Chapter 5

DCT based Effective Fragile Watermarking Scheme for Image Authentication and Restoration

5.1 Introduction

In recent years with the rapid development of Internet and computer technology, multimedia information in digital format such as document, graphics, drawings, still and moving images (video), audio can be easily spread, copied and edited. Therefore, the authentication and restoration of the original content in the tampered regions arise as one of the recent imperative issue. Many techniques are available to protect the integrity and ownership for digital images. Digital watermarking is one such technique that embeds imperceptibly a meaningful signature or some secret information called as the watermark, into the cover image. This imperceptible and secret watermark can be later detected/extracted for a wide variety of purposes including authentication, content identification, copyright protection and ownership verification [3].

The performance of the watermarking schemes can be measured by essential properties like fidelity, embedding effectiveness, robustness, data payload and blind or informed detection etc. The relative significance of each property is dependent on

the requirements and nature of the application [4, 139]. The fidelity of a watermarking scheme refers to the perceptual similarity between the original and watermark versions of the cover image. The embedding effectiveness means the probability that the embedder will successfully insert a watermark in arbitrarily selected cover work. In other word, the embedding effectiveness is probability of recognition imminently after embedding. Robustness of the watermark system refers to the ability to resist different image processing attacks. Data payload refers to the number of bits a watermark encodes within a unit of time or within a cover image. For an image, the data payload would refer to the number of bits encoded within the image. In many application detection must be performed without access to the original work, so the detectors that do not require any information related to the original are referred to as blind detector (public watermarking system) whereas opposite is called informed detector (private watermarking system). Hence, the watermarking technique is designed to be robust, fragile or semi fragile depending on the application area like legacy enhancement, copy control,transaction tracking, device control, content authentication etc. The watermark techniques can also be classified into two major types, spatial and transform domain techniques on the basis of domain specific data embedding [7, 140]. In the spatial domain techniques, watermark directly applies on pixel value of the image [9, 10]. Transform domain techniques perform the watermarking by changing the coefficients in the transformed domain of the cover image. Primary examples for this method include watermarking obtained by modifying the Discrete Wavelet Transform (DWT) coefficients [12, 141, 142], Discrete Fourier Transform (DFT) coefficients [143] and Discrete Cosine Transforms (DCT) coefficients [65, 144].

A robust watermark should be able to resist intentional or unintentional manipulations and is used for ownership verification and copyright protection[6, 7, 8]. A fragile watermark is intended to be destroyed even after the minor unintentional or intentional manipulation [7, 9, 10]. The third category, semi-fragile watermarking uses watermarks that have the ability to resist unintentional manipulations caused by common image processing operations like JPEG compression and are fragile against intentional, malicious manipulations [12, 142]. The main application field of fragile and semi-fragile watermarking is image and video content authentication and tamper detection [7]. Here we are focusing on fragile watermarking scheme because it is very sensitive to any minor modification in watermarked image.

Fragile watermarking can be divided into two major classes, block-wise fragile watermarking [145, 146] and pixel wise fragile watermarking [10, 52]. The main concept of the pixel-wise fragile watermarking technique is obtaining watermark information by gray scale value of each pixel of the cover image and then embedding them into the LSBs of that pixel itself or corresponding mapped pixel. This procedure is called self-embedding. If the gray scale value of any pixel is changed, the embedded watermark corresponding to that pixel will also change and hence one can easily localize each altered pixel. In block-wise fragile watermarking, the cover image is divided into small blocks and each block has watermark information. This watermark may be any function based on principal content of the cover image. If image is altered intentionally or unintentionally, the tampered block and watermark contained in that block will mismatch. The fragile watermark scheme can identify these tampered blocks.

The remaining part of this paper is organized as follows. Section 5.2 summarizes the related research work with quality-related trade-offs. The detailed methodology of proposed watermark embedding procedure and extraction procedure is presented in Section 5.3. The experimental evaluation scenario and details of comparison with the existing approach are described in Section 5.4. Finally, the conclusion of this paper is drawn in Section 5.5.

5.2 Related Prior Research

Due to the trade-off between the image quality and the restoration conditions, there is no general model of the content reconstruction problem despite the existence of the variety of restoration schemes. The image quality can be decided by PSNR i.e.measured distortion of the original image. The restoration conditions are usually the maximum tampering rate for which the restoration is still possible.

Korus and Dziech proposed a random linear fountain (RLF) based watermarking scheme of the content restoration problem in self-embedding systems in [63]. The scheme formulates the reconstruction as a communication over an erasure channel, and gives a closed-form expression for the achievable success bounds for traditional

self- restoration with uniform image quality. This scheme is capable to recover even when 50 % of the image area becomes tampered.

Zhang et al. proposed a content reconstruction problem in terms of compressive sensing in [65]. In this scheme, the embedded watermark data for content recovery are computed from the DCT coefficients of the host image. When a part of a watermarked image is tampered, the watermark data in the area without any modification can be extracted. The recovered image quality by this scheme depends on the tempered area. If the smaller the tampered area, the more the amount of available watermark data will be, leading to a better quality of recovered content. This scheme is capable to restore up to tampering rate 60 %.

Two self-embedding watermarking techniques were proposed by Zhang et al. [66], called as a reference sharing mechanism. In the first technique, the watermark was derived from 5 MSBs of the original cover image. In the second technique, the cover image was decomposed into three levels based on the hierarchical self-embedding scheme. 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. So that a better restored image can be obtained in second scheme, from a tampered version with less fake content.

In [67], Zhang et al. first permuted the image pixels based on a secret key and then divided them into a series of pixel-pairs. The restoration data was generated by exclusiveOR operation within the original MSBs of pixel pairs and authentication data derived from MSBs and restoration data. Finally, watermark data was embedded into the three LSBs planes. In this scheme, the five MSBs of a pixel pair can be either fully or partially recovered by using reference data . The remaining uncertainty is resolved by manipulating local pixel correlations.This scheme is capable to restore up to tampering rate 54 %.

In [64], Qian et al. presented a scheme based on Discrete Cosine Transform to reduce the embedding data for self-restoration. The DCT coefficients of 8×8 blocks were encoded into different numbers of bits and the authentication-bits and restoration-bits were embedded into the three LSBs planes of the cover image. However the accuracy of tamper localization decreases because of large block size.

A DCT and fractal compression coding based self-embedding fragile watermarking scheme was proposed by Zhang et al.[71]. In this scheme, three kinds of watermarks were generated for image authentication and restoration, which was in turn based on an interleaved and overlapped 8×8 image block. Three versions of restoration watermarks for each block were embedded into different quadrants, which provide three chances for block restoration in case of any image modification. However in this scheme the accuracy of tamper localization also decrease because of large block size of 8×8.

The proposed method in this chapter is based on a self-embedding blind fragile watermarking scheme. Here, a block-wise mechanism is used for tampered area detection and restoration. We also present a comprehensive performance evaluation on well-known images like Lena, Baboon, Cameraman etc. 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 removal of blocking artifact problem. The presented scheme is also secured using six predefined keys. Experimental results show that for tampering rate up to 50 %, it allows for reconstruction with high PSNR and NCC values which is effective enough for fidelity.

5.3 Proposed Methodology

In this proposed scheme, the five most significant bits (MSBs) of each pixel in the cover image are kept unchanged, while the three least significant bits (LSBs) of each pixel are replaced with watermark. The watermark is determined by the five MSBs planes of the cover image and divided into two part. They are used to locate and detect the tampered blocks and to restore the original content respectively. 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. Let the original image F contain X and Y numbers of rows and columns respectively, and let M represent the total number of pixels ($M = X \times Y$). The intensity range of each pixel of the image is denoted by $f_m \in [0, 255]$, where m=1, 2, 3, $\cdots$, M.

Each f_m can be represented by 8 bits, $f_{m,8}$, $f_{m,7}$,$\cdots$, $f_{m,1}$, where

$$f_{m,i} = \lfloor \frac{f_m}{2^{i-1}} \rfloor \bmod 2, i = 1, 2, ..., 8 \tag{5.1}$$

5.3.1 Watermark Embedding Procedure

Watermark embedding procedure can be divided into three phases. The first one is the multitasking bits generation, second one is the authentication bits generation and last one is block mapping as shown in Fig. 5.1. Authentication bits are those which are used for tamper detection and localization to verify the integrity of the cover image whereas multitasking bits are those bits which are used as authentication bits as well as to recover the extensive content of the cover image. For each block having size 2×2 of the cover image, a vector V of size twelve bits will be generated in which first ten bits will be dedicated for multitasking bits and remaining two bits will be dedicated for authentication bits.

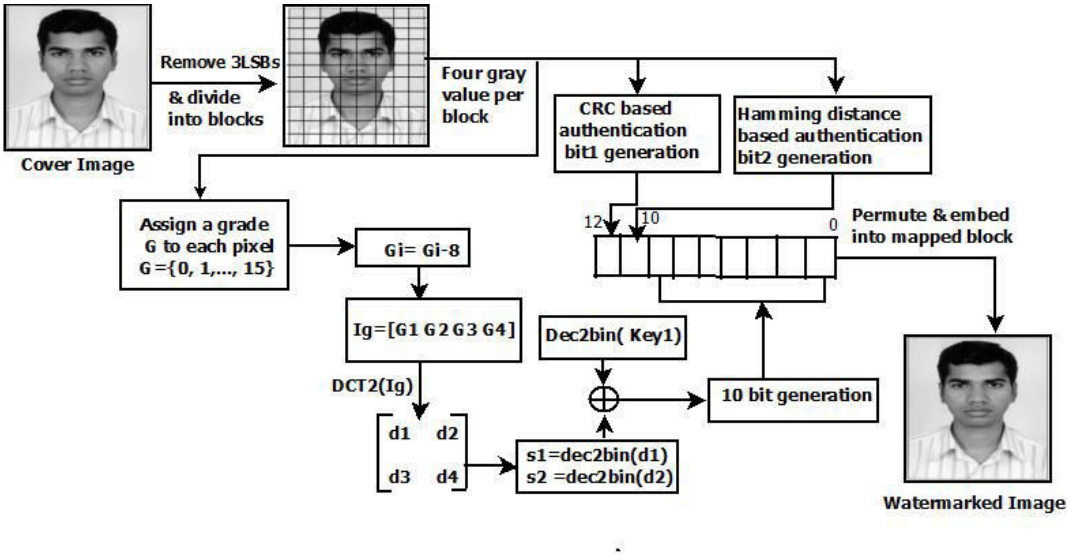


FIGURE 5.1: Block diagram for watermark embedding procedure.

5.3.1.1 Multitasking bits Generation Procedure

The multitasking bits generation procedure can be done by the following steps:

Step 1. Remove three LSBs from each pixel and divide the cover image F into non

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

Step 2. Now each pixel has range from 0 to 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_m is assigned by equation 5.2.

$$G = \lfloor \frac{f_m}{2} \rfloor \tag{5.2}$$

Step 3. Now each pixel has a grade $G \in [0, 15]$. Here sixteen possible levels, so shift the level of grade value $G_m^{i,b}$ of pixel $f_m^{i,b}$ by subtracting each pixel grade by half of the maximum possible values, i.e. eight.

$$G_m^{i,b} = G_m^{i,b} - 8, i = 1, 2, 3, ..., M/4; b = 1, 2, 3, 4. \tag{5.3}$$

Step 4. Perform the DCT on each block after step 3.

$$D_i = DCT2(G_m^{i,b}) = \begin{bmatrix} d_{i1} & d_{i2} \\ d_{i3} & d_{i4} \end{bmatrix}, i = 1, 2, 3, ..., M/4; b = 1, 2, 3, 4. \tag{5.4}$$

where D_i is the DCT coefficients of i^{th} block.

Step 5. By thorough analysis of experimental results it has been observed that second row of DCT coefficients (d_{i3} and d_{i4}) value almost remains zero or negligible while the first row of DCT coefficients (d_{i1} and d_{i2}) range from -11 to 11. Using the first row of DCT coefficients (d_{i1} and d_{i2}) and generate the 10 bits and put in a vector V as follows:

$$s_j = round(d_{ij}), i = 1, 2, 3, ..., M/4; j = 1, 2 \tag{5.5}$$

$$sign(s_j) = \begin{cases} 0 & \text{if } s_j \geq 0; \\ 1 & \text{if } s_j < 0. \end{cases} \tag{5.6}$$

$$\begin{aligned} V(1) &= sign(s_1) \\ V(2:5) &= dec2Bin(s_1, 4) \\ V(6) &= sign(s_2) \\ V(7:10) &= dec2Bin(s_2, 4) \end{aligned} \tag{5.7}$$

Step 7. Enter a secret key Key_1 and convert this Key_1 into ten bits binary and store in a vector V_1 of size 10. Finally, calculate the ten bits multitasking bits using equation 5.8.

$$V(i) = V_1(i) \oplus V(i); i = 1, 2, 3, ..., 10. \tag{5.8}$$

Now the initial tenth position of vector V is filled by the ten multitasking bits with a systematic manner as shown in Fig. 5.2. It is noted that the non zero values always range from -11 to 11. This is the reason for assigning only five bits including sign bit for each value.

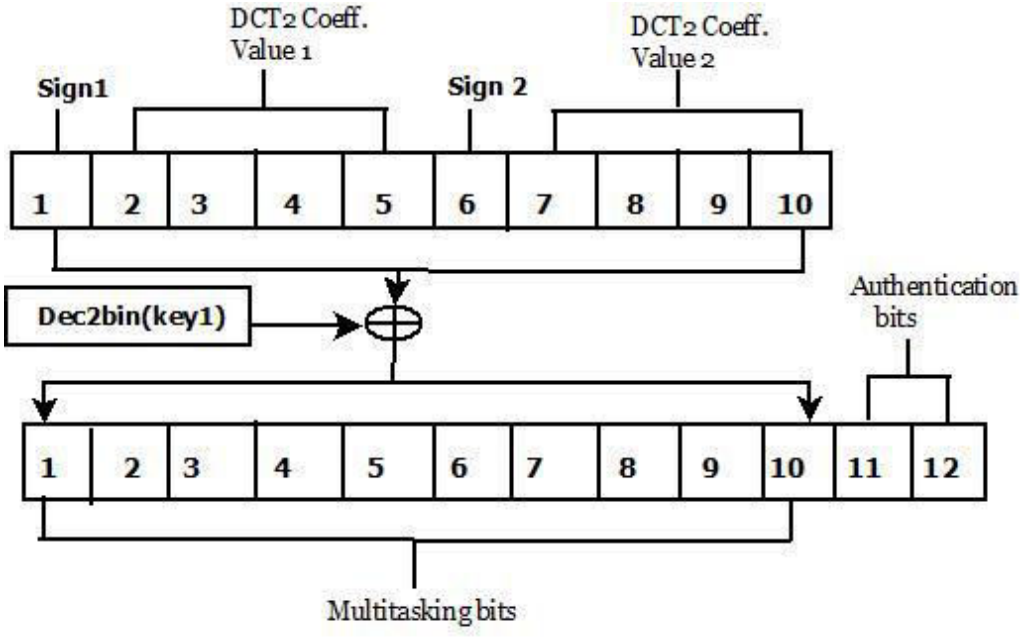


FIGURE 5.2: Block Diagram for twelve bit vector V with watermark value.

5.3.1.2 Authentication bits Generation Procedure

Using five MSBs planes and spatial location of pixels, two authentication bits for each block can be calculated in the following manner:

Authentication bit A_{b1} Generation

In order to generate first authentication bit, generate a pseudo random binary matrix R of size $\frac{X}{2} \times \frac{Y}{2}$ based on a secret key Key_4 which refers the seed value for R.

Consider a pixel f_m is in i^{th} block $f_m^{i,b}$ and all five bits of f_m are represented as $f_{m,p}$ where $p \in (4...8)$.

Step 1. Enter a secret key Key_2 for all blocks and convert into four bits binary using equations 5.9 and 5.10.

$$Key_2 = mod(Key_2, 16) \tag{5.9}$$

$$K_2 = Dec2bin(Key_2, 4) \tag{5.10}$$

Step 2. Append the three zeros in the last of each pixel of a block as shown in Fig. 5.3.

$$A^a = append(f_m^a, 000); a = 1, 2, 3, 4; \tag{5.11}$$

Step 3. Calculate three bits cyclic redundancy check (CRC) of A^a with K_2.

$$B_j^a = CRC(A^a, K_2); a = 1, 2, 3, 4; j = 1, 2, 3. \tag{5.12}$$

Step 4. Finally, calculate the first authentication bit A_{b1} from three CRC bits of each pixel of a block in the following way:

$$B_1^a = (B_1^a \oplus B_2^a); a = 1, 2, 3, 4 \tag{5.13}$$

$$b^a = (B_1^a \oplus B_3^a); a = 1, 2, 3, 4 \tag{5.14}$$

$$A_{b1} = \sum_{a=1,2..4} (b^a) \bmod 2 \tag{5.15}$$

$$A_{b1} = A_{b1} \oplus R^i; i = 1, 2, 3, ..., M/4. \tag{5.16}$$

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

Authentication bit A_{b2} Generation

Consider a pixel f_m is in i^{th} block $f_m^{i,b}$ and all five bits of f_m are represented as $f_{m,p}$ where $p \in (4...8)$. Similarly f_m^r and f_m^c are binary value of corresponding row and column value of f_m in spatial image plane. The second authentication bit A_{b2} can be calculated in the following steps:

Step 1. Calculate the correlation between row and column value of each pixel of a

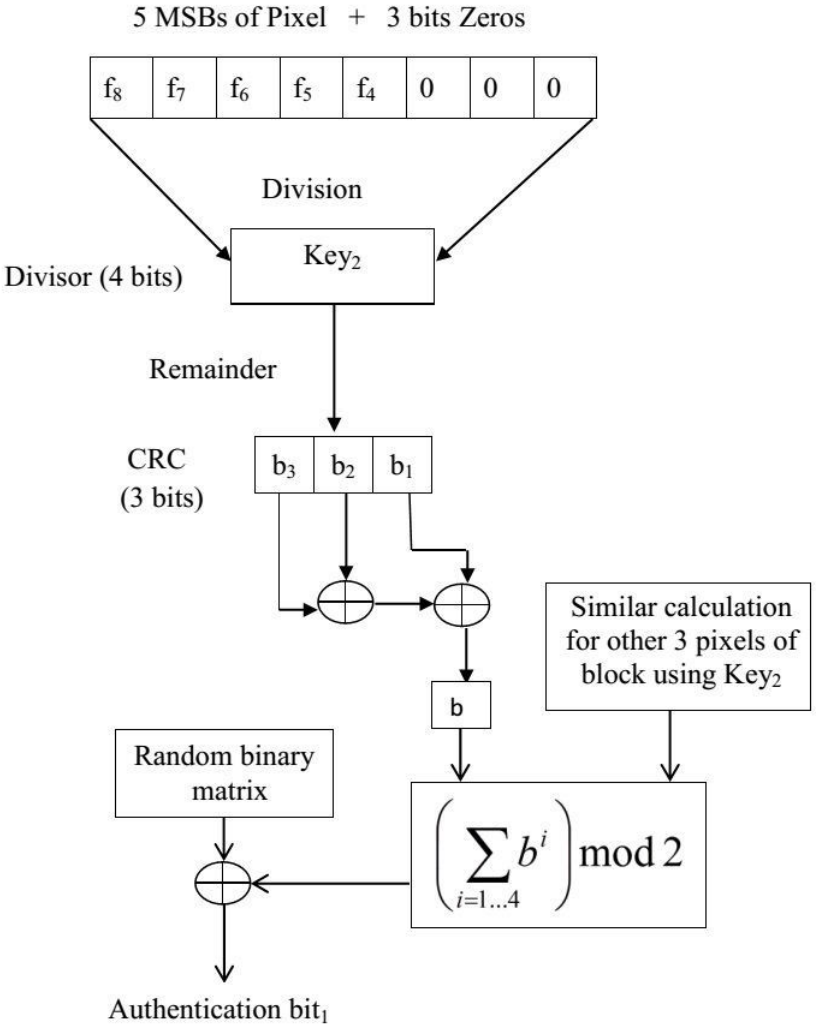


FIGURE 5.3: Block Diagram for Authentication bit A_{b1} Generation.

block in a vector of size 8 and permute them with secret Key using the equations 5.17 and 5.18.

$$b^a = (f_m^r \oplus f_m^c) \tag{5.17}$$

$$b^a = randPermute(b^a) \tag{5.18}$$

Step 2. Convert the value obtained in equation 5.18 into one bit and store into a binary variable, let say s as shown in Fig. 5.4.

Step 3. Calculate the hamming distance of each pixel with a secret key Key_3 using equation 5.21 and store it into one bit binary variable d using equation 5.22.

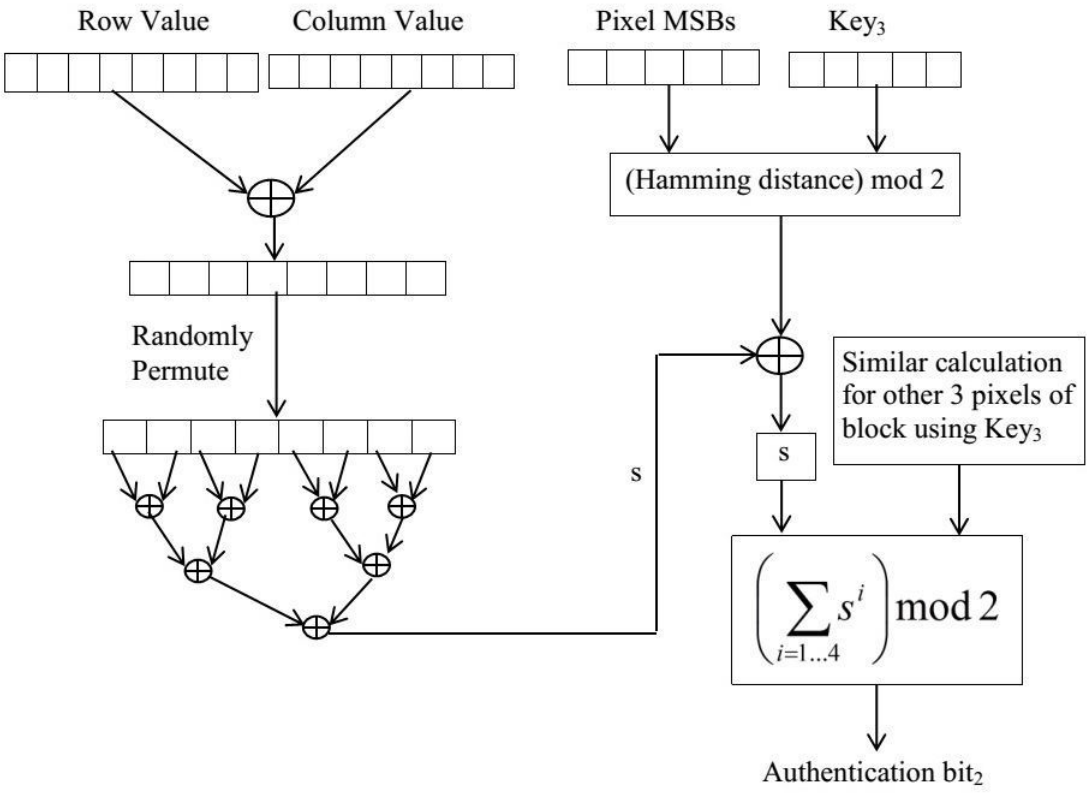


FIGURE 5.4: Block Diagram for Authentication bit A_{b2} Generation.

$$Key_3 = mod(Key_3, 32) \tag{5.19}$$

$$K_3 = Dec2bin(Key_3, 5) \tag{5.20}$$

$$d = HammingDistance(f_{m,p}, K_3); p = 4, 5, 6, 7, 8 \tag{5.21}$$

$$d = mod(d, 2) \tag{5.22}$$

Step 4. Finally, calculate the A_{b2} using equations 5.23 and 5.24.

$$s^a = (s^a \oplus d^a); a = 1, 2, 3, 4 \tag{5.23}$$

$$A_{b2} = \sum_{a=1,2,...,4} (s^a) \bmod 2 \tag{5.24}$$

Where A_{b2} is the notation of second authentication bit

After calculating these two authentication bits for each block, put them in the last two indexes of vector V. In this way twelve bits watermark generated including ten

multitasking bits and two authentication bits for each block as shown in Fig. 5.2.

5.3.1.3 Block Mapping

The watermark generated from a block B_i should be embedded into another block B_j instead of the same block $\{i \neq j, \forall i, j | i, j \in [1, M/4]\}$. 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 $i \in \{1, 2, ..., M/4\}$ to each image block in the raster scan order, where $M/4$ is the total number of blocks.

Step 2. Enter a secret key $Key_6 \in [1, M/4 - 1]$, which is a prime number.

Step 3. For each block number i, apply a 1-D transformation equation 5.25 to acquire j, the number of its mapping block.

$$f(i) = (Key_6 \times i) mod M_1 \tag{5.25}$$

$$j = f(i) + 1 \tag{5.26}$$

where$i, j (\in [1, M/4])$ are the block number and $M_1 = M/4$. Here $Key_6 \in [1, M/4-1]$ must be a prime number in order to acquire a one-to-one mapping; otherwise, the period is less than $M/4$ and a many-to-one mapping may occur.

Step 4. The vector V is permuted by using a secret key Key_5 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 achieved in which five MSBs planes of the cover image are preserved and three LSBs planes are replaced with the recovery bits and authentication bits

5.3.2 Content Restoration Procedure

Suppose an attacker alters original contents of the watermarked image without changing the image size. Firstly it needs to locate the tampered blocks using the embedded authentication bits and multitasking bits, then recover the tampered blocks

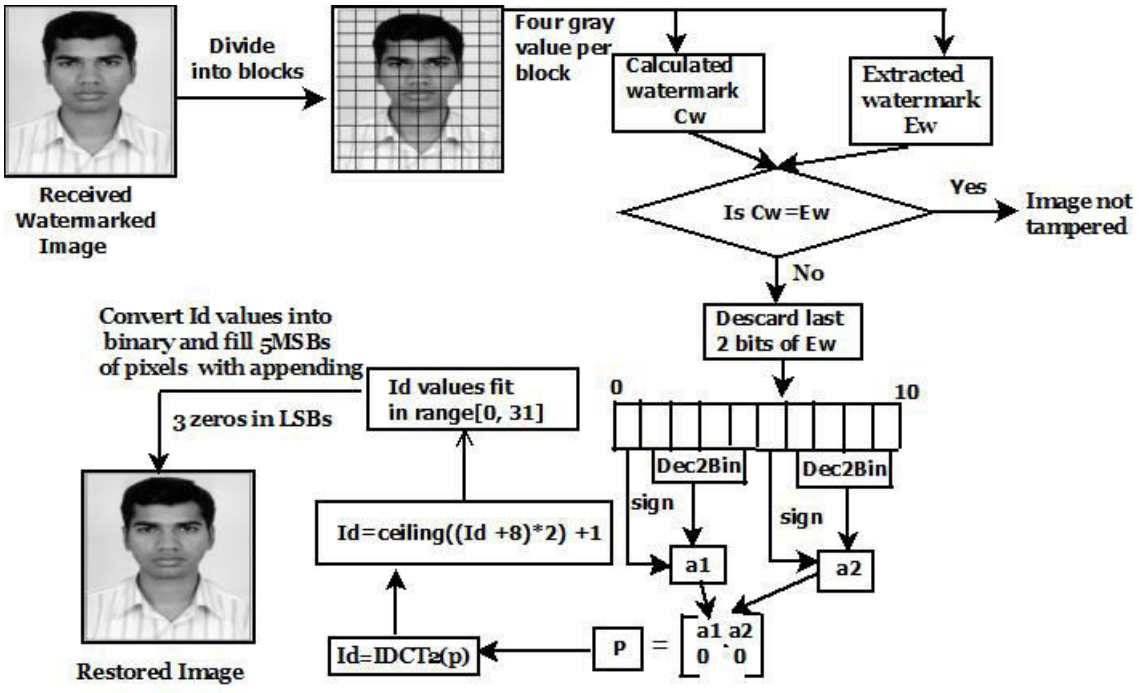


FIGURE 5.5: Block diagram for watermark extraction and image restoration procedure.

by using the multitasking bits extracted from the corresponding mapped blocks as shown in Fig. 5.5. Therefore, the watermark extraction process can be divided into two phases, tampered block identification and tempered block restoration.

5.3.2.1 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 procedure of tampered block identification procedure is described in the following.

Step 1. First of all, receiver extracts twelve bits from each corresponding block using Key_6 which was used at the time of block mapping, in a twelve bit vector V for each block and reshuffle it using Key_5. Also enter all the secret Keys that is used at embedding time.

Step 2. Generate the pseudo random binary matrix of size $\frac{X}{2} \times \frac{Y}{2}$ using secret key Key_4 which was used at the time of watermark embedding.

Step 3. Calculate the authentication bits with the help of secret Key, Key_2 and

Key_3 and multitasking bits for each block using Key_1 as discussed in Subsection 5.3.1.1 and 5.3.1.2.

Step 4. Now compare the calculated watermark bits with extracted watermark bits, if mismatch is found, mark that block as invalid, otherwise, mark the block valid.

Step 5. Assign the white color as seen in the image (i.e. 255) to the invalid blocks.

5.3.2.2 Tampered Block Restoration Procedure

After the tamper block identification procedure, all blocks are marked either invalid or valid. Once all invalid blocks are identified, needs to be restored. The restoration procedure of an invalid block B can be performed from extracted watermark as follows.

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. Convert the 2^{th} to 5^{th} index position value of vector V into decimal and store in a variable, say a_1 with proper sign that is decided by V first index position value by using equation 5.6. Similarly convert the 6^{th} to 10^{th} index position value of vector V into decimal and store in a variable, say a_2 with proper sign that is decided by V sixth index position value.

Step 2. Initialize a 2×2 empty matrix R and fill two locations of the first row by a_1 and a_2 as shown in Fig. 5.5. The remaining two locations of second row are filled with zeros.

Step 3. Take Inverse DCT of matrix R in the matrix I_d of size 2×2.

$$I_d = InverseDCT2(R) \tag{5.27}$$

Step 4. Fit the I_d values in the range $\in [0, 31]$ using equations 5.28 and 5.29.

$$I_d^{ij} = \left\lceil (I_d^{ij} + 8) + 1 \right\rceil ; i = 1, 2; j = 1, 2. \tag{5.28}$$

$$I_d^{ij} = \begin{cases} 0 & \text{if } I_d^{ij} < 0 \ ; \\ r_{ij} & \text{if } 0 \leq I_d^{ij} \leq 31 \ ; \\ 31 & \text{if } I_d^{ij} > 31. \end{cases} \tag{5.29}$$

Step 5. Now convert the I_d^{ij} value into five bits and fill the five MSBs of corresponding pixel in tampered 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 6. Finally after the completion of restoration of the block B, mark it as a valid block.

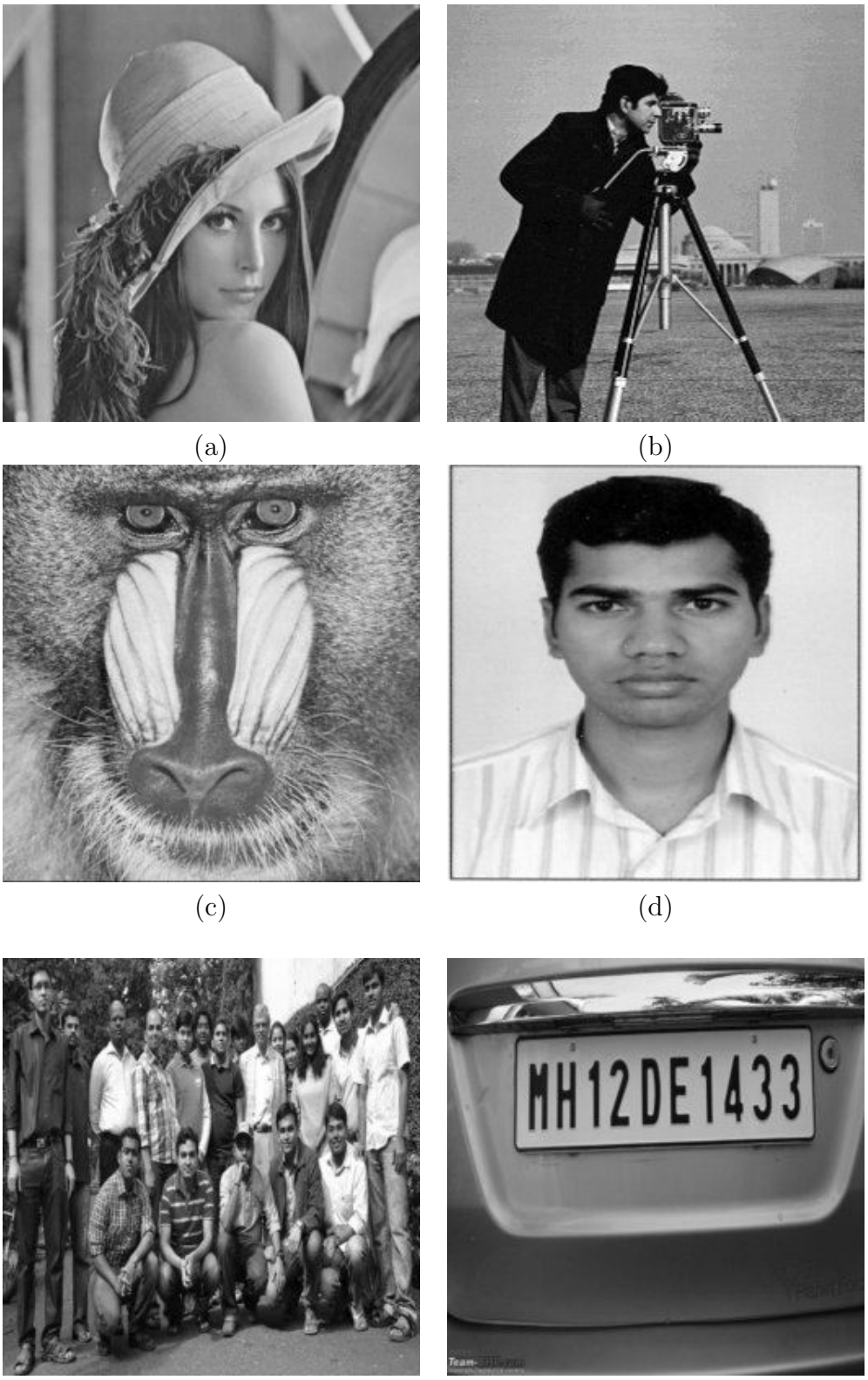
Case 2: When mapped block marked as invalid block

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. Finally 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 that is very similar to the original watermarked image with approximate intensities that is demonstrated by the experimental results.

5.4 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. 5.6 shows some of the test images which were used in the experiments and Table 5.1 shows the corresponding watermark embedding PSNR and NCC value. As the embedding PSNR and NCC values are very high, so it is highly difficult to differentiate between the watermarked



(a) (b) (c) (d) (e) (f)

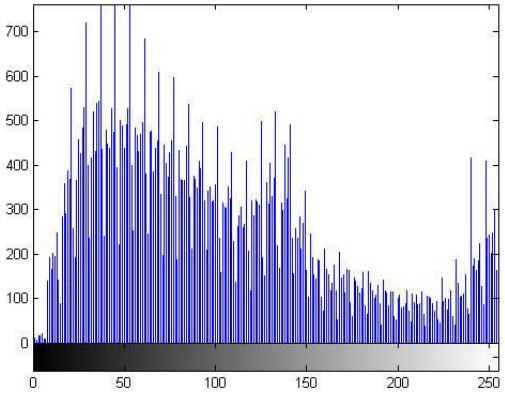
FIGURE 5.6: Set 1. Test images used in our experiments (a) Lena (b) Cameraman (c) Baboon (d) Own Photo (e) Group Photo (f) Number Plate.

(a)Watermarked Image

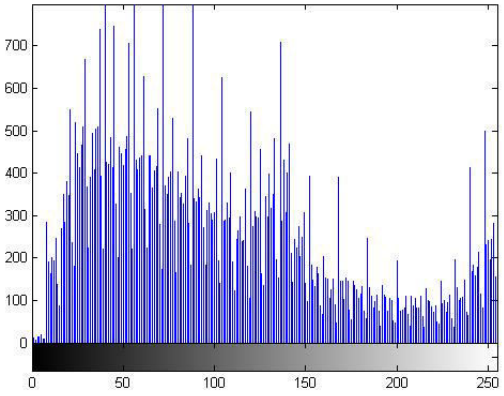
(b) Tampered Image

(c) Tampered detected Area

(d) Restored Image



(e)Watermarked Image Histogram



(f)Restored Image Histogram

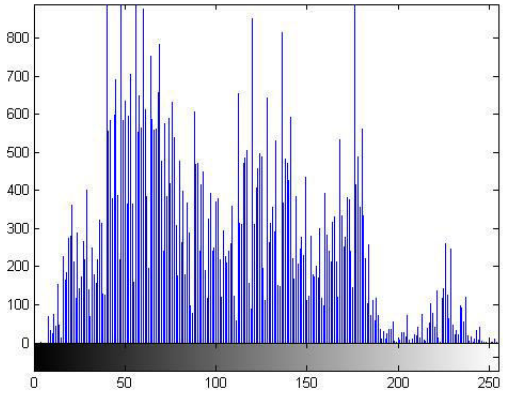
FIGURE 5.7: Tampered detection and restoration of Group Photo.

(a)Watermarked Image

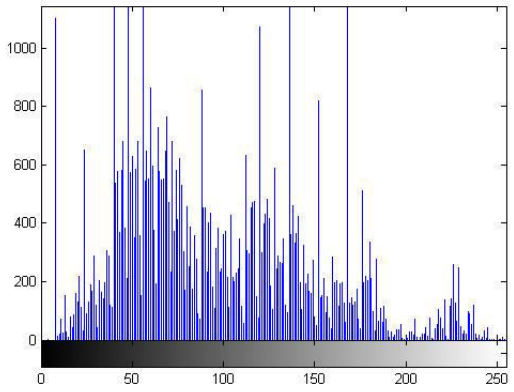
(b) Tampered Image

(c) Tampered detected Area

(d) Restored Image



(e)Watermarked Image Histogram



(f)Restored Image Histogram

FIGURE 5.8: Forgery detection and restoration of a Number Plate.

and the original image in vision. Table 5.1 also lists the watermark embedding time of proposed scheme. This time is very less ($\sim$ 6 seconds) for an image with a size of 256×256. This low embedding time indicates that the proposed algorithm is simple and efficient.

Fig. 5.7 shows the tampered detection and restoration of Group Photo. Fig. 5.7(a) gives the watermarked version of Group Photo. The PSNR and NCC values due to watermark embedding are 40.69 dB and 0.9988 respectively. These high PSNR and NCC values indicate that the distortion due to watermark is imperceptible. The corresponding embedding time is 6.4428 seconds. We modified the watermarked image by shuffling the head portion(i.e. sensitive content) between the people. This tampered is shown in Fig. 5.7(b). By using the authentication bits embedded in the image, all the modified blocks were detected as shown in Fig. 5.7(c), in which white regions are indicating tampered blocks while black regions are indicating reserved blocks. The final image restored from the extracted multitasking bits is shown in Fig. 5.7(d). The PSNR and NCC of this restored image are 41.40 dB and 0.9983 respectively with the reference of watermarked image. The restoration time is 3.8376 seconds. The histogram of the watermarked image as shown in Fig. 5.7(e), is almost similar to the histogram of the restored image as shown in Fig. 5.7(f).

Similarly Fig. 5.8 shows the Number Plate forgery detection and recovery which is very useful in court evidence. Fig. 5.8(a) shows the watermarked version of Number Plate. The PSNR and NCC values due to watermark embedding are 39.72 dB and 0.9968 respectively. The embedding time is 6.368 seconds. We modified the watermarked image by replacing the original written text and digits with fake information as shown in Fig. 5.8(b). The modified blocks were detected as shown in Fig. 5.8(c) and final restored image is shown in Fig. 5.8(d) in which we gets exactly original text and digits. The PSNR and NCC of this restored image are 38.71 dB and 0.9968 respectively with the reference of watermarked image. The restoration time is 4.0716 seconds. The histogram of the watermarked image as shown in Fig. 5.8(e), is almost similar to the histogram of the restored image as shown in Fig. 5.8(f).

Table 5.2 shows PSNR value of the restored image in tampered area with respect to the different tampering rates and Fig. 5.9 shows the corresponding graph. Five gray images Lena, Cameraman, Group Photo, Baboon, Own Photo of size 256×256

Table 5.1: Essential information observed during watermark embedding.

Cover Image (Set 1: Gray Image)	PSNR (Embedding)	NCC (Embedding)	Embedding Time (in Sec.)
Lena	39.31 dB	0.9977	6.5520
Cameraman	39.00 dB	0.9986	6.5832
Baboon	39.03 dB	0.9977	6.3960
Own Photo	39.36 dB	0.9990	6.2556
Group Photo	40.69 dB	0.9988	6.4428
Number Plate	39.72 dB	0.9982	6.3648

were used as the covers image. Similar results were found after doing experiments on other images. In Fig. 5.9, it can be seen that the curve of PSNR with respect to tampering rate decreases smoothly but it has been observed that, even if the tampering rate is up to 50%, the restored contents have PSNR values more than 33.16 dB which are quite satisfactory.

Table 5.2: PSNR(dB) of restored content in the tampered area with different tampering rates.

Cover Image	Tampering rate								
	5%	10%	20%	25%	30%	35%	40%	45%	50%
Lena	48.56	45.09	40.58	39.50	38.25	37.48	36.84	36.32	35.79
Group Photo	44.20	41.14	37.62	36.68	35.91	35.40	34.92	34.29	33.85
Baboon	42.39	39.92	37.00	36.06	35.22	34.65	34.12	33.55	33.16
Cameraman	45.03	42.45	38.77	37.48	36.37	35.51	34.76	34.05	33.53
Own Photo	42.04	39.98	37.36	36.59	35.68	35.04	34.57	34.13	33.73

Fig. 5.10 shows some color images which were used in experiments. Their corresponding watermarked images are shown in Fig. 5.11. The watermark is embedded

Table 5.3: Essential information observed during watermark embedding.

Cover Image (Set 2.Color Images)	PSNR (Embedding)	NCC (Embedding)	Embedding Time (in Sec.)
Color Lena	39.7928 dB	0.9973	19.6405
Color Boat	39.9285 dB	0.9984	19.7653
Color Baboon	39.5467 dB	0.9980	19.8433
Color House	39.1570 dB	0.9981	19.9525
Color Pepper	40.1907 dB	0.9980	19.9993
Color Woman	39.7255 dB	0.9980	19.6405

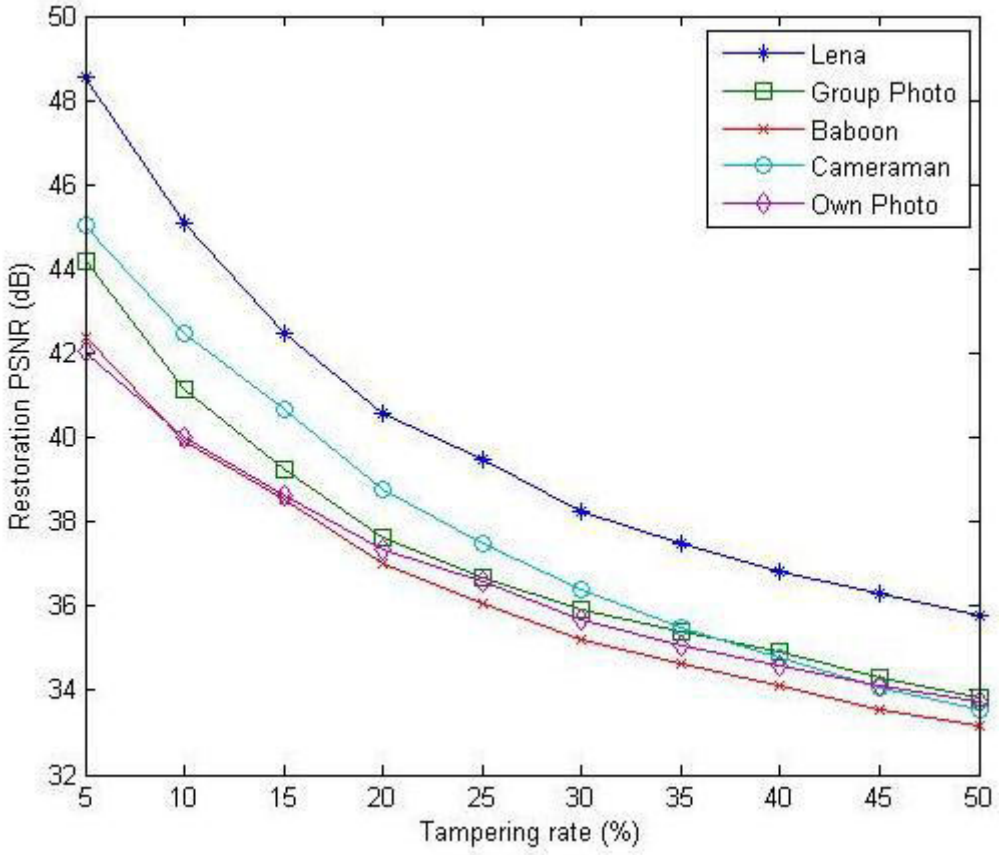


FIGURE 5.9: PSNR of restored content with respect to the tampering rates.

in the R,G,B channels. The PSNR and NCC values of watermarked image relative to the original color images are shown in Table 5.3. As the embedding PSNR and NCC values are very high, the visual qualities of the watermarked images are excellent. Table 5.3 also lists the watermark embedding time of proposed scheme. It is very low ($\sim$ 20 seconds) for an image with a size of 256×256.

Fig. 5.12, shows the tampered detection and restoration of the color image *Color Boat*. The watermarked image is shown in Fig. 5.12(a). The PSNR and NCC values due to watermark embedding are 39.9285 dB and 0.9984 respectively. These high PSNR and NCC values indicate that the distortion due to watermark is imperceptible. The corresponding embedding time is 19.7653 seconds. The object addition attack is done on watermarked image in which the boat part in the watermarked image is inserted into two arbitrary positions. The tampered image is shown in Fig. 5.12(b). The tamper detection result is shown in Fig. 5.12(c), in which non black

(a) (b) (c)

(d) (e) (f)

FIGURE 5.10: Set 2. Test images (a) Color Lena (b) Color Boat (c) Color Woman (d) Color Baboon (d) Color House (e) Color Pepper.

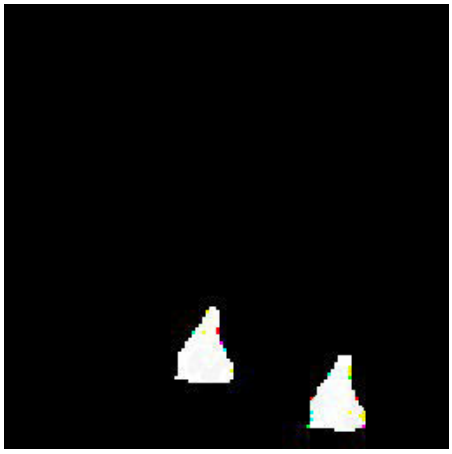
(a) (b) (c)

(d) (e) (f)

FIGURE 5.11: Watermarked images of Set 2 test images as given in Fig 5.10.

(a)Watermarked Image

(b) Tampered Image



(c) Tampered detected Area

(d) Restored Image

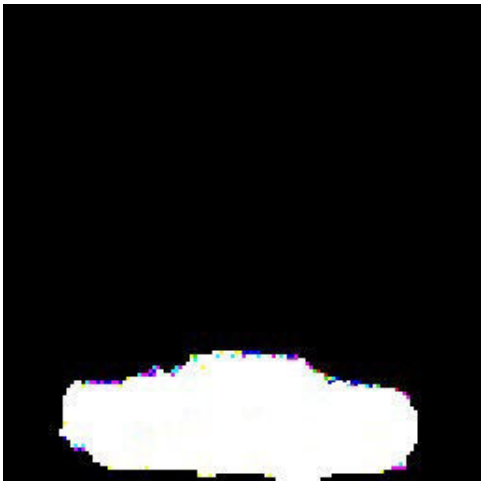
FIGURE 5.12: Tampered detection and restoration of Color Boat.

regions are indicating tampered blocks.The restored image from the extracted multitasking bits is shown in Fig. 5.12(d). The PSNR and NCC of this restored image are 46.5702 dB and 0.9998 respectively with the reference of watermarked image. The restoration time is 10.1869 seconds. The experiment reveals that the ability of the proposed method to detect and localization tampering with restoration quality is adequate.

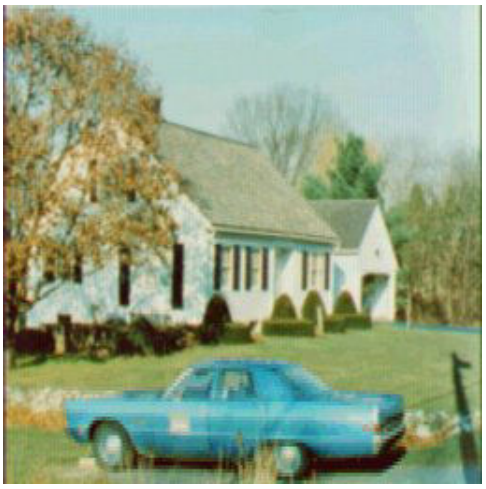
Similarly Fig. 5.13, shows the tampered detection and restoration of the image "Color house". The watermarked image is shown in Fig. 5.13(a). The PSNR and NCC values due to watermark embedding are 39.2398 dB and 0.9984 respectively. The embedding time is 19.9525 seconds.The object removal attack is done on the watermarked image in which the car part in the watermarked image is removed

(a)Watermarked Image

(b) Tampered Image



(c) Tampered detected Area



(d) Restored Image

FIGURE 5.13: Tampered detection and restoration of Color House.

and grass color is spread over it. The tampered image is shown in Fig. 5.13(b). The tamper detection result and the recovered image are shown in Figs. 5.13(c) and 5.13(d), respectively. The PSNR and NCC values of this restored image are 39.7690 dB and 0.9972 respectively with the reference of watermarked image. The restoration time is 12.1993 seconds.

Table 5.4 compares several fragile watermarking schemes with restoration capability. For doing comparison we have taken Lena and Baboon as cover images. At the tampering rate 33 %, we have listed the PSNR of the restored images. PSNR of restored images in the proposed scheme is effectively higher than the method used

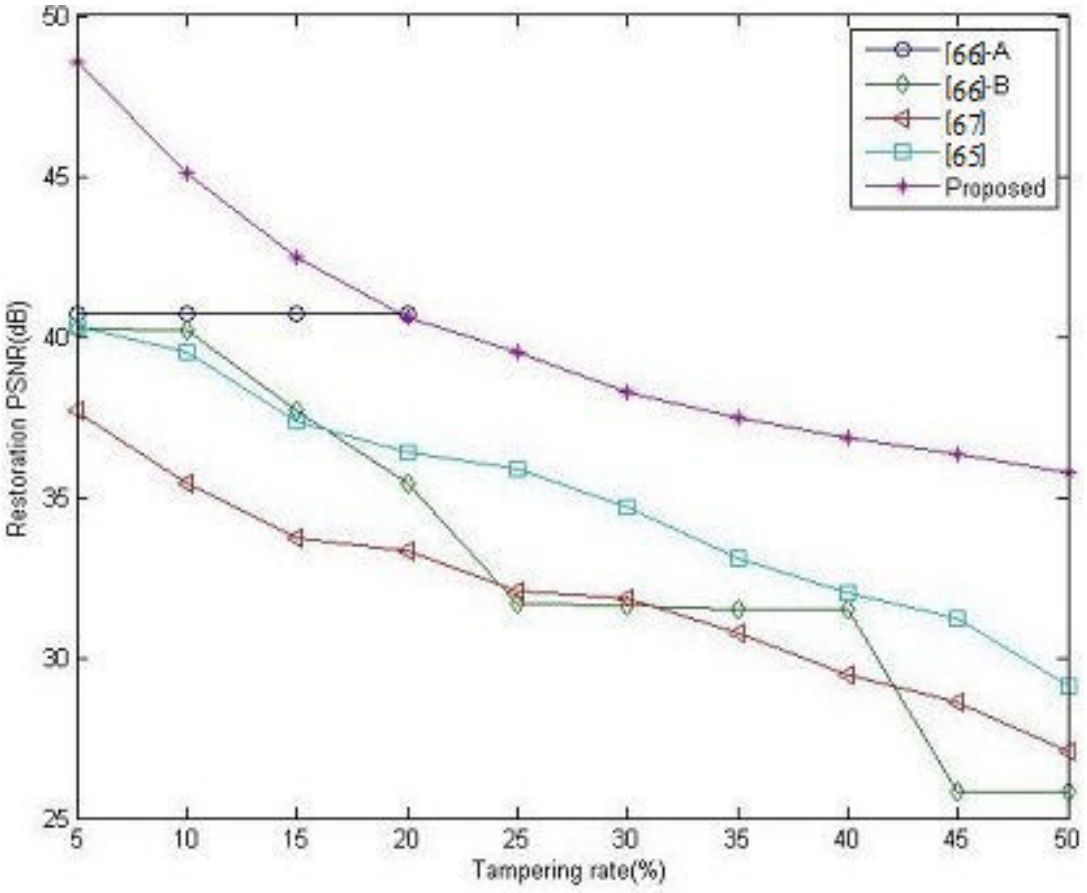


FIGURE 5.14: Reconstruction PSNR for Lena image under varying tampering rates.

in [64], [65], [66](two methods called as [66]-A and [66]-B) and [67]. The proposed scheme effectively restores image and also provides high accuracy in tampered pixel localization due to use of small size blocks. Watermarking Schemes proposed in [64], [65], [66] and [67] are using blocks of size 8×8. If only a single pixel in a block of size 8×8 is tampered, the whole sixty four pixels of that block will be treated as tampered region. In [64], [65], [66] and [67], the accuracy of tamper localization is decreased and encounter blocking artifacts because of these schemes are using such large blocks size.

The restored PSNR(dB) scores for test images Lena and Baboon are shown in Fig. 5.14 and Fig. 5.15. The plots not only clearly show the threshold tampering rates, but also demonstrate the characteristic behavior of the systems. The schemes [65],[66]-A and [67]reveal systematic deterioration of the reconstruction fidelity. The scheme [65] is more susceptible to the distributions of details in the

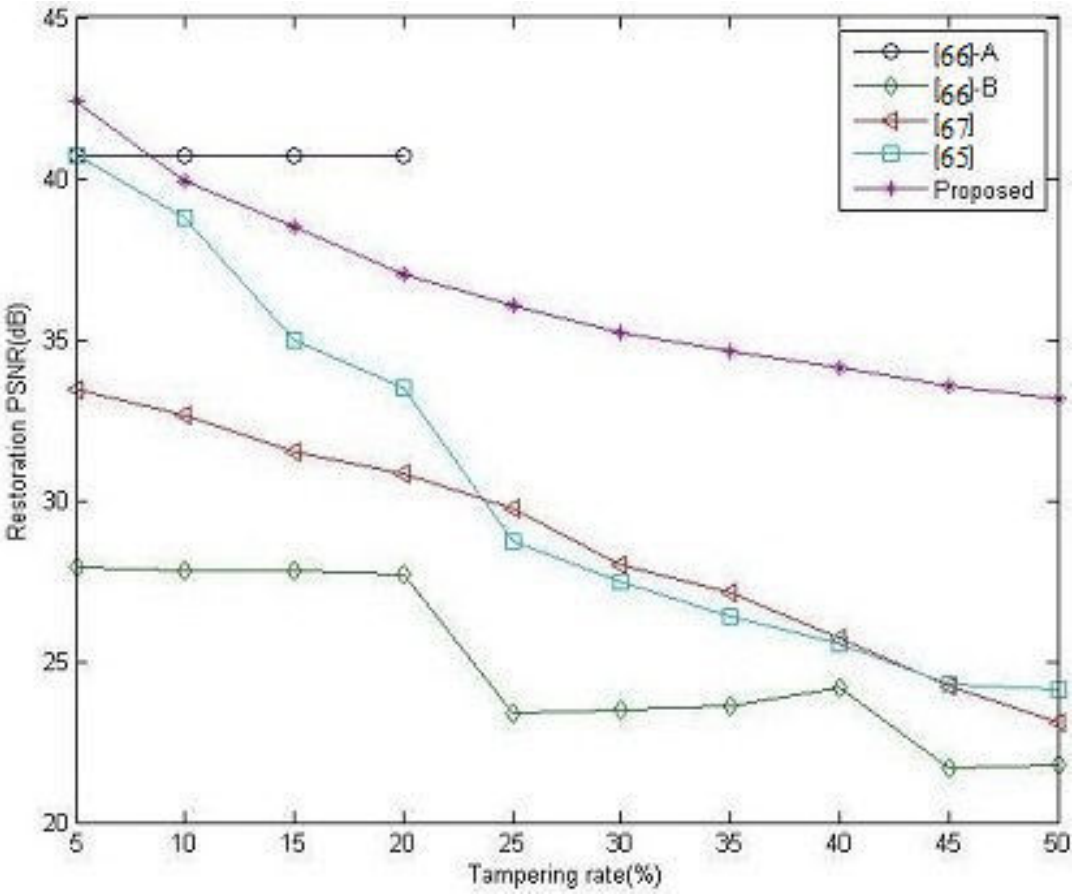


FIGURE 5.15: Reconstruction PSNR for Baboon image under varying tampering rates.

image. For images with large areas of solid low detail blocks, the curves may not be monotonic. The scheme [67] operates directly on pixel intensities, and is not affected by the problem. The plots also demonstrate the expected three distinct quality levels for the [66]-B scheme. The method 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]. Also, the proposed scheme can get very ideal recovered image when the tampering rate is low. The proposed scheme is also free from blocking artifacts using small size blocks. Thus proposed scheme is more flexible than previously existing schemes.

TABLE 5.4: Comparison of restoration capability using two cover images at 33% tampering rate (γ).

Water-marking Scheme	Embedding PSNR when cover image Lena	Restoration PSNR when cover image Lena	Embedding PSNR when cover Baboon	Restoration PSNR when cover Baboon	Condition of Restoration
[64]	37.90 dB	34. 33 dB	37.70 dB	27.87 dB	$\gamma < 35\%$
[67]	37.82 dB	32.56 dB	37.67 dB	28.43 dB	$\gamma < 54\%$
[66]-A	37.90 dB	...	37.90 dB	...	$\gamma < 24$ %
[66]-B	37.90 dB	31.5 dB	37.90 dB	23.52 dB	$\gamma < 66\%$
[65]	37.90 dB	33.20 dB	37.90 dB	26.20 dB	$\gamma < 60\%$
Proposed	39.31 dB	37.87 dB	39.03 dB	35.01 dB	$\gamma \leq 50\%$

5.5 Conclusion

In this chapter, a quantization and DCT based novel self-embedding fragile watermarking scheme is presented to extreme localization and restoration properties. In the process of watermark bits generation and embedding, a small non overlapping block sized 2×2 is used to improve the accuracy of localization. Experimental results demonstrate that the performance of the proposed scheme is better than that of previous techniques. The principal content of tampered image can still be restored with high accuracy up to 50% tampering rate. Since the watermark is embedded into three LSBs planes, the watermark data may be destroyed by some image processing operations. Therefore in the future a semi-fragile watermarking scheme that can simultaneously resist some common image processing operations with good restoration capability needs to be developed.