Two of Three Threshold Visual Cryptography in Block Truncation Coding Scheme with Error Diffusion

Puneeth C N, Shetty Santosh Kumar, Sachin R Naik, Bhaskara Rao N.

Abstract—***A new method of embedding a secret binary data in the Block Truncation Coded image shares is presented. The proposed scheme implements two of three threshold visual encryption. Error reduction is achieved by Floyd-Steinberg Error diffusion.***

Keywords—***Block Truncation Coding, Error Diffusion, Secret Sharing, Threshold Visual Cryptography.***

I. INTRODUCTION

Apart from data compression, Block Truncation Coding (BTC) of images provides data hiding [1], [2], [3], [4]. Hiding a secret watermark in two BTC encoded shares is described in [5]. In this paper, we describe a technique of hiding a secret in three BTC encoded shares such that any two shares combined can recover the secret. This is the 'two of three' threshold visual cryptographic scheme.

II. BASIC BTC

In Block Truncation Coding (BTC), the given gray scale image is divided into non overlapping equal sized blocks and these blocks are processed to generate their respective Binary Blocks (Bit Maps) as follows.

A. BTC Encoding (Binarization) of blocks

Let matrix X represent the block under consideration. Let the mean pixel value and the standard deviation of X be μ and σ respectively. Let the pixel elements of the block at row i and column j be x(i, j) for i =1 to M and j = 1 to N, where M is the number of rows and N is the number of columns of X. Then, the Binarized Block BB of X is obtained using the following Equation.

$$bb(i,j) = \begin{cases} 1, & \text{if } x(i,j) > \mu \\ 0, & \text{otherwise} \end{cases} \qquad (1)$$

for i =1 to M and j = 1 to N.

Here, bb(i, j) is the element of BB at row i and column j. The encoding process also generates two quantization pixel levels called low-mean and high-mean as follows.

The low-mean L is calculated as,

$$L = \mu - \sigma\sqrt{\frac{q}{M*N-q}} \qquad (2)$$

and the high-mean H is calculated as,

$$H = \mu + \sigma\sqrt{\frac{M*N-q}{q}} \qquad (3)$$

Here, q is the number of 1's in the binarized block B and M*N is the total number of elements in B.

B. BTC Decoding (Reconstruction)

The Binarized Block BB is decoded by substituting for its 1's and 0's to get the reconstructed block RB, as follows.

$$rb(i,j) = \begin{cases} H, & \text{if } bb(i,j) = 1 \\ L, & \text{if } bb(i,j) = 0 \end{cases} \qquad (4)$$

for i =1 to M and j = 1 to N.

The H and L values as given by Eqs. (2) and (3) are so chosen that the mean and standard deviation of RB are same as those of X. The BTC encoding (binarization) and decoding (reconstruction) process can be represented as shown in Fig.1.

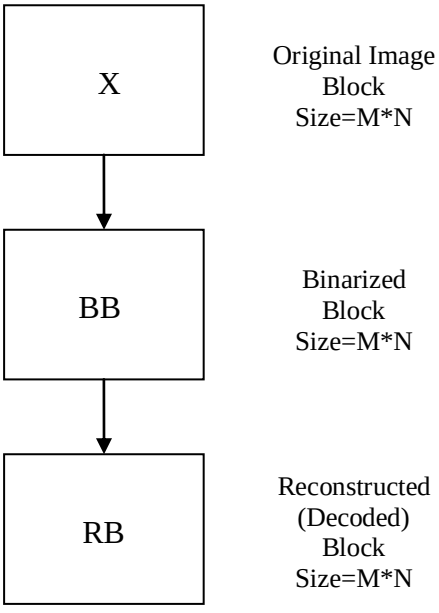


Fig.1. Basic BTC operation

Here, RB is a two level quantized approximation of X.

C. Special case when all the elements of X are equal

In this case u = x(i, j) for all i's and j's and σ = 0. Hence, from Eq. (1), bb(i, j) = 0 for all i's and j's. Therefore BB is an all zero block. In this

Manuscript Received May, 2013.

Prof. Bhaskar Rao N, Computer Science Department, Dayananda Sagar College of Engineering, Bangalore.
Puneeth C N, 8th semester, Computer Science Department,Dayananda Sagar College of Engineering, Bangalore.
Sachin R Naik, 8th semester, Computer Science Department, Dayananda Sagar College of Engineering, Bangalore.
Shetty Santosh Kumar, 8th semester, Computer Science Department, Dayananda Sagar College of Engineering, Bangalore

case, from Eqs. (2) and (3), L = H =μ. Since bb(i, j)'s are all zeros, the reconstructed block C according to Eq. (4) would be all L's. Thus when all the elements of C are same, the corresponding Binary Block is an all zero matrix. In this case C and X are equal.

Example 1. Consider a 3*3 image block X with values as shown in Table 1a. The mean μ and standard deviation σ of X are calculated and found to be, $\mu = 34.77$ and $\sigma = 10.98$. The Binarized Block BB, found using Eq. (1) is shown in Table 1b. The low-mean and high-mean are found to be 22.49 and 44.60. On rounding off, L = 22 and H =45. The decoded
(reconstructed)block C is shown in Table 1c.

Table 1a. Image Block X.

21	53	36
39	24	20
48	40	32

Table 1b. Binarized Block BB.

0	1	1
1	0	0
1	1	0

Table 1c. Decoded Block C.

22	45	45
45	22	22
45	45	22

The BTC operation is applied to all the blocks of the given image to get the corresponding Binarized Blocks. From these Binarized Blocks, the BTC decoded image is reconstructed. The aggregation of the Binarized Blocks in the corresponding order is designated as the Binarized Image which is used in data hiding operations.

III.HIDING A SECRET IN TWO SHARES

Let us consider the case of hiding a binary secret image W in two shares of a BTC encoded image. Let I be the original gray scale image of size m*n. Let the Binarized Image of I be designated as B. The size of B is same as that of I. One way to hide W in two shares is to create two binary shares from B as follows.

A. Case 1, Embedding w in B2

B1 = B (5)

and B2 = B xor W (6)

Here, we assume the size of W is also m*n which is same as the size of B. Now consider the xor result of B1 and B2. From Eqs. (5) and (6),

B1 xor B2 = B xor (B xor W) (7)

The RHS side of Eq. (8) is simply W itself.

Therefore W = B1 xor B2 (8)

Thus W can be recovered from the shares B1 and B2. Here, the secret data W is hidden in B2, without affecting B1.

B. Case21, Embedding w in B1

A second way is to hide W in B1 without affecting B2 as follows.

B1 = B xor W (9)

and B2 = B (10)

Here also, W is recovered using Eq. (8).

In the data hiding scheme, B1 and B2 are decoded to get R1 and R2 based on Eq. (4). R1 and R2 are the quantized approximations of the original image I and they constitute the 2 final shares.

C. Recovery of B1, B2 and W from R1 and R2

Each block of R1 or R2 is a two level block. Each block is binarized by replacing the H (higher) value by 1's and L (lower) values by 0's. In the special case where a single value is present in the block, In the special case where a single value is present in the block, its binary representation is an all zero matrix. The binarized matrices of R1 and R2 are B1 and B2 respectively. After getting B1 and B2, Eq. (8) is used to recover W.

D. Error due to data hiding

When W is hidden in B2, see Eqs. (5) and (6)), B2 is different from B, and thus B2 carries the error. The error E2 between B2 and B is obtained by getting the bit differences between B2 and B. This can be expressed as,

E2 = B2 xor B (11)

This is same as W as seen by Eq. (8). Thus,

E2 = W (12)

The magnitude of the error is obtained by counting the number of error bits in E2 which is same as the number of 1's in E2 or W. That is,

En2 = the number of error bits in E2
= the number of 1's in W =Wn (13)

Here Wn represents the number 1's in W.

Therefore for low error magnitude, the number of 1's (white pixels) in W should be small compared to its size m*n.

If W is embedded in B1 as given by Eqs. (9) and (10), then also the error magnitude in B1, represented by nE1 would be,

En1 = the number of 1's in W=Wn (14).

In case 1, where W is embedded in B2, all the error is in B2 and in case 2, where W is embedded in B1 all the error is present in B1. Thus the error is unequally distributed in either case. To overcome this disadvantage, a modified scheme is presented where 50% of W is embedded in B1 and the remaining 50 % in B2. This scheme eventually generates 3 shares which provide 'two of three' visual encryption.

IV.TWO OF THREE SCHEME

The three shares S1, S2 and S3 are generated from the binarized Image B of original image I as follows.

S1 = B
S2 = B embedded with 50% white pixels of W
S3 = B embedded with the other 50% white pixels of W.

The embedding operation of the white pixels of W into S2 and S3 is done by *alternate pixel interleaving* as described below.

A. Alternate White Pixel Embedding

The white pixels of W are split alternatively into two equal non over lapping divisions and stored in matrix W1 and W2 respectively. W1 and W2 have the same size as that of W. The location of a white pixel of W occupies the same location (row-column position) when stored either in W1 or W2. When a white pixel from W goes into W1, its 4-connected neighbourhood white pixels in W go into W2 and vice-versa.

Alternative white pixels from W are extracted using a checkerboard mask which provides both horizontal and vertical alternative distribution of white pixels A checker board matrix has alternating 0's and 1's along both rows and columns. A 6*6 checker board matrix is shown in Table 2. In our method, the checker board matrix C of the same size as that of W is used to extract W1 and W2 from W using logical *and* operation as follows.

W1 = and(W, C) (15)

W2 = and(W, ~C) (16)

Here, ~C is the complement of C

Table 2. 6*6 checkerboard matrix

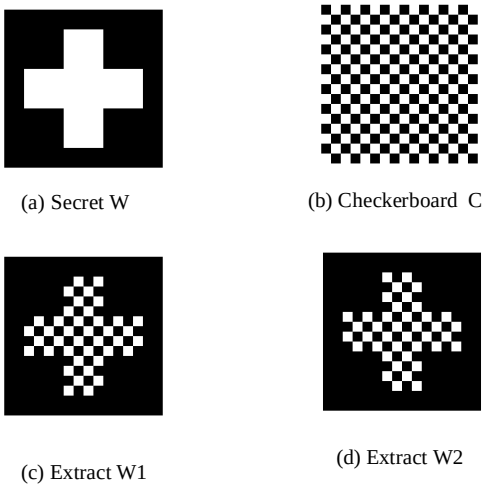
0	1	0	1	0	1
1	0	1	0	1	0

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

0	1	0	1	0	1
1	0	1	0	1	0
0	1	0	1	0	1
1	0	1	0	1	0

The logical *and* operation of W and C in Eq. (15), extracts those white pixels (1's) of W which are common to both W and C. Since ~C is the complement of C, *and* operation in Eq. (16) extracts the remaining white pixels.

Example 2. A 16*16 sized W is shown in Fig. 2(a). Same sized checkerboard C is shown in Fig. 2(b).



(a) Secret W (b) Checkerboard C

(c) Extract W1 (d) Extract W2

Fig.2. Alternate white pixel Extraction From W

In an given region of the checkerboard, the number of 1's is almost 50% of the total number of pixels. Therefore the number of white pixel in W1 will be approximately 50% of W. Similarly, for W2, the number will be about 50% of W. When alternate white pixels are extracted to W1 and W2, the shape of the secret image W is retained in W1 and W2 with super imposed checkerboard mask. The user can easily identify the object in W1 and W2. W1 and W2 are almost same in shape and size.

B. Generation of 3 Shares

Now consider the generation of 3 shares as mentioned at the beginning of this section. S1 is simply obtained as B.

$$S1 = B \qquad (17)$$

S2 is obtained as the result of xor operation of B and W1.

$$S2 = B \text{ xor } W1 \qquad (18)$$

S3 is obtained as the result of xor operation of B and W2.

$$S3 = B \text{ xor } W2 \qquad (19)$$

Here the error is distributed equally between S2 and S3 and the error magnitude is 50% compared to the case of unequal sharing as described in section III.

C. recovery of W from the shares S1, S2 and S3

Now the recovery of W from the shares S1, S2 and S3 is done using xor operations as follows.

Consider S2 xor S3. Substituting for S2 and S3 from Eqs. (18) and (19),

$$S2 \text{ xor } S3 = (B \text{ xor } W1) \text{ xor } (B \text{ xor } W2)$$

On simplification of the RHS, B gets cancelled and we get,

$$S2 \text{ xor } S3 = W1 \text{ xor } W2 \qquad (20)$$

Substituting for W1 and W2 from Eqs. (15) and (16) and further Boolean simplification leads to,

$$S2 \text{ xor } S3 = W$$

Or $$W = S2 \text{ xor } S3 \qquad (21)$$

Thus W is fully recovered using Eq. (21).

Now, consider S1 xor S2. From Eqs. (17) and (18),

$$S1 \text{ xor } S2 = B \text{ xor } (B \text{ xor } W1)$$

On simplification, S1 xor S2 = W1. Therefore,

$$W1 = S1 \text{ xor } S2 \qquad (22)$$

Here the recovered secret W1is 50% of W. But the shape is retained and the user can easily recognise the secret object. Similar to Eq. (22), W2 is recovered from S1 and S3 as,

$$W2 = S1 \text{ xor } S3. \qquad (23)$$

Because of 50% representation, the effective brightness of W1 and W2 will be 50 % of the brightness of W.

In a data hiding scheme, S1, S2 and S3 are not the final shares. They are the intermediate formats. The final shares are obtained using the BTC reconstruction of these binary shares. Let R1, R2 and R3 be the BTC reconstructs of S1, S2 and S3 as indicated by Eq.(4). Then the images R1, R2 and R3 look almost same as the original image I, except for the BTC error and hiding error. The visual appearance of any final share camouflages the hidden data.

D. Algorithm for two of three data hiding and recovery

Two of three threshold visual cryptographic share generation and secret recovery can be described as follows.

1) Share Generation (Data Hiding)

Algorithm 1. Input: Gray scale Image I and the binary secret (watermark) W.

Output: Three gray scale shares R1, R2 and R3.

1. Get the BTC binarized image B of I as described in section II.
2. Create a black and white checker board mask C of size same as that of I.
3. Get W1 and W2 from W using Eqs. (15) and (16) as,
   W1 = and(W, C)
   W2 = and(W, ~C)
4. Create S1, S2 and S3 as,
   S1= B
   S2 = (B xor W1)
   S3 = (B xor W2)
5. From S1, S2 and S3 reconstruct the corresponding BTC quantized gray scale images R1, R2 and R3 respectively. Use the relation of Eq.(4).

These are the final shares for two of three visual cryptography.

2) Secret recovery

Algorithm 2.

Input: Any two shares out of S1, S2 and S3.

Output: Secret Images WR23, W12, and W13.

1. Convert the given two shares block by block, into their respective BTC binary equivalents as described in section III.C. Call them as Bi and Bj.
2. Recover the secret image as WRij = Bi xor Bj.

WRij will be either W or W1 or W2 depending on the shares Bi and Bj as dictated by Eqs. (21), (22) and (23). Thus,

WR23 = B2 xor B3,
WR12 = B1 xor B2,
WR13 = B1 xor B3.

When there is no error,

WR23 = W,
WR12 = W1
WR13 = W2.

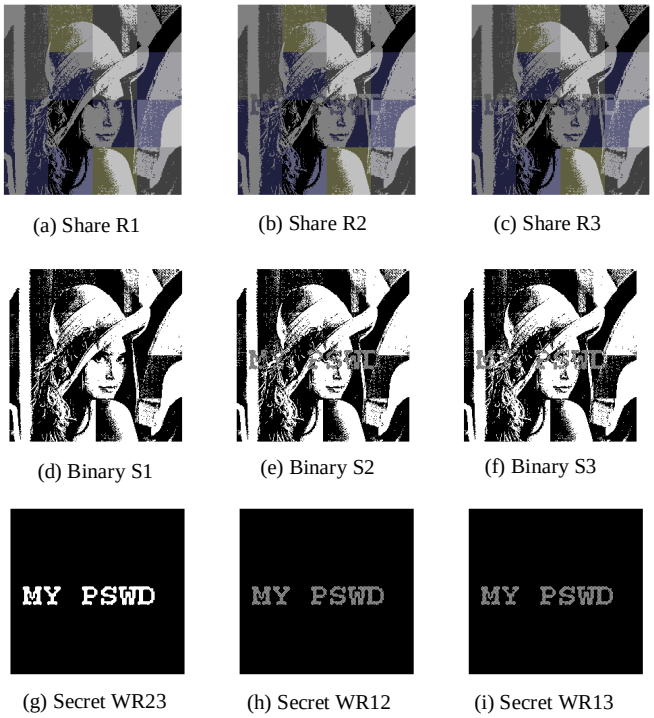
E. Error reduction by diffusion

When a secret is hidden in a binary share, the corresponding bit values are complemented. This in turn contributes additional errors in the BTC reconstructed shares because of the quantization process involved in BTC. To reduce the severity of the truncation error, Error Diffusion techniques can be adopted [5], [6], [7]. In this paper we have adopted Floyd Error diffusion technique to reduce the effect of quantization error. Since the technique is well known, it is not discussed here. The Error diffusion corrections are applied to those pixels affected by data hiding operation, because each white pixel of W1/W2 complements the corresponding BTC bit in the share. This results in the interchange of L and H values in the BTC encoding process. Because of error diffusion the edges and contours of the hidden data are diffused. The results of Error diffusion are given in the next section.

V. TEST RESULTS

Example 3. A secret black and white image W containing the word "MY PSWD" is embedded in 3 shares of image 'Lena" of size 256*256. The results are shown in Fig. 3. Here a relatively large BTC block size is chosen at 64*64 to highlight the error due to larger block sizes.

Fig. 3(a) shows the BTC reconstruct share R1 without any embedded secret (watermark). Corresponding Binarized image S1 is shown in Fig. 3(d). Figures 3(b) and 3(c) show the embedded reconstructed shares R2 and R3. Corresponding Binarized images S2 and S3 are shown in Fig. 3(e) and 3(f). Recovered secrets by S2 and S3 is shown in Fig. 3(g) which is the full recovery of w. W1 which is the xor of S1 and S2 is shown in Fig. 3(h). Only 50% of shape retaining W is seen here. Similarly, Fig. 3(i) shows W2. The brightness levels of W1 and W2 are 50% of W. In this case the hidden secret is vaguely visible in Fig. 3(b) and (c) and more prominently in Fig. 3(e) and 3(f).



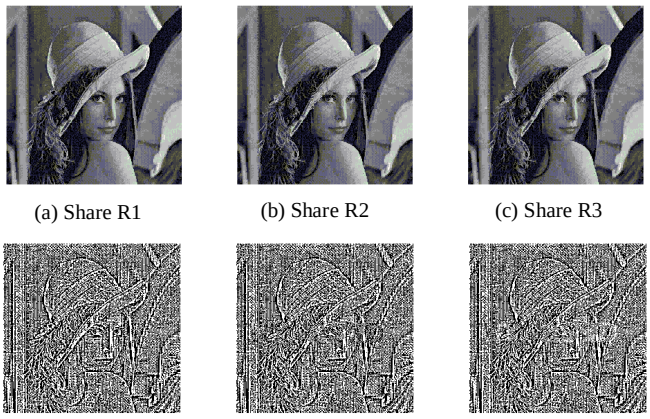
(a) Share R1 (b) Share R2 (c) Share R3

(d) Binary S1 (e) Binary S2 (f) Binary S3

(g) Secret WR23 (h) Secret WR12 (i) Secret WR13

Fig. 3. BTC Shares, their Binary Formats and the Recovered Secret Images. Block Size **64*64**.

This visibility of hidden object in the shares can be almost eliminated by choosing a smaller block size for BTC. This is demonstrated in Fig. (4), where a BTC block size of 4*4 is used.



(a) Share R1 (b) Share R2 (c) Share R3

(d) Binary S1 (e) Binary S2 (f) Binary S3

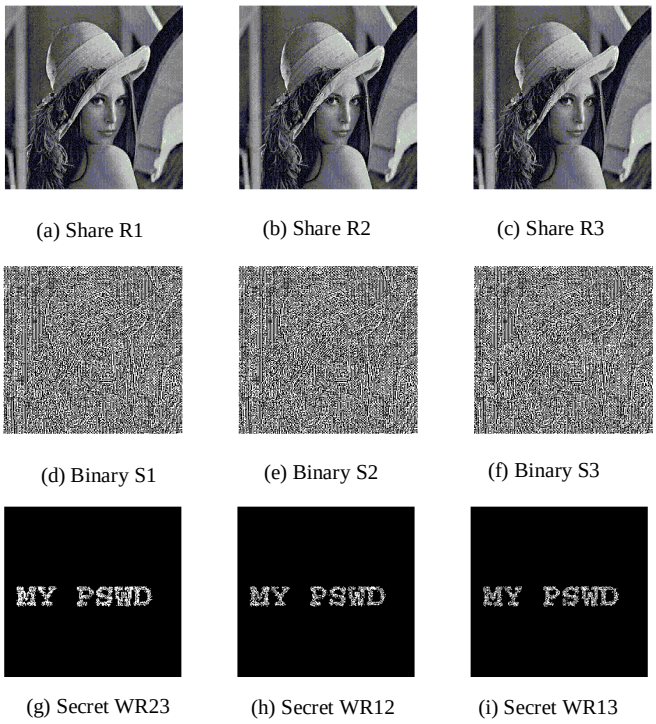
Fig. 4. BTC Shares, their Binary Formats and the Recovered Secret

In Fig.(4) Recovered Secret set is not shown, because it remains same irrespective of the block size.

When the block size decreases, the magnitude of the quantization error also decreases. But the time required for data hiding/recovery increases as the block size decreases.

A. Results with Error diffusion

The image shares, binary images and recovered secret images with Error Diffusion are shown in Fig. (5) in which the binary images have lost their edge sharpness.



(a) Share R1 (b) Share R2 (c) Share R3

(d) Binary S1 (e) Binary S2 (f) Binary S3

(g) Secret WR23 (h) Secret WR12 (i) Secret WR13

Fig. 5.BTC Shares, their Binary Formats and the Recovered Secret Images. Block Size **4*4**, with Error Diffusion

Because of Error Diffusion, The recovered secret also gets diffused. Then the error in the recovered secret WR can be expressed using the Correct Decode Rate (CDR) defined as [5],

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$\text{CDR} = \frac{\text{number of elements in (WR xnor W)}}{\text{number of elements in W}} \qquad (24)$$

CDR calculated for different block sizes is given in Table 3. The size of W is 256*256.

Table 3. CDR values for different block sizes

Block size	CDR WR23	CDR WR12	CDR WR13
4*4	0.7492	0.5694	0.5742
4*2	0.7286	0.5809	0.5786
2*4	0.7341	0.5912	0.5738
2*2	0.7016	0.5786	0.5627

When the block size is reduced, CDR of WR23 decreases slightly. The CDR's of WR12 and WR13 do not change much.

The MSE between the original image I and the BTC data hidden shares for different block sizes are shown in Table 4.

Table 4. MSE values for different block sizes.

Block size	MSE R1	MSE R2	MSE R3
4*4	140.6	219.8	212.3
4*2	91.5	157.4	149.6
2*4	107.6	186.6	178.9
2*2	45.5	104.2	92.5

From Table 4, it can be seen that MSE decreases as the block size decreases.

VI. CONCLUSION

A new technique of hiding data in three BTC shares to realize two of three threshold encryption is presented. The technique can be used with or without Error Diffusion.

REFERENCES

[1] Nimrod Peleg, "Block Truncation Coding" *cs.haifa.ac.il/~nimrod/Compression/Image/I5btc2005.pdf.*
[2] Pasi Fränti, Olli Nevalainen and Timo Kaukoranta," Compression of Digital Images by Block Truncation Coding: A Survey". The Computer Journal, 37 (4), 308-332, 1994.
[3] Doaa Mohammed, Fatma Abou-Chadi, "Image Compression Using Block Truncation Coding", Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), February Edition, 2011.
[4] Block Truncation Coding - Wikipedia, the free encyclopedia *en.wikipedia.org/wiki/Block_Truncation_Coding.*
[5] Jing-Ming Guo and Yun-Fu Liu," High Capacity Data Hiding for Error Diffused Block Truncation Coding", IEEE transactions on image processing, vol. 21, no. 12, december pp.4808-4818, 2012.
[6] Error diffusion - Wikipedia, the free encyclopedia **en.wikipedia.org/wiki/Error_diffusion.**
[7] Halftoning by Error Diffusion www.ece.utexas.edu/~bevans/projects/.../talks/ErrorDiffusion.html