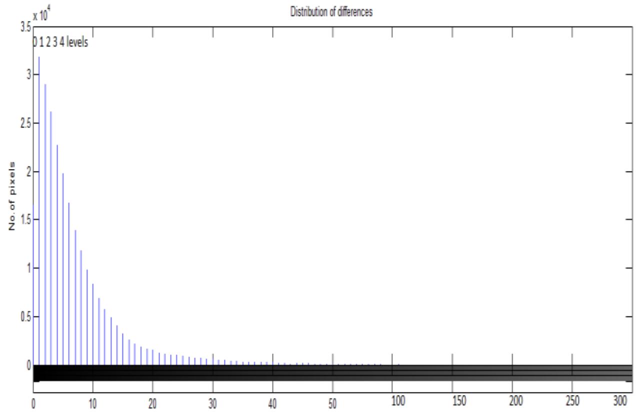


Figure 3: Difference distribution of pixels

### A. *Procedure for embedding the data in the image with multiple difference histogram peak points:*

For an *N*-pixel 8-bit grayscale host image *H* with a pixel value x(i),where x(i) denotes the grayscale value of the *i*th pixel, 0 < *i* < *N* - 1,x(i) ∊**Z**,x(i)g 0,255.

a. Find the level $L$ of the binary tree
b. Shift the histogram from both sides by $2^L$ units.
c. Scan the image *H* in an inverse s-order . Calculate the pixel difference d(i) between pixels x(i-1) and x(i).
d. Again scan the image in the same inverse s-order. If d(i)>=$2^L$

y(i)=x(i), if i=0
=x(i)+ $2^L$ if d(i)>= $2^L$ and x(i)>=x(i-1)
y(i) = x(i)- $2^L$ if d(i)>= $2^L$ and x(i)<x(i-1)

Here y(i) is the watermarked image.

e. Embed the binary message b(k) when d(i)< $2^L$

y(i)=x(i)+(d(i)+b(k)),if x(i)>=x(i-1)
y(i)=x(i)-(d(i)+b(k)),if x(i)<x(i-1)

The number of bits that can be embe-
dded into the image depends on the number of difference values less than $2^L$ .

### B. *Procedure for extracting the data in the image with multiple difference histogram peak points:*

This procedure extracts data from the watermarked image and losslessly recovers the host image. Let $L$ be the level of the proposed binary tree. For an *N*-pixel 8-bit watermarked image *y* with a pixel value y(i), where y(i) denotes the grayscale value.

Step 1) Asusal scan the watermarked image y in an inverse s-order.

2) Extract the original image from the water marked image y Here the image x can be reconstructed pixel by pixel as follows

x(i)=y(i)+(|y(i)-x(i-1)|)/2 if|y(i)-x(i-1)|< $2^{L+1}$ and y<x(i-1)
x(i)=y(i)-(|y(i)-x(i-1)|)/2 if|y(i)-x(i-1)|< $2^{L+1}$ and y>x(i-1)
x(i)=y(i)+ $2^L$ ,if |y(i)-x(i-1)|>= $2^{L+1}$ and y(i)<x(i-1)
x(i)=y(i)- $2^L$ ,if |y(i)-x(i-1)|>= $2^{L+1}$ and y(i)>x(i-1)
x(i)=y(i), otherwise.

3)Bring the data back when the difference
|y(i)-x(i-1)|< $2^{L+1}$
b(k)=0 when |y(i)-x(i-1)| is even
b(k)=1 when |y(i)-x(i-1)| is odd

The above process is applied on every pixel of the watermarked image and the data and image are completely restored.

## IV. EXPERIMENTAL RESULTS

Using the MATLAB code the program for the above stated algorithm is written and the results are noted. All the calculations are carried on 256 by 256 pixel gray scale images. For example 'Lena', 'baboon', 'hat' images are used here.

Table (1):data embedding capacity, bit rate, distortion for a single peak point embedding.

| Image (256 by 256) | Capacity (Bits) | Bit rate (bpp) |
|---|---|---|
| Lena | 9961 | 0.152 |
| Baboon | 2760 | 0.042 |
| Hat | 8604 | 0.131 |

Table (2):Data embedding capacity(bits)

| Host image 256 by 256 | Level 1 Capacity | Level2 capacity | Level3 capacity | Level4 capacity | Level5 capacity |
|---|---|---|---|---|---|
| Lena | 15544 | 30158 | 44115 | 53482 | 59614 |
| Baboon | 4404 | 10446 | 21123 | 36182 | 51267 |
| Hat | 13748 | 27447 | 43957 | 56618 | 61285 |

Table (3):Number of bits embedded per pixel(bpp)

| Host image 256 by 256 | Level 1 Bit rate | Level2 Bit rate | Level3 Bit rate | Level4 Bit rate | Level5 Bitrate |
|---|---|---|---|---|---|
| Lena | 0.237 | 0.466 | 0.673 | 0.816 | 0.909 |
| Baboon | 0.067 | 0.159 | 0.322 | 0.552 | 0.782 |
| Hat | 0.209 | 0.418 | 0.670 | 0.863 | 0.935 |

Table (4):Distortion for a multiple peak point embedding.

| Host image 256 by 256 | Level 1 PSNR | Level2 PSNR | Level3 PSNR | Level4 PSNR | Level5 PSNR |
|---|---|---|---|---|---|
| Lena | 51.455 | 42.462 | 33.502 | 24.643 | 16.151 |
| Baboon | 50.327 | 41.392 | 32.421 | 23.609 | 15.099 |
| Hat | 52.256 | 43.232 | 34.116 | 25.034 | 16.024 |

From the above table 1 the embedding capacity of different images for a single difference histogram peak point can be seen. From that table we can infer that the data hiding capacity of the smooth and less noisy image is higher than that of other images. Therefore the smooth 'Lena' image embeds more data in it than 'Baboon' and 'Hat'.

From the table2 and table3 increase in the capacity performance with increasing level of peak points can be observed. With the increase in the level 'L' there is a similar increase in the capacity. But there is almost a linear drop in the quality i.e., in the PSNR. Thus a compromise between quality and capacity is inevitable.

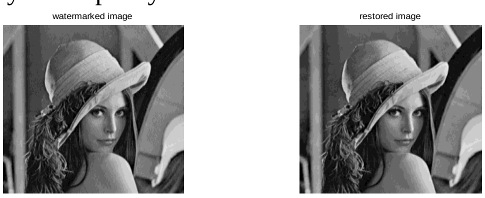


Figure 4 (LENA) (a) L=1(watermarked image) (b) (Restored image)

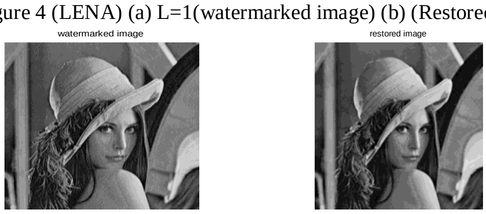


Figure 4 (c) L=5(Watermarked image) (d) (Restored image)

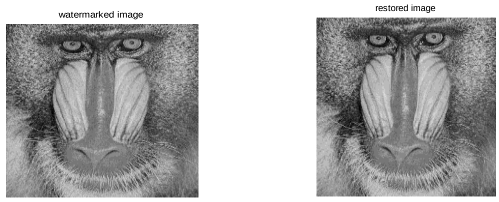


Figure 5 (Baboon)(a)L=1 watermarked image (b)Restored image

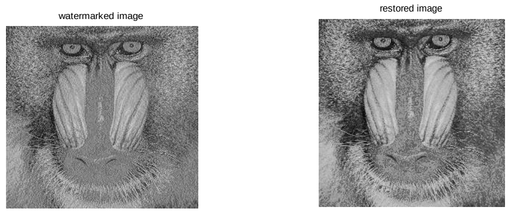


Figure 5 (c) L=5 Watermarked image (d) Restored image

From the fig (4), fig (5) the watermarked image and the restored images of levels 1 and 5 respectively are shown. As in the table the PSNR decreases with the increase in L, same is the case here with the increase in L the quality of the image degrades. Also note that the more smooth 'Lena' embeds more data than the 'baboon'. For easy observation we can draw graphs between Level L and PSNR and also between bit rate and PSNR.

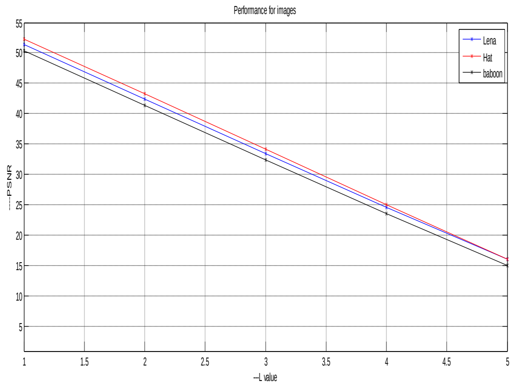


Figure (6a) Graph between PSNR and level L

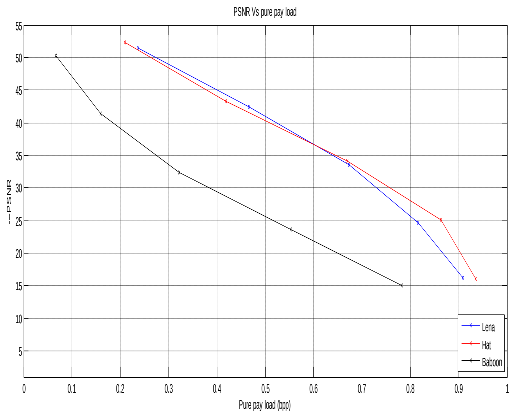


Figure (6b) Graph between PSNR and bit rate

## V. CONCLUSION

Finally when compared to the other schemes where the data hiding capacity is very less in[7],the present scheme is proven to be much better. Even in the schemes defined in[1]-[3] there is a problem in the form of sending increased number of peak points in a parallel channel well in advance. Thus there is a great demand for the current scheme , as this scheme does not require parallel communication of peak points. But here we need to send the level of the binary tree 'L'. This 'L' can be stated as the key for the entire communication .Only those who have 'L' at the receiver can bring back the original image and the embedded data.

In future there is a scope for improvement in the bit rate as in the present scheme it is <1, there is chance to improve the bit rate. At the same time efforts to arrest the image degradation i.e., in the PSNR should be made.

## VI. REFERENCES


[1]. M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, "Lossless generalized-LSB data embedding", IEEE Trans. Image Process.,vol.14, no.2,pp253-256,feb 2005.

[2]. J. Tian, "Reversible data embedding using a difference expansion", IEEE Trans. Circuits Syst. Video technol, vol 13,no.8.pp.890-896 aug2003

[3]. A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform", IEEE Trans. Image Process.,vol.13, No.8,pp.1147-1156,aug 2004.

[4]. M. Van der Veen, F. Bruekers, A. Van Leest, and S. Cavin, "High capacity reversible watermarking For audio", in Proc. SPIE Security Watermarking Multimedia Contents V, Santa Clara,CA,Jan2003, Vol.5020,pp.1-11

[5]. D. Rui and J. Fridrich, "Lossless authentication of MPEG-2 video", in Proc. IEEE Int. Conf. Image Process., vol 2.Rochester, NY, 2002, pp 893-896.

[6]. Y.Hu and B.Jeon, "Reversible visible watermarking and lossless recovery of original images," IEEE Trans. Circuits Syst. Video Technol., vol.16,no.11,pp.1423-1429, nov. 2006.

[7]. 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.

[8]. M. Fallahpur and M.H. Sedaaghi, " High capacity lossless data hiding based on histogram modification", IEICE Electron.Exp.,vol4,no.7,pp.205-210,Apr.2007.

[9]. S.K.Lee, Y.S.Ho, "Reversible image authentication based on watermarking", in Proc. IEEE Int. Conf. Multimedia Expo, Toronto, ON, Canada, jul.2006,pp. 1321-1324.