Chapter 3

Block Truncation Coding based Efficient Watermarking Scheme for Image Authentication with Recovery Capability

3.1 Introduction

In these days image authentication and restoration has become more important due to the rapid growth of Internet technology, medical imaging applications, electronic commerce and multimedia technologies. The era of the Internet eased the fast transmission and exchange of huge amount of information without any bound. However, multimedia information in digital format can be tampered or modified (intentionally or unintentionally) with ease using a lot of image processing tools [111, 112]. So, it is a big problem to ensure the integrity of received images for potential security loopholes of the public Internet.

A digital image is considered authentic if and only if its each pixels remain unchanged. The integrity and authenticity of digital images can be assured by using digital image watermarking. It is a technique to embed secret information into an image. The secret information is known as watermark and resultant image is known

as watermarked or stego image. The watermark can later be extracted by using a pre-designed extracting scheme for various purposes, including content integrity verification, ownership authentication, ownership assertion and so forth [3, 4, 113].

As previously mentioned in chapter 2, watermarking schemes can be broadly classified into three categories: robust, semi-fragile and fragile. Robust watermarking schemes are designed to resist intentional or unintentional manipulations. So, robust watermarking schemes are usually used for copyright protection and ownership verification [7, 8, 114, 115, 116]. A fragile watermark is intended to be destroyed even after the minor unintentional or intentional manipulation in the watermarked image [8, 10, 64, 117]. The third category, semi-fragile watermarks are designed for detecting any unauthorized manipulations, while allowing some general image processing like JPEG compression [11, 12, 13, 15, 118]. The main application of fragile and semi-fragile watermarking is image and video content authentication. In addition, watermarking schemes with the capability of tampered recovery, as well as tamper detection, are also desirable for protection of the content and the integrity of images.

In the past decades many watermarking schemes have been proposed for verifying image authenticity and recovery. Some are remarkable in various aspects while lacking at others. In [62], He et al. proposed a watermarking scheme based on adjacent-block based statistical detection method (SDM). In this scheme discrete cosine transform (DCT) was applied on the non-overlapping block of size 8×8, and the first eleven DCT coefficients of a block were embedded into the corresponding mapped block. To determine the validity of a block, a statistical method was used by considering its adjacent blocks and its mapping block. If only a single pixel in a block of size 8×8 pixels is tampered, the whole block will be marked as invalid. So, the accuracy of tamper localization gets decreased because of its large block size. The scheme only works under the circumstances that regions storing the original information of tampered areas must be preserved. So, this scheme suffers from tampering coincidence problem. The schemes presented in [21, 63, 67] are capable to resolve the tampering coincidence problem. A similar DCT based approach was proposed by Qian et al. [64]. Several DCT coefficients were encoded with the variable-length coding and embedded into the three least significant bits (LSBs). The length of the code is determined by the type of the block. This method recovers

with high-quality when the tamper areas are less than 35 % ; but accuracy of tamper localization is low due to large block size as in [62]. In [77], Zhang and Wang proposed a hierarchical fragile watermarking scheme using a lossless difference expansion (DE) technique, which can recover the tampered regions without any errors. The main drawback of this approach is tampered regions must be less than 3.2 % of the total area. Zhang et al. proposed a DCT based fragile watermarking scheme for successful restoration of extensive tampering demonstrated in [65]. This scheme is capable to recover the tampered regions upto tampering rate 59 %. In [63], Korus and Dziech proposed a self-recovery watermarking scheme based on an erasure communication channel. This scheme is capable to recover even when 50 % of the image area becomes tampered. Two self-embedding watermarking schemes were proposed by Zhang et al. [66], called as a reference sharing mechanism. In the first scheme,the reference in-formation was derived from 5 MSBs-planes. In the second scheme, the cover image was decomposed into three levels based on the hierarchical self-embedding scheme defines a three-part scalable reference stream.The first scheme, the original data in five layers of original watermarked image can be recovered when tampering rate is no more than 24 %. In the second scheme, the reference sharing methods with different restoration capabilities are employed to protect the data at different levels. Hence, a better restored image can be obtained in second scheme, from a tampered version with less fake content.

This chapter presents a new self-embedding fragile watermarking scheme with flexible recovery quality and higher localization accuracy. Here, Block Truncation Coding (BTC) is employed for tampered regions recovery. The proposed scheme is capable to perform under extensive tampering with high quality restoration. The advantages of using this scheme is based on the usage of small block size (2×2). This simple fact results in various benefits like production of higher localization accuracy and reduction of blocking artifact problem. The presented scheme is also secured using three predefined keys. Experimental results show that up to 50 % tampering rate, it allows reconstruction with high PSNR and NCC values which is effective enough for fidelity.

The remaining part of this chapter is organized as follows. The key concepts of BTC is outlined in Section 3.2. Section 3.3 gives the detailed methodology of proposed watermark embedding procedure and extraction procedure. Experimental results

along with performance analysis are given in Section 3.4. Finally, the concluding remarks are given in Section 3.5.

3.2 Theoretical Foundation Overview

Block truncation coding (BTC) is a spatial domain coding technique introduced by Delp and Mitchell [119]. BTC is a simple, fast and fixed length lossy compression technique for gray scale images [120, 121, 122, 123, 124]. This is a block-adaptive binary encoding scheme based on moment preserving quantization. BTC never requires any supplementary information during the encoding and decoding procedures. In addition, BTC-compressed images usually maintain acceptable visual quality, and the output can be compressed further by using other lossless compression methods.

In the simplest form of BTC, the first two moments are preserved and blocks are represented by two quantization levels. In BTC method, an image is divided into non-overlapping blocks of size $n \times n$ and each block is coded separately. For each block B, first two sample moments m_1 and m_2 are calculated and defined as follow:

$$m_1 = \frac{1}{n \times n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \tag{3.1}$$

$$m_2 = \frac{1}{n \times n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f^2(x, y) \tag{3.2}$$

Where $f(x, y)$ indicates the pixel value in the position (x, y) of the block. The standard deviation σ of the block B can be calculated and defined as follow:

$$\sigma = \sqrt{m_2 - m_1^2} \tag{3.3}$$

Then a two-level quantization is performed on the block. The pixels with intensity greater than the quantization threshold are quantized to value H, and the other pixels are quantized to value L. Here, the first moment m_1 is used as the quantization threshold. The quantization partition of B with respect to the threshold m_1 gives two sets of pixels B_0 and B_1, such that $B_0 \bigcup B_1 = B$ and $B_0 \bigcap B_1 = \phi$. The

quantization partition of B can be represented by a binary-pattern b. This binary pattern b is generated by the using following rule:

$$b(x,y) = \begin{cases} 0 & \text{if } f(x,y) \leq m_1 \\ 1 & \text{if } f(x,y) > m_1 \end{cases} \tag{3.4}$$

The partition of B is defined as $B_0 = \{f(x,y)|b(x,y) = 0\}$ and $B_1 = \{f(x,y)|b(x,y) = 1\}$.

Suppose q pixels in the block are greater than m_1. The quantization levels H and L are used to preserve first two sample moments. Then

$$n^2 * m_1 = (n^2 - q) * L + q * H \tag{3.5}$$

$$n^2 * m_2 = (n^2 - q) * L^2 + q * H^2 \tag{3.6}$$

and solving for H and L yields

$$L = m_1 - \sigma * \sqrt{\frac{q}{n^2 - q}} \tag{3.7}$$

$$H = m_1 + \sigma * \sqrt{\frac{n^2 - q}{q}} \tag{3.8}$$

By using BTC a compressed image block B is represented by the triplet (b, L, H). The intensity $\hat{f}(x,y)$ of the pixels of the corresponding block of the reconstructed image is given by

$$\hat{f}(x,y) = \begin{cases} L & \text{if } b(x,y) \in B_0 \\ H & \text{if } b(x,y) \in B_1 \end{cases} \tag{3.9}$$

3.3 Proposed Methodology

In this proposed scheme, the five most significant bits (MSBs) of all pixels in the host image are kept unchanged, while the three least significant bits (LSBs) of each pixel are replaced with the watermark. The watermark is determined by the five MSBs planes of the host image and divided into two part: authentication data and recovery data. They are used to locate and detect the tampered blocks and to restore

the original content respectively. This proposed scheme can be described as the following 3-phase procedure: (1) watermark embedding, which embeds watermark of each block to another mapping block; (2) tampered block identification, which can be achieved through the two bits authentication data and 8-neighborhood tampered block checking; (3) tampered block recovery, which can be achieved through a 10 bits recovery data. Some digital images have more than one color space (eg. RGB images). For such images, all the color channels (i.e. R, G and B) are watermarked by the proposed algorithm. The detailed procedures are discussed in the following subsections.

3.3.1 Watermark Embedding Procedure

Watermark embedding procedure can be divided into following three phases: recovery bits generation, authentication bits generation and block mapping as shown in Fig. 3.1. Authentication bits are those which are used for tamper detection and localization, which in turn is used to verify the integrity and authenticity of the host image. Recovery bits are those bits which are used to recover the extensive content (5 MSBs) of the host image. For each block having size 2×2 of the host image, a twelve bits binary watermark will be generated in which ten bits will be dedicated for recovery bits and remaining two bits will be dedicated for authentication bits. Let X and Y denote the height and width of a host image F and N represents the total number of pixels ($N = X \times Y$)). The pixel values of the host image are denoted by $f_n \in [0, 255]$, where n=1, 2, 3, $\cdots$, N and 8 bits of each f_n can be represented by $f_{n,i}$ where

$$f_{n,i} = \lfloor \frac{f_n}{2^{i-1}} \rfloor \bmod 2, i = 1, 2, ..., 8 \tag{3.10}$$

3.3.1.1 Recovery bits Generation Procedure

The recovery bits generation procedure can be done by the following steps:
Step 1. Remove three LSBs from each pixel of the host image F and divide into

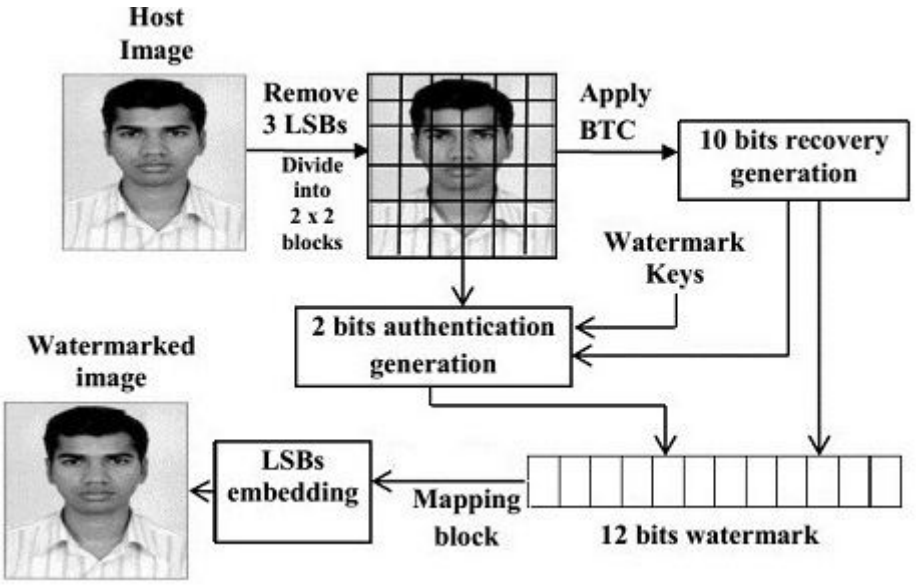


FIGURE 3.1: Block diagram for watermark embedding procedure.

size of 2×2 non overlapping blocks, so that total number of blocks are $\frac{N}{4}$. The i^{th} block pixels are denoted by $f_n^{i,p}$ where i is from 1 to $\frac{N}{4}$ and p is from 1 to 4.

Step 2. Now intensity range of all pixels have reduced from [0, 255] to [0, 31]. These thirty two values are divided into sixteen quantization levels and each pixel is assigned a grade $g \in [0, 15]$, according to the quantization level. Grade g of pixel f_n is assigned by Eq. (3.11).

$$g_n = \lfloor \frac{f_n}{2} \rfloor \tag{3.11}$$

Step 3. Apply BTC on each block after step 2 and generate the triplet (b,L,H) using Eq.(3.4), Eq.(3.7) and Eq.(3.8).

$$(b, L, H)^i = BTC(g_n^{i,p}), i = 1, 2, 3, ..., N/4; p = 1, 2, 3, 4. \tag{3.12}$$

where b is a binary vector of size 4, while$L$ and H are integer variables.

Step 4. Compute A and d, using quantization levels L and H as follow.

$$A = \frac{H + L}{2} \tag{3.13}$$

$$d = \frac{H - L}{2} \tag{3.14}$$

Step 5. Convert A and d into 4 bits binary and 2 bits binary respectively as defined in Eq.(3.15) and Eq. (3.16).

$$A_b = dec2bin(A, 4) \tag{3.15}$$

$$d_b = dec2bin(d, 2) \tag{3.16}$$

Step 6. Now generate a 12 bits empty binary vector V and the initial ten positions of vector V are filled by the binary vectors b, A_b and d_b with a systematic manner. These 10 bits binary values are called as recovery bits

3.3.1.2 Authentication bits Generation Procedure

Five MSBs planes, coordinate position of pixels and recovery bits are used to generated two authentication bits of each block in the following way:

Authentication bit A_{b1} Generation

In order to generate first authentication bit A_{b1} , consider f_n^r and f_n^c are eight bits binary value of corresponding row and column coordinate value of a pixel f_n of i^{th} block. Block diagram for A_{b1} shown in figure 3.2.

Step 1. Enter a secret random integer called as Key_1 for all blocks and convert into five bits binary using Eq. (3.17) and Eq. (3.18).

$$Key_1 = mod(Key_1, 32) \tag{3.17}$$

$$K_1 = Dec2bin(Key_1, 5) \tag{3.18}$$

Step 2. Calculate the hamming distance between MSBs of the pixel of a block with K_1 and convert it into three bits binary.

$$d = HammingDistance(f_{n,i}, K_1), i = 4, ..., 8. \tag{3.19}$$

$$d_b = Dec2bin(d, 3) \tag{3.20}$$

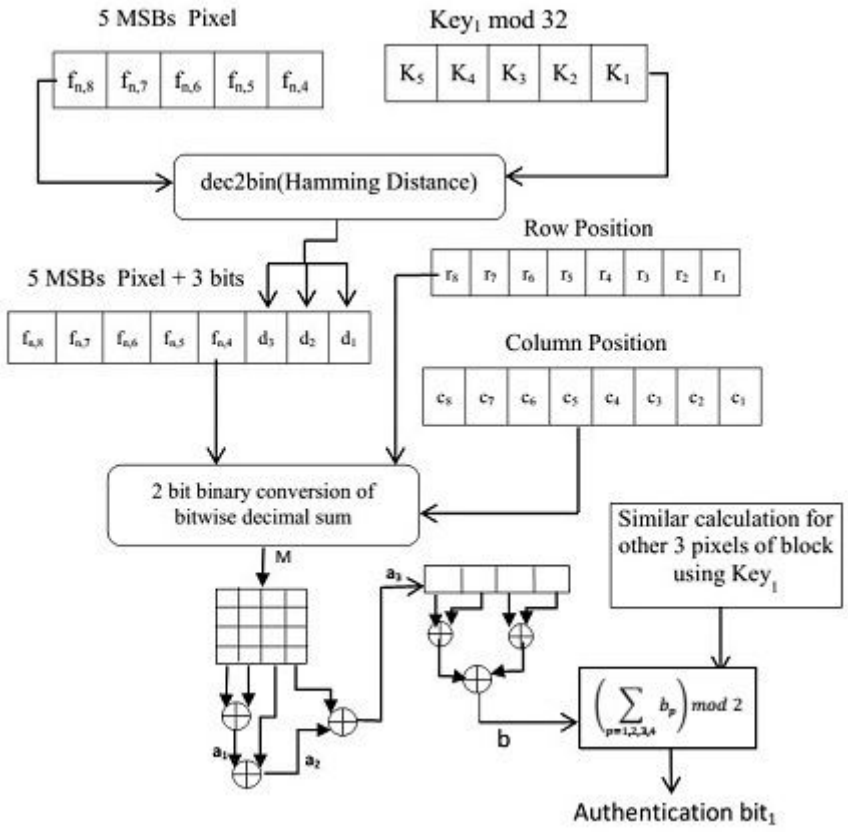


FIGURE 3.2: Block Diagram for Authentication bit A_{b1} Generation.

Step 3. Append three bits d_b in three LSBs of the pixel of a block as shown in Fig. 3.2.

$$h = append(f_n, d_b) \tag{3.21}$$

Step 4. Take the bitwise decimal sum of h , f_n^r and f_n^c corresponding to each pixel of a block in a vector C of size 1×8. So the value range of vector C varies from 0 to 3.

$$C = bitwise(h + f_n^r + f_n^c) \tag{3.22}$$

Step 5. Now fill a 4×4 matrix M by converting each index value of vector C in two bit binary representation.

$$M(1:16) = Dec2bin(C(i), 2); i = 1, 2, ..., 8 \tag{3.23}$$

Step 6. Take column wise X-OR operation of matrix M as follows.

$$m = M_1 \oplus M_2 \oplus M_3 \oplus M_4 \tag{3.24}$$

where M_1 , M_2, M_3, M_4 are column vectors of matrix M and m is a 1×4 column vector.

Step 7. Now take the bitwise X-OR within vector m and store this one bit in a binary variable b.

$$b = \sum_{i=1,2,3,4} (m(i)) \bmod 2 \tag{3.25}$$

Step 8. Repeat steps 2 to 7 for other three pixels of a block and calculate the first authentication bit A_{b1} of a block in the following way:

$$A_{b1} = \sum_{i=1,2,3,4} (b_i) \bmod 2 \tag{3.26}$$

where A_{b1} is the notation of first authentication bit.

After calculating authentication bit A_{b1} for each block, put them in the eleventh index of vector V.

Authentication bit A_{b2} Generation

The second authentication bit A_{b2} can be calculated as below:

$$v = V(1)$$

$$v = V(i) \oplus V(i), i = 2, 3, ..., 11 \tag{3.27}$$

$$A_{b2} = \begin{cases} 1 & \text{if } t = 0 \\ 0 & otherwise \end{cases} \tag{3.28}$$

After calculating second authentication bit A_{b2} for each block, put them in the last index of vector V. In this way twelve bits watermark is generated including ten recovery bits and two authentication bits for each block.

3.3.1.3 Block Mapping

The watermark generated from a block B_i should be embedded into another block B_j instead of the same block $\{i \neq j, \forall i, j | i, j \in [1, N/4]\}$. Then the block mapping $\{(B_i, B_j), i \neq j\}$ is required for watermark embedding. For this, the generation algorithm of the block mapping sequence is as follows.

Step 1. Allot a consecutive unique number $i \in \{1, 2, ..., N/4\}$ to each image block in the raster scan order, where $N/4$ is the total number of blocks.

Step 2. Enter a key $Key_2 \in [1, N/4 - 1]$, which is a prime number.

Step 3. For each block number i, apply a 1-D transformation Eq.(3.29) to acquire j, the number of its mapped block.

$$f(i) = (Key_2 \times i) mod N_1 \tag{3.29}$$

$$j = f(i) + 1 \tag{3.30}$$

where$i, j (\in [1, N/4])$ are the block number and $N_1 = N/4$. Here $Key_2 \in [1, N/4 - 1]$ must be a prime number in order to acquire a one-to-one mapping; otherwise, the period is less than $N/4$ and a many-to-one mapping may occur.

Step 4. The vector V is permuted by using a seed based generated key, Key_3 and insert this permuted vector of block B_j into three LSBs planes of corresponding mapped block B_j .

In this way, the watermarked image is generated in which five MSBs planes of the host image are preserved and three LSBs planes are replaced with the watermark information.

3.3.2 Content Restoration Procedure

Content restoration procedure can be divided into following two phases: tampered block identification and tempered block restoration as shown in Fig. 3.3. Suppose an attacker alters original contents of the watermarked image without changing the image size.

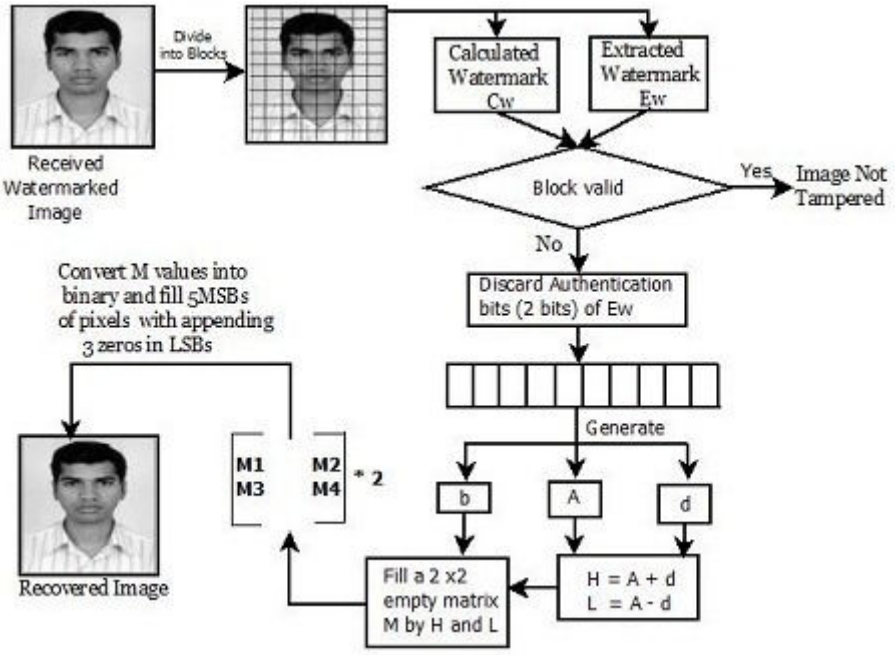


FIGURE 3.3: Block diagram for watermark extraction and image restoration procedure.

3.3.2.1 Tampered Block Identification Procedure

The test image is first divided into non-overlapping blocks of 2×2 pixels, as in the watermark embedding process. The tampered block identification procedure is described in the following steps.

Step 1. First of all, receiver extracts twelve bits watermark from each corresponding mapped block using Key_2 which was used at the time of block mapping, in a twelve bit vector V for each block and reshuffle it using Key_3.

Step 2. Calculate the recovery bits and authentication bits with the help of Key_1 for each block as discussed in subsection 3.3.1.

Step 3. Now compare the calculated authentication bits with extracted watermark authentication bits, if mismatch is found, mark that block as invalid otherwise consider it as valid block.

Step 4. Assign the white color as seen in the image (i.e. 255) to the invalid blocks.

3.3.2.2 Tampered Block Restoration Procedure

After the tamper block identification procedure, all blocks are marked either invalid or valid. Once all invalid blocks are identified, they are needed to be restored. An invalid block B can be restored from extracted watermark in the following.

Case 1: When mapped block marked as valid block

If the mapped block is marked as valid, then restoration of block B can be performed in following steps:
Step 1. Generate triplet (b, H, L) from the extracted watermark vector V as:

$$b(1:4) = V(1:4) \tag{3.31}$$

$$A = bin2Decimal(V(5:8))$$

$$d = bin2Decimal(V(9:10))$$

$$H = A + d \tag{3.32}$$

$$L = A - d \tag{3.33}$$

Step 2. Initialize a 2×2 empty matrix R and fill this matrix by H and L as follow.

$$R(x,y) = \begin{cases} H & \text{if } b(2*x+y) = 1 \\ L & \text{if } b(2*x+y) = 0 \end{cases} \tag{3.34}$$

Step 3. Multiply the matrix R with quantization value 2. Now matrix values become in the range $\in [0, 31]$.

$$R(x,y) = 2 * R(x,y) \tag{3.35}$$

Step 4. Now convert the $R(x,y)$ value into five bits and fill the five MSBs of corresponding pixel in invalid block of tampered image and append three zeros in first three LSBs of each pixel to make gray value range from 0 to 255.
Step 5. Finally after the completion of restoration of the block B, mark it as a valid block.

Case 2: When mapped block marked as invalid block

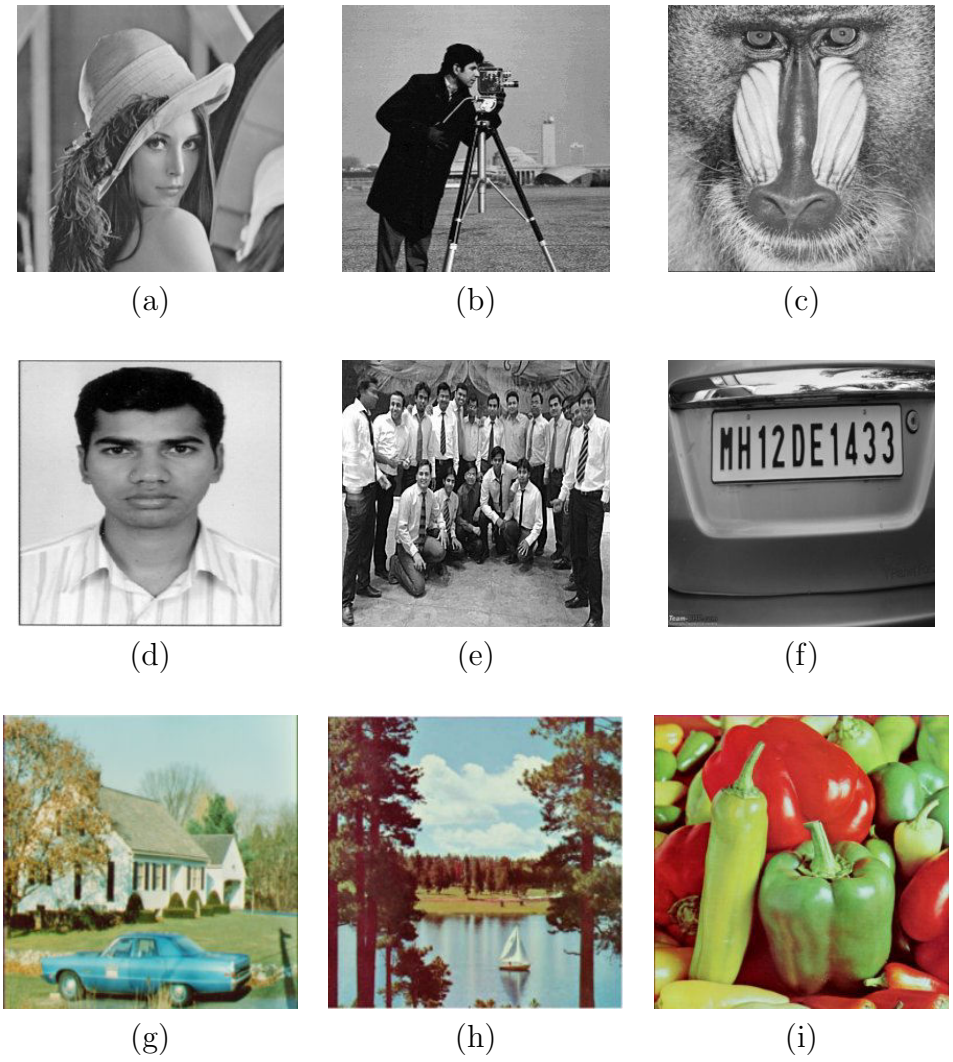
If the mapped block is marked as invalid, then restoration of block B can be performed in following steps:
Step 1. Extract all valid blocks from 8-neighborhood of block B and calculate mean values of those blocks.
Step 2. Take the average value of calculated means in step 1 and substitute for every pixels of the block B.
Step 3. After the completion of restoration of the block B, mark it as a valid block.

Finally in this way, restored image is generated by restoring the tampered block by pixel by pixel manner. The restored image is very similar to the original watermarked image with approximate intensities that is demonstrated by the experimental results.

3.4 Experimental Results and Discussions

The proposed method described in this paper is implemented in MATLAB environment. The computational platform was a Core i7-3770 processor with a speed of 3.40 GHz and 2 GB of RAM. To evaluate the performance of the proposed methodology, a set of test images of size 256×256 are chosen. Fig. 3.4 shows some of the test images which were used in the experiments. Their corresponding watermarked images are shown in Fig. 3.5. Table 3.1 shows the PSNR and NCC values of watermarked image relative to the original images. As the embedding PSNR and NCC values are very high, so it is very difficult to differentiate between the watermarked and the original image in vision.

Fig. 3.6 shows the tampered detection and restoration of Group Photo. Fig. 3.6(a) gives the watermarked of Group Photo. The PSNR and NCC value due to watermark embedding are 40.15 dB and 0.9989 respectively. Such high PSNR and NCC values confirm that the distortion due to watermark is imperceptible. We modified the watermarked image by shuffling the head portion(i.e. most sensitive content) between the people as shown in Fig. 3.6(b) and corresponding modified blocks were detected by using the authentication bits as shown in Fig. 3.6(c). In Fig. 3.6(c) white regions indicate tampered blocks. The final recovered image is shown in Fig.



(a) (b) (c)

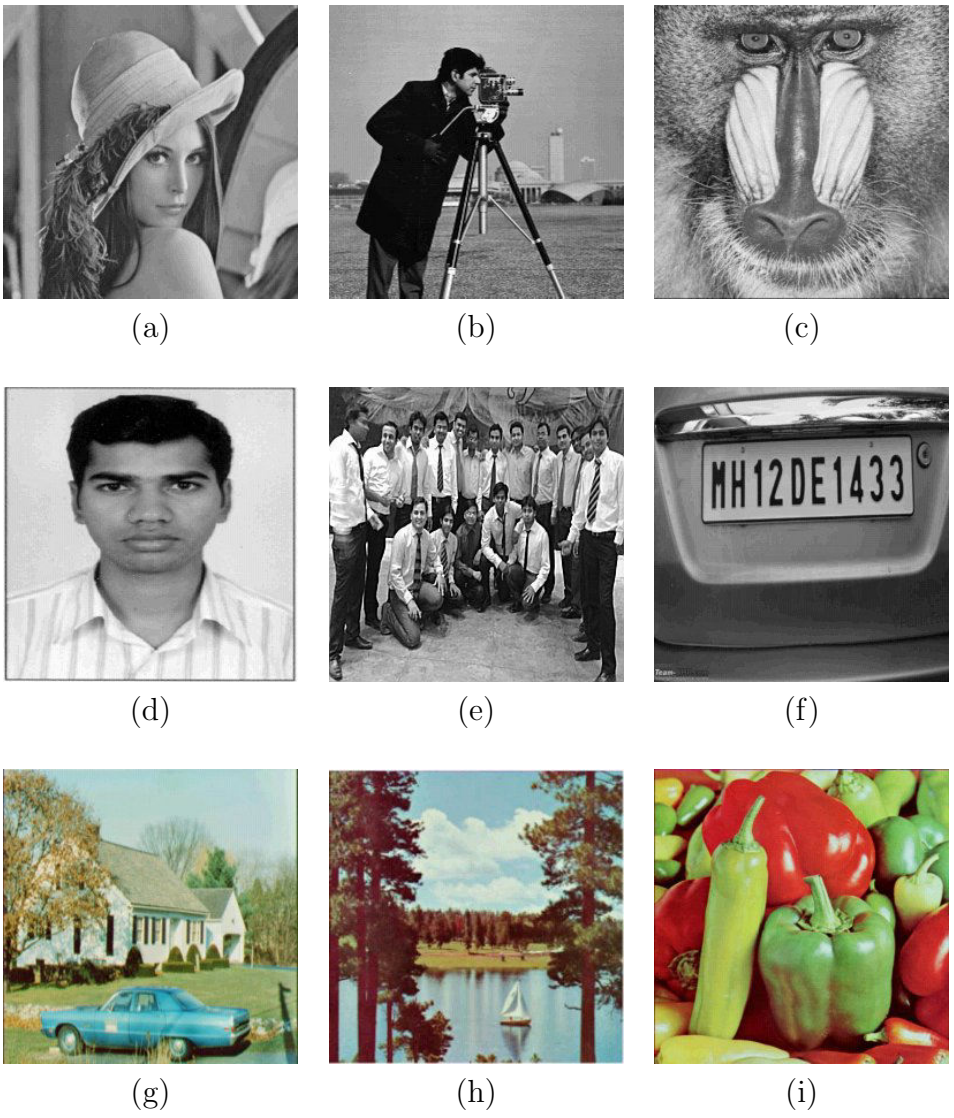
(d) (e) (f)

(g) (h) (i)

FIGURE 3.4: Test images used in our experiments (a) Lena (b) Cameraman (c) Baboon (d) Own Photo (e) Group Photo (f) Number Plate (g) House (h) Boat (i) Pepper.

3.6(d). The PSNR and NCC values of restored image are 41.35 dB and 0.9812 respectively with the reference of watermarked image. So, fidelity of recovered image in Fig. 3.6 is very satisfactory.

Similarly Fig. 3.7(a) is a tampered image whose 50 % area is tampered and corresponding modified blocks were detected by using the authentication bits as shown in Fig. 3.7(b). Fig. 3.7(c) is the recovered image, whose PSNR and NCC are 28.44 dB and 0.9792 respectively with respect to the watermarked images. As the PSNR and NCC values of recovered image are very high, therefore the visual qualities of the recovered images are very satisfactory.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 3.5: Watermarked images (a) Lena (b) Cameraman (c) Baboon (d) Own Photo (e) Group Photo (f) Number Plate (g) House (h) Boat (i) Pepper.

Fig. 3.8 shows the tampered detection and restoration of the color image Boat. The watermark is embedded in the R,G,B channels. The watermarked image is shown in Fig. 3.8(a). The PSNR and NCC values due to watermark embedding are 38.7821 dB and 0.9984 respectively. These high PSNR and NCC values indicate that the distortion due to watermark is imperceptible. The tampered image is shown in Fig. 3.8(b), in which the boat part in the watermarked image is inserted into three arbitrary positions. The tamper detection result is shown in Fig. 3.8(c), in which non black regions are indicating tampered blocks. The restored image is shown in Fig. 3.8(d). The PSNR and NCC values of this restored image are 37.9913 dB

TABLE 3.1: Essential information observed during watermark embedding.

Cover Image	PSNR (Embedding)	NCC (Embedding)
Lena	39.86 dB	0.9977
Cameraman	39.00 dB	0.9986
Baboon	40.96 dB	0.9976
Own Photo	39.95 dB	0.9989
Group Photo	40.15 dB	0.9989
Number Plate	39.08 dB	0.9981
House	38.93 dB	0.9979
Boat	38.78 dB	0.9984
Pepper	38.67 dB	0.9978

(a)Watermarked Image

(b) Tampered Image

(c) Tampered detected Area

(d) Restored Image

FIGURE 3.6: Tampered detection and restoration of Group Photo.

(a) (b) (c)

FIGURE 3.7: Tampered detection and recovery of Group Photo at tampering rate 50 %.

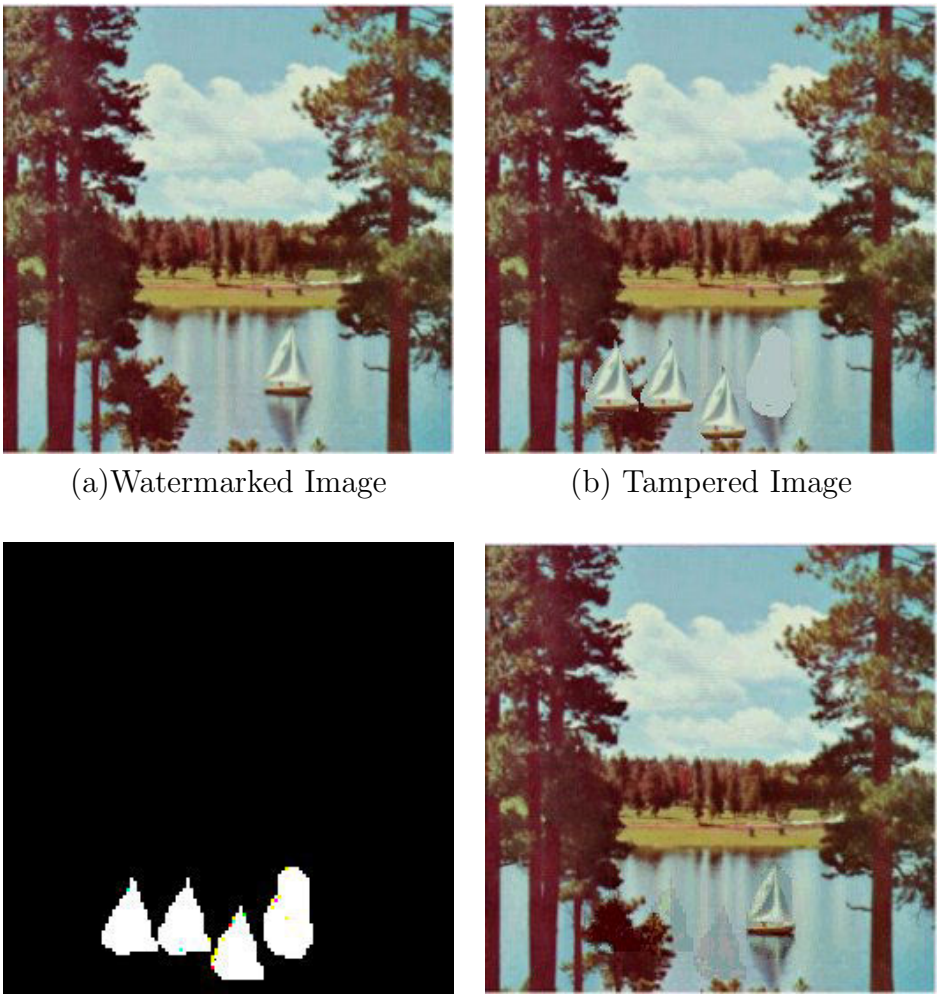
TABLE 3.2: PSNR(dB) of restored content in the tampered area with different tampering rates.

Cover Image	Tampering rate(%)					
	5	10	20	30	40	50
Lena	39.14	36.04	32.67	30.88	29.76	28.77
Group Photo	37.78	35.2	32.35	30.53	29.39	28.45
Camera-man	37.91	35.35 3	2.43	30.79	29.84	29.01
Own Photo	37.86	36.02	33.78	31.78	30.32	29.15
Baboon	38.07	35.35	32.31	30.55	29.39	28.42

and 0.9965 respectively with the reference of watermarked image. The experiment reveals that the ability of the proposed method to detect and localization tampering with restoration quality is adequate.

Table 3.2 shows PSNR value of the restored image in tampered area with respect to the different tampering rates and Fig. 3.9 shows the corresponding graph of five gray images Lena, Cameraman, Group Photo, Baboon, Own Photo. it can be seen from Fig. 3.9 that the PSNR value with respect to tampering rate decreases smoothly. Moreover, it has been observed that even if the tampering rate is up to 50%, the restored contents have PSNR values more than 28.42 dB which are quite satisfactory.

Table 3.3 compares several fragile watermarking schemes with restoration capability.



(a)Watermarked Image (b) Tampered Image

(c) Tampered detected Area (d) Restored Image

FIGURE 3.8: Tampered detection and restoration of Color Boat.

The schemes in [64, 117] and [66]-A do not work with a large tampering rate. Embedding PSNR of the proposed scheme is effectively higher than the method used in [64, 65, 66] and [67]. By using the proposed scheme, the original content in an extensive area (50 %) can be recovered better than [65], [66]-B and [67]. The proposed scheme also provides high accuracy in tampered pixel localization and is also effective in removing blocking artifacts due to use of small size blocks. In [64, 65, 66, 67], the accuracy of tamper localization decreased since these schemes are using large blocks of size 8×8.

Fig. 3.10 gives the PSNR of the recovered images using our scheme and other schemes in [65, 66, 67]. The plot demonstrates the characteristic behavior of the

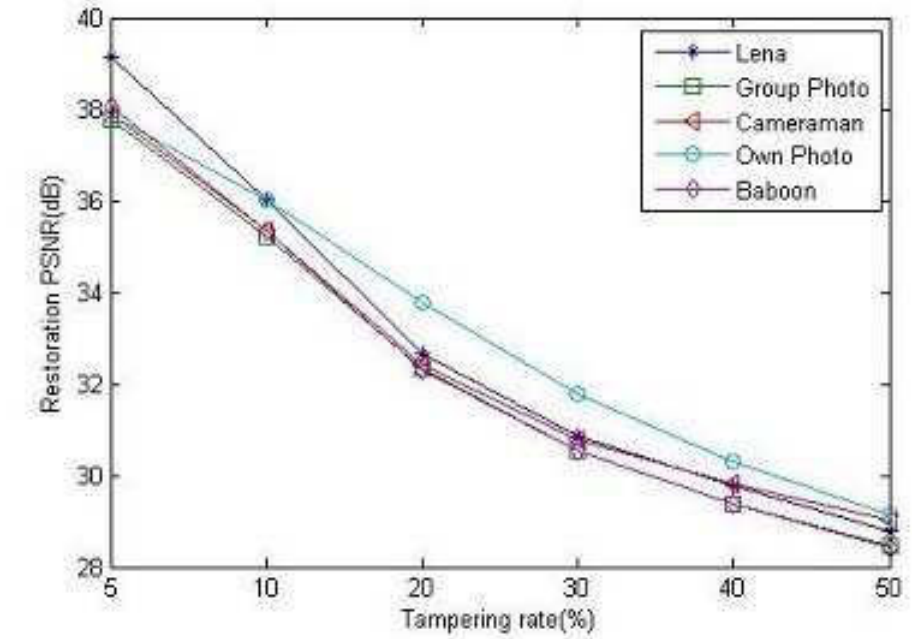


FIGURE 3.9: PSNR of restored content with respect to the tampering rates.

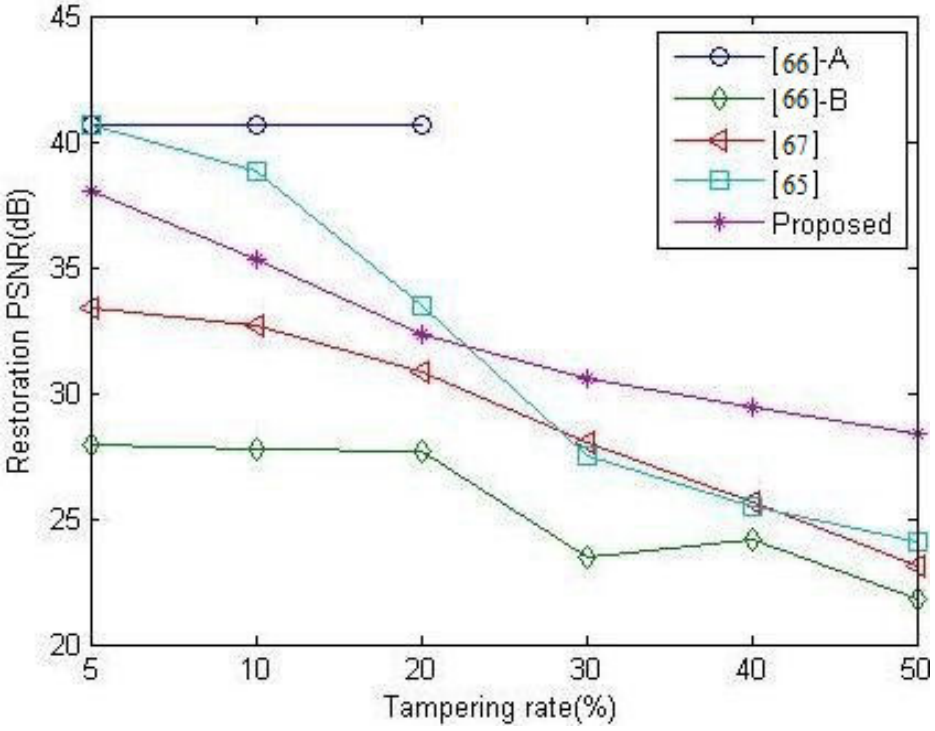


FIGURE 3.10: Reconstruction PSNR for Lena image under varying tampering rates.

TABLE 3.3: Comparison of restoration capability among several fragile watermarking schemes.

Scheme	Embedding PSNR(dB)	Restoration PSNR(dB)	Condition of Restoration
[64]	37.90	35.0	Tampering rate $< 35\%$
[65]	37.90	[24,41]	Tampering rate $< 60\%$
[117]	37.90	37.90	Tampering rate $< 6.6\%$
[66]-A	37.90	40.7	Tampering rate < 24 %
[66]- B	37.90	[22, 40]	Tampering rate $< 66\%$
[67]	37.90	[22, 38]	Tampering rate $< 54\%$
Proposed	39.0	[28.42, 40]	Tampering rate $\leq 50\%$

systems. The scheme [65] is more susceptible to the distributions of details and curve may not be monotonic. The scheme [67] operates directly on pixel intensities, and is not affected by the problem. The plots also demonstrate the different quality levels for the [66]-B scheme. The scheme in [66]-A does not work with a large tampering rate. By using the proposed scheme, the content in an extensive area (50 %) can be recovered better than [65], [66]-B and [67]. Thus proposed scheme is more flexible than previously existing schemes.

3.5 Conclusion

This chapter presents a quantization and BTC based efficient and effective self-embedding fragile watermarking scheme for tampered detection and restoration. This scheme is efficient in time complexity due to use of BTC and simple XOR operations only. The quality of the watermarked images is high, with the average of 39.0 dB PSNR. In the process of watermark bits generation and embedding, small non overlapping blocks sized 2×2 are used to improve the accuracy of localization. Experimental results demonstrate that the accuracy of tampered detection and localization is effectively high and the recovery quality scores of the proposed scheme is better than the previously reported schemes. Even under extensive tampering, high quality restoration is possible by this proposed scheme.