Multimed Tools Appl (2017) 76:6389–6428
DOI 10.1007/s11042-015-3198-9

# A self recoverable dual watermarking scheme for copyright protection and integrity verification

**Priyanka Singh[1] · Suneeta Agarwal[1]**

Received: 1 April 2015 / Revised: 16 November 2015 / Accepted: 22 December 2015 /
Published online: 9 February 2016


**Abstract** Dual watermarking implies embedding of robust as well as fragile watermarks into the same cover image. It facilitates integration of copyright protection and integrity verification into the same scheme. However, most of such existing state of art approaches either lacked the feature of tamper detection and original content recovery or provided an approximation using coarser block level approach. The proposed self recoverable dual watermarking scheme integrates all the aforementioned functionalities of copyright protection, tamper detection and recovery into one scheme. The scheme is independent of the order of embedding of robust and fragile watermarks as these are embedded in different regions of the cover image. It performs tamper detection and recovery, both at the pixel level. The scheme obtains recovery information for each $2 \times 2$ image block in just eight bits which are further encoded to only four bits via mapping table. This reduction in recovery bits allows efficient embedding of copyright information which is tested against comprehensive set of attacks. The scheme is found to be robust against noises, filtering, histogram equalization, rotation, jpeg compression, motion blur etc. Besides the normalized cross correlation value, the evaluation of the extracted copyright information is also being done using various objective error metrics based on mutual relation between pixels, their values and locations respectively. The imperceptibility and visual quality of the watermarked as well as recovered image is found to be satisfactorily high. Three major categories of images: natural, texture as well as satellite have been tested in the proposed scheme. Even minute alterations can be chalked out as the detection accuracy rate has been enumerated on pixel basis. The scheme can tolerate tampering ratios upto 50 percent though the visual quality of

✉ Priyanka Singh
priyankaap@gmail.com

Suneeta Agarwal
suneeta@mnnit.ac.in

1 Motilal Nehru National Institute of Technology, Allahabad, India

the recovered image deteriorates with increasing tampering ratio. Comparative results based on normalized cross correlation, probability of false acceptance, probability of false rejection and peak signal to noise ratio metrics validate the efficacy of the proposed scheme over other existing state of art approaches.

**Keywords** Self recoverable · Dual watermarking · Copyright protection · Integrity verification · Normalized cross correlation (NCC) · Probability of false rejection (PFR) · Probability of false acceptance (PFA) · Peak signal to noise ratio (PSNR)

## 1 Introduction

The advancement in technology has eased life with lots of amenities available at hand, accompanied with easy sharing and distribution of multimedia content across the internet. However, the rate of illegal distribution and malicious tampering increased exponentially through the easy and online availability of various softwares and tools. Hence, techniques securing the multimedia content like cryptography, digital signatures, steganography, watermarking etc are promoted and serve as active research areas. Each technique is designed for a specific purpose like cryptography is meant for delivering documents unreadable, steganography conceals the very existence of the message whereas watermarking assures the integrity of the multimedia content and proves the rightful ownership. Watermarking is basically a two phase technique. First phase involves watermark embedder that embeds a secret information into the cover image to obtain a watermarked image that is transmitted via internet. In the second phase, the watermark extractor extracts this secret information on the receiving end to proof the integrity of the content.

Based on the purpose, watermarking schemes can be mainly categorized as fragile, semi-fragile and robust watermarking schemes. Fragile watermarking aims for authenticity verification whereas robust watermarking is meant for proving the rightful ownership. Intermediate schemes between these two extremes are termed as semi-fragile watermarking schemes. Variation of these schemes called as dual watermarking schemes are gaining attention these days. Dual watermarks are combination of both, fragile as well as robust watermarks and could fetch benefits of both ownership assertion as well as integrity verification. Dual watermarking schemes can be broadly categorized into three main types: First type of schemes generate a dual watermark from the combination of fragile and robust watermarks and then embed it into the cover image as the typical watermarking scheme. If this watermark is tampered, then the whole purpose of its dual functionality is destroyed. Second kind of dual watermarking schemes follow a pipeline pattern while embedding of fragile and robust watermarks. The embedding and functionality of one watermark must not get affected by embedding of the other watermark. Third kind of dual watermarking schemes embed both fragile as well as robust watermarks into separate areas of the cover image. They are considered to be most versatile schemes as they become independent from any of the constraints of interference while embedding or functioning of the respective watermarks.

In literature, varied robust watermarking schemes have already been proposed [7, 12, 16, 17, 25, 31, 36] to maintain the integrity of the content. Copyright protection schemes for e-government document images based on discrete cosine transform (DCT) with zigzag space-filling curve (SFC) was proposed in [6], singular value decomposition (SVD) exploiting luminance masking in [1, 4] where the singular values of the DCT transformed coefficients of the watermark was embedded into the left singular value of the host image. The genetic algorithm was utilized to find the optimum value of the scaling factor depending on the content of the image. A reversible watermarking scheme for authentication of relational databases has been proposed in [2] where exact original document was recovered even though 95 % tuples of watermarked data were deleted. Another svd based copyright protection scheme presented in [5] increased the reliability of the scheme by embedding the principal contents of the watermark into DCT and DWT domains. The robustness factor was also enhanced via incorporation of particle swarm optimization for finding suitable scaling factors. However, many malicious attacks could not be detected by such schemes. Hence, came the need of fragile watermarking schemes that could sense even minute manipulations.

Tamper assessment function was proposed in literature [13] in this respect. Various transform domain and quantization based schemes have been proposed to enhance the security of the scheme and prevent tampering [39]. A fragile watermarking scheme for authentication of H.264/AVC content having high sensitivity to video attacks was proposed in [3]. Minimum deterioration of perceptual quality was guaranteed by incorporation of spatiotemporal analysis. However, they failed against incidental manipulations [18, 22]. Thus, intermediate kind of schemes that could tolerate incidental distortions along with sensitivity to malicious attacks came into picture [23, 26]. Though a lot of schemes have already been proposed, but still a lot of improvement is needed. A integrity check authentication scheme has been presented in [28]. It detected the tampers but localization accuracy was compromised. In [29], the accuracy of tampered regions increased but it could not perform recovery of the altered regions. Hence, schemes with dual functionalities are more preferable nowadays. A scheme with both authentication as well as recovery was proposed in [33], but the security and visual quality was compromised for its sake. Secret keys are often used to enhance the security of the schemes [34]. If security of these schemes is compromised, then the whole algorithm fails. Hence, correlating the watermark with pixel values of the cover image and thereby, embedding coefficients in other regions would serve as safety enhancement against such distortions [33]. A dual watermarking utilizing both spatial as well as frequency domain has been proposed in [9]. Firstly, 5/3 wavelet transform of the cover image was calculated and a robust watermark was embedded into the middle frequency coefficients. Thereafter, LSB substitution was done in spatial domain to embed the fragile watermark. However, this scheme suffered from the limitation that the embedding of fragile watermark affected the extraction of the robust watermark. One such dual watermarking scheme based on DCT coefficient, separating the integer and decimal portions to embed the robust and fragile watermarks has been proposed in [18]. This ensured that the two watermarks doesn't affect each other. However, there was no means to recover the lost content. In this paper, a selfrecoverable dual watermarking scheme providing all the three functionalities of ownership assertion, tamper detection and recovery has been proposed. It minimized the storage requirements for embedding of recovery information to just four bits for each $2 \times 2$ sized image block and utilized the space for embedding of copyright information. The tamper detection and recovery are both performed at pixel level.

The rest of the paper is organized as follows: Section 2 describes the proposed approach in detail, experimental results with analysis are presented in Section 3. Conclusions along with the scope of future work has been concluded in Section 4 followed by references.

## 2 Proposed methodology

The proposed watermarking scheme consists of six main phases: generation and embedding of recovery information, embedding of copyright information, generation and embedding of authentication information, ownership verification via extraction of copyright information, tamper detection and recovery of tampered image.

Consider a gray scale cover image $I$ having $M$ rows and $N$ columns where $M$ and $N$ are even . Then $T$ represent the total number of pixels ($T = M \times N$). Let the intensity value of each pixel of the cover image be denoted by $P_n \in [0, 255]$ where $n = 1, 2, 3, ...., T$.

The individual bit of $P_n$ is denoted by $b(P_n, 8)$, $b(P_n, 7)$, $b(P_n, 6)...b(P_n, 1)$ and it can be represented in binary form as follows:

$$b(P_n, m) = \lfloor \frac{P_n}{2^{m-1}} \rfloor mod2, m = 1, 2, 3, .....8 \quad (1)$$

The decimal equivalent can be represented as:

$$P_n = \sum_{m=1}^{8} b(P_n, m)2^{m-1} \quad (2)$$

A principal content image $I_c$ is formed from the cover image $I$ by obtaining the major information content of the cover image via taking the five most significant bits (MSBs) of all pixels. All the phases of the proposed watermarking scheme will take this principal content image $I_c$ as input for further processing.

A basic flow of the proposed scheme is depicted in Fig. 1. The principal content image serves as input on the sender end where generation and embedding of recovery, embedding of copyright information and generation and embedding of authentication information is done which produces the watermarked image ($I_w$) as output. This output image is transmit-

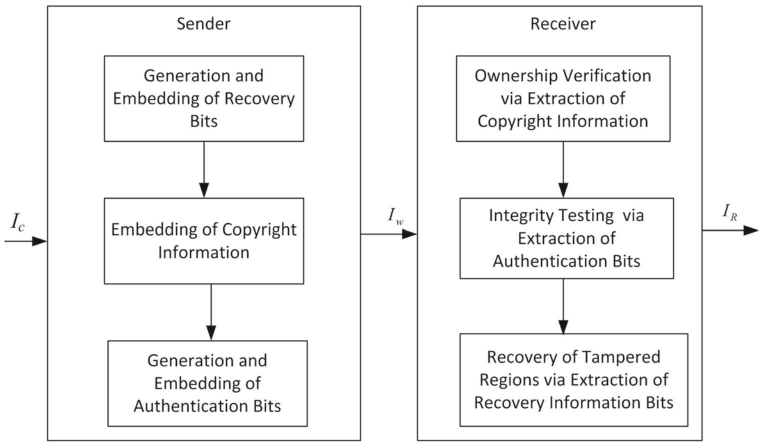



**Fig. 1** Basic flow of the Proposed Scheme

ted to the receiver end where the ownership verification is done via extraction of copyright information. Then, its authenticity is checked and in case of tampering, a recovery image is obtained as output. The algorithmic flowchart of the proposed scheme has been depicted in Fig. 2 along with symbols, abbreviations and functions listed in Tables 1 and 2 respectively. The detailed approach is presented as follows:

## 2.1 Generation and embedding of recovery information

In this phase,the principal content image $I_c$ of size $M \times N$ is divided into non overlapping blocks $B_i$ of size $2 \times 2$ pixels each. A eight bit recovery information ($R_{B_i}$) is generated for each of the image blocks $B_i$. The method for recovery generation is done in the spatial domain whereas embedding is done in the frequency domain to maintain its robustness in case of tampers.

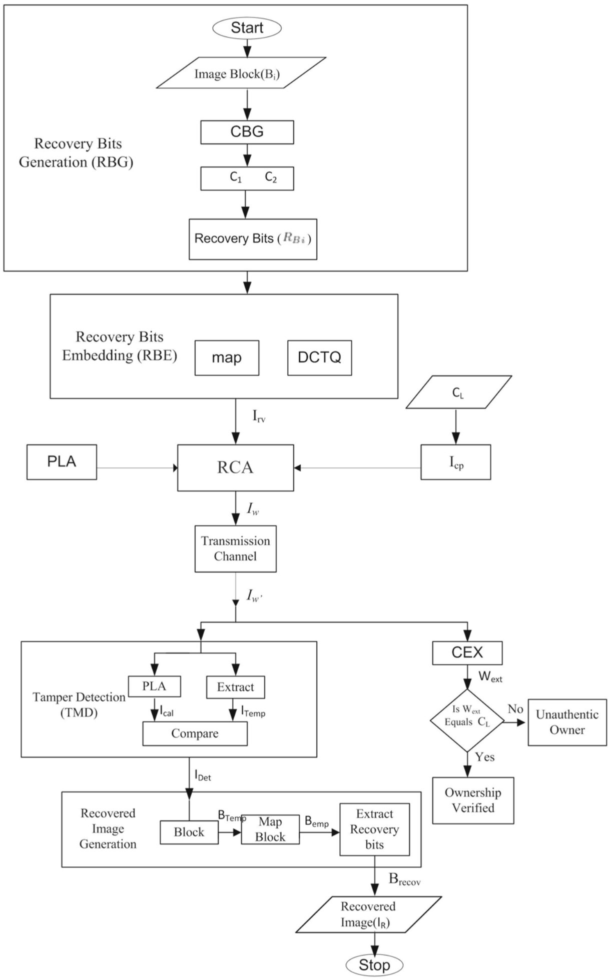



**Fig. 2** Flowchart of the Proposed Scheme

**Table 1** Symbols and abbreviations used in the flowchart

| Symbol or Abbreviation Used | Significance |
|---|---|
| $B_i$ | Image Block |
| $C_1, C_2$ | Two Clusters of Block Pixels |
| RBG | Recovery Bits Generation |
| CBG | Cluster Based Generation |
| $R_{B_i}$ | Recovery Bits of the Block (8 Bits) |
| RBE | Recovery Bits Embedding |
| $D^j$ | Mapped Value |
| DCTQ | DCT based quantization |
| RCA | Recovery Copyright Authentication |
| PLA | Pixel Level Authentication |
| $P_{xy}$ | Pixel Intensity Value |
| $x, y$ | co-ordinate values of the pixel $P_{xy}$ |
| $M, N$ | Size of Cover Image |
| $C_L$ | Copyright Logo |
| $W_L$ | Bit sequence corresponding to $C_L$ |
| CPE | Copyright Embedding |
| $I_{cp}$ | Copyright Information Encoded Image |
| $\Psi$ | Authentication Matrix of size $M \times N$ generated via PLA |
| $I_w$ | Watermarked Image |
| $I_w^{'}$ | Suspected Watermarked Image |
| CEX | Copyright Information Extraction |
| $W_{ext}$ | Extracted Watermark |
| TMD | Tamper Detection |
| $I_{cal}$ | Calculated Authentication Bits Matrix of size $M \times N$ |
| $I_{Temp}$ | Extracted Authentication Bits Matrix of size $M \times N$ |
| $I_{Det}$ | Tamper Detected Image of size $M \times N$ where black regions signify untampered areas whereas white regions represent tampered ones |
| $B_{Temp}$ | Tampered Blocks |
| $B_{emp}$ | Contains Recovery Information |
| $B_{recov}$ | Recovered Blocks |
| $I_R$ | Recovered Image of size $M \times N$ |

The detailed methodology for the block recovery generation has been enlisted as follows:

1. Image Block Division: The principal content image $I_c$ of size $M \times N$ is divided into non overlapping blocks $B_i$ of size $2 \times 2$ pixels each where :

$$B_i = \begin{bmatrix} X_{p,q} & X_{p,q+1} \\ X_{p+1,q} & X_{p+1,q+1} \end{bmatrix} \tag{3}$$

where, $X_{p,q}$, $X_{p,q+1}$, $X_{p+1,q}$, $X_{p+1,q+1}$ represent the neighboring block pixels at $(p,q)^{th}$, $(p,q+1)^{th}$, $(p+1,q)^{th}$ and $(p+1,q+1)^{th}$ co-ordinates of $I_c$ respectively.

**Table 2** Functions used in the algorithms

| Function Used | Significance |
|---|---|
| $length(C_i)$ | returns the number of elements in $C_i$ |
| $min(B_i)$ | returns minimum element of the block $B_i$ |
| $max(B_i)$ | returns maximum element of the block $B_i$ |
| map | returns mapped value of the input arguments according to the mapping table. |
| $Extract^u$ | returns authentication bit from the $u^{th}$ LSB position of the input argument |
| Compare | compares pixelwise values between the input arguments |
| Exor | Bitwise Exor of the input arguments |
| Block | finds tampered blocks corresponding to tampered pixel locations |
| MapBlock | maps tampered blocks with its corresponding recovery information embedded blocks |

Hence, total number of such blocks formed $T_b$ :

$$T_b = \frac{M \times N}{2 \times 2} \tag{4}$$

2. Block Recovery Generation: The recovery information is obtained for each block depending upon the content of the block, directly from the pixel values using the recovery bit generation ($RBG$) and cluster based generation ($CBG$) methods as Algorithm 1 and Algorithm 2.

**Algorithm 1** RBG

**INPUT:** Image Block $(B_i)$ of size $2 \times 2$
**OUTPUT:** $R_{B_i}$ is a vector of size $1 \times 8$
**Ensure:**
(1) $m_2 > m_1$
(2) $length(C_i)$ returns the number of elements in $C_i$
(3) $C_1$ and $C_2$ are two clusters containing elements of $B_i$ in vector form
(4) $B_i^j$ denotes the $j^{th}$ element of image block $B_i$

$\forall B_i$,where $i = 1, 2, \ldots, T_b$ ▷ For each $2 \times 2$ block of image
$[C_1\ C_2] \leftarrow CBG(B_i)$ ▷ Classify the block elements into two clusters $C_1$ and $C_2$
$l_1 \leftarrow length(C_1)$ ▷ Count number of elements belonging to cluster $C_1$
$l_2 \leftarrow length(C_2)$ ▷ Count number of elements belonging to cluster $C_2$
$m_1 \leftarrow \sum_{i=1}^{l_1} C_1^i / l_1$ ▷ Compute mean of cluster $C_1$
$m_2 \leftarrow \sum_{i=1}^{l_2} C_2^i / l_2$ ▷ Compute mean of cluster $C_2$
$T_1 \leftarrow \lceil m_1/4 \rceil$ ▷ Reduce number of bits required to store mean value $m_1$
$T_2 \leftarrow \lceil m_2/4 \rceil$ ▷ Reduce number of bits required to store mean value $m_2$
$D \leftarrow T_2 - T_1$ ▷ Difference of the reduced mean values
$R_{B_i}^{8-u} \leftarrow \lfloor T_1/2^u \rfloor mod 2$ ▷ Storing binary equivalent of smaller mean value in the recovery vector
where $u \leftarrow 0$ to 2
$R_{B_i}^5 \leftarrow D$ ▷ Storing binary equivalent of difference value in the recovery vector
**for** $j \leftarrow 1$ to 4 **do** ▷ Indicating cluster for each image block element
  **if** $B_i^j \in C_1$ **then**
    $R_{B_i}^{5-j} \leftarrow 0$
  **else**
    $R_{B_i}^{5-j} \leftarrow 1$
  **end if**
**end for**
**return** $R_{B_i}$ ▷ Recovery vector of the image block

*Example 2.1* Consider a block of cover image $B_i$ of size $2 \times 2$. The block elements are clustered into two clusters $C_1$ and $C_2$ using CBG Algorithm.

$B_i$ = [29 29; 28 25] ⇒ $P_{B_i}$ = 25 ⇒ $Q_{B_i}$ = 29 ⇒ $U_{B_i}$ = 2 ⇒ $S_{B_i}$ = 27 ⇒ $C_1$ = 25, $C_2$ = [28, 29, 29]

**Algorithm 2** CBG

**INPUT:** Image Block ($B_i$) of size $2 \times 2$
**OUTPUT:** $C_1$ and $C_2$ composed of elements of $B_i$
**Ensure:**
(1)$min(B_i)$ returns minimum element of the block $B_i$
(2)$max(B_i)$ returns maximum element of the block $B_i$

```
∀B_i, where i = 1, 2, .....T_b                 ▷ For each 2 × 2 block of image
P_{B_i} ← min(B_i)                              ▷ Minimum element of block
Q_{B_i} ← max(B_i)                              ▷ Maximum element of block
U_{B_i} ← ⌈0.5(Q_{B_i} − P_{B_i})⌉               ▷ Semi Range of the block
S_{B_i} ← P_{B_i} + U_{B_i}                     ▷ Cluster Boundary for the block
for j ← 1 to 4 do                               ▷ For each element of the image block
    if P_{B_i} ≤ B_i^j < S_{B_i} then           ▷ Range of Cluster C_1
        C_1 ← B_i^j
    else
        C_2 ← B_i^j                             ▷ Range of Cluster C_2
    end if
end for
return C_1, C_2                                 ▷ Clusters C_1 and C_2
```

*Example 2.2* Recovery vector bits $R_{B_i}$ for image block $B_i$ is generated using RBG Algorithm.

$C_1$ = 25, $C_2$ = [28, 29, 29] ⇒ $l_1$ = 1, $l_2$ = 3 ⇒ $m_1$ = 25, $m_2$ = 29 ⇒ $T_1$ = 7, $T_2$ = 8 ⇒ $D$ = 1 ⇒

$R_{B_i}$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |

After generation of eight bit recovery information for each image block through Algorithms 1 and 2, this generated information must be embedded into mapping blocks such that it could handle worst tampering scenarios. To achieve this goal of increasing chances of accurate localization and recovery, the extracted recovery information of a block is permuted using secret key prior to embedding in the corresponding mapping blocks. Random mapping of recovery information enhances the robustness against cryptanalysis as well as enhances the security of the scheme. Based on a secret key($K_1$), a non convergent and non periodic logistic chaotic map, sensitive to the initial conditions is generated. The sequence is as follows:

$$z_{n+1} = \xi z_{n+1}(1 - z_n) \tag{5}$$

where $3.57 < \xi < 4$ and $0 < z_0 < 0.5$. The composition of secret key is as follows: $K_1$ =($\xi$, $z_0$). The generated chaotic sequence after binarization is sub divided into small series, each composed of eight bit binary information. The sub series is as follows: $y_i$=($y_{i_1}, y_{i_2}, y_{i_3}, y_{i_4}, y_{i_5}, y_{i_6}, y_{i_7}, y_{i_8}$), $i = 1, 2, ..., T_b$. The generated recovery information bits is encoded after operating in exclusive-or mode with the chaotic sub series to obtain the final sequence as follows: $W = W_1, W_2, ....W_{T_b}$.

$$w_{i,j} = R_{B_{ij}} \bigoplus y_{ij} \tag{6}$$

where $1 \leq i \leq T_b, 1 \leq j \leq 8$.

3) Embedding position generation. The embedding of the encoded block recovery information is mapped randomly to another block using a sequence generated based on a secret key $K_2$. A random sequence of length $T_b$, $r = (r_1, r_2, \ldots r_{T_b})$ is obtained using chaotic map in [20] as follows:

$$r_{n+1} = (1 + 0.3 \times \left(r_{n-1} - 1.08) + 379 \times r_n^2 + 1001 \times q_n^2\right) mod3 \tag{7}$$

Here, $q_n$ signifies the initial values $q_0, r_0, r_1$ of the logistic chaotic map [20]. The secret key $K_2 = (r_0, r_1, q_0)$ where $(r_0, r_1)\epsilon(-1.5, 1.5)$, $q_0\epsilon(0, 1)$. This random sequence $(r_1, r_2, ....r_{T_b})$ is sorted to obtain an ordered index sequence $(I_1, I_2, ...I_{N_b})$ used to select mapping block positions for embedding.

The recovery information is embedded in the frequency domain so as to increase its robustness against various tampers while transmission and also enhance its imperceptibility. First of all, the eight recovery bits are converted pairwise into their decimal equivalents to obtain a four valued resulting vector holding values within range of 0 to 3. The mapping has been depicted in Fig. 3 for mapping recovery bits to their decimal equivalents. Thereafter, discrete cosine transform is calculated for each of the image blocks and each DCT coefficient value of the block pixels is quantized to a new modified value depending upon the recovery bits using DCT based quantization method $(DCTQ)$ as Algorithm 4. Thereafter, inverse DCT is computed for each of the modified blocks to obtain the recovery embedded blocks to finally compose the watermarked image $(I_{rv})$. The detailed methodology of the recovery bit embedding $(RBE)$ is described in Algorithm 3.

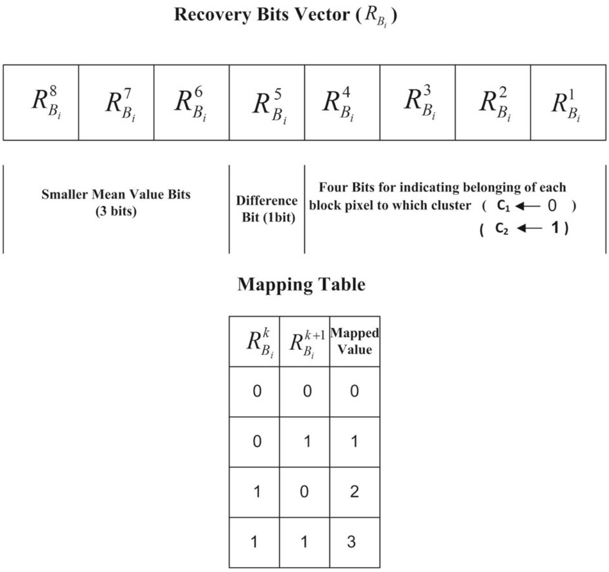



**Fig. 3** Mapping of Recovery Bits

**Algorithm 3** RBE

**INPUT:** Image Block ($B_i$) of size $2 \times 2$, Recovery Information Bits $R_{B_i}$
**OUTPUT:** Embedded Block $B_{e_i}$ of size $2 \times 2$
**Ensure:**
(1) $flag = 1$
(2) $D_j$ is the $j^{th}$ element of $D$ where $D$ is a $1 \times 4$ vector containing mapped values according to the mapping table.
(3)$e$ is a matrix of size $2 \times 2$.
(4)map returns mapped value according to the mapping table.

```
∀B_i,where i = 1, 2, ... T_b                          ▷ For each 2 × 2 block of image
H_i ← Discrete Cosine Transform of B_i                ▷ Compute DCT of the block
where j ← 1 to 4
while k ≤ 8 do
    D^j ← map[R^k_{B_i} R^{k+1}_{B_i}]                ▷ Map two consecutive bits of the Recovery Vector
end while
if H_i^j ≤ 0 then      ▷ Flag is initialized with negative sign if DCT coefficient is negative
    flag ← -1
end if
a ← ⌊|H_i^j/10|⌋        ▷ Integer portion of the DCT coefficient after dividing by 10
b ← a * 10              ▷ Lower bound of the DCT coefficient in multiples of 10
e^j ← DCTQ(D^j, b)
e^j ← e^j * flag                                      ▷ Modified DCT coefficient value
B_{e_i} ← Inverse Discrete Cosine Transform of e^j    ▷ Modified block
return B_{e_i}   ▷ Recovery embedded watermarked image(I_rv) is composed of these B_{e_i}
```

*Example 2.3* Consider a block of cover image($B_i$) of size $2 \times 2$. The recovery bits $R_{B_i}$ = [11110111] is embedded using RBE and DCTQ Algorithm into the block $B_i$ to obtain recovery information embedded block $B_{e_i}$.

For $H_i^j$=57, and $D^j$=3, flag=1

$B_i$

| | |
|---|---|
| 27 | 27 |
| 22 | 38 |

⇒ $H_i$

| | |
|---|---|
| 57 | -8 |
| -3 | 8 |

| $a$ | | $b$ | | $g$ | | $h$ | | $V_m$ | $e^j$ |
|---|---|---|---|---|---|---|---|---|---|
| 5 | ⇒ | 50 | ⇒ | 60 | ⇒ | 57 | ⇒ | 58.5 | 58.5 |

For $H_i^j$=-8, and $D^j$=3, flag=-1

| $a$ | | $b$ | | $g$ | | $h$ | | $V_m$ | $e^j$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ⇒ | 0 | ⇒ | 10 | ⇒ | 7 | ⇒ | 8.5 | -8.5 |

For $H_i^j$=-3, and $D^j$=1, flag=-1

| $a$ | | $b$ | | $g$ | | $h$ | | $V_m$ | $e^j$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ⇒ | 0 | ⇒ | 4 | ⇒ | 2 | ⇒ | 3 | -3 |

For $H_i^j$=8, and $D^j$=3, flag=1

| $a$ | | $b$ | | $g$ | | $h$ | | $V_m$ | $e^j$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ⇒ | 0 | ⇒ | 10 | ⇒ | 7 | ⇒ | 8.5 | 8.5 |

$e$

| | |
|---|---|
| 58.5 | -8.5 |
| -3 | 8.5 |

⇒ $B_{e_i}$

| | |
|---|---|
| 27.75 | 27.75 |
| 22.25 | 39.25 |

**Algorithm 4** DCTQ

**INPUT:** $S$ is one of the elements of $D$ and $W$ is the lower bound of absolute DCT coefficient value of the Image Block ($B_i$) elements.
**OUTPUT:** $V_m$ as modified absolute DCT coefficient value of the image block elements.

**if** $S \leftarrow 0$ **then** ▷ Mapped consecutive recovery bits as 0
  $g \leftarrow W + 2$ ▷ Upper bound of Modified Coefficient
  $h \leftarrow W$ ▷ Lower bound of Modified Coefficient
  $V_m \leftarrow 0.5 * (g + h)$ ▷ Modified coefficient value
**end if**
**if** $S \leftarrow 1$ **then** ▷ Mapped consecutive recovery bits as 1
  $g \leftarrow W + 4$ ▷ Upper bound of Modified Coefficient
  $h \leftarrow W + 2$ ▷ Lower bound of Modified Coefficient
  $V_m \leftarrow 0.5 * (g + h)$ ▷ Modified coefficient value
**end if**
**if** $S \leftarrow 2$ **then** ▷ Mapped consecutive recovery bits as 2
  $g \leftarrow W + 7$ ▷ Upper bound of Modified Coefficient
  $h \leftarrow W + 4$ ▷ Lower bound of Modified Coefficient
  $V_m \leftarrow 0.5 * (g + h)$ ▷ Modified coefficient value
**end if**
**if** $S \leftarrow 3$ **then** ▷ Mapped consecutive recovery bits as 3
  $g \leftarrow W + 10$ ▷ Upper bound of Modified Coefficient
  $h \leftarrow W + 7$ ▷ Lower bound of Modified Coefficient
  $V_m \leftarrow 0.5 * (g + h)$ ▷ Modified coefficient value
**end if**
**return** $V_m$ ▷ Modified coefficient value

## 2.2 Embedding of copyright information

After obtaining the recovery information embedded cover image ($I_{rv}$), next comes the task of embedding the copyright information. Copyright logos ($C_L$) are in the form of binary watermarks containing total pixels as $\frac{M \times N}{4}$. The copyright logo is traversed in a sequential row by row manner to get an equivalent bit sequence $W_L$ of the logo pixel values. Thereafter, corresponding to each bit of the sequence, a $2 \times 2$ block is generated according to the copyright embedding algorithm ($CPE$) as Algorithm 5 to obtain the copyright encoded image $I_{cp}$. The detailed algorithm is as follows:

*Example 2.4* Consider a recovery information embedded block of cover image($B_{emb}$) of size $2 \times 2$. The extraction of recovery bits is done using the RIG Algorithm from it.

For $F_{dct}^i$=58.5,
For $F_{dct}^i$=-8.5,
For $F_{dct}^i$=-3,
For $F_{dct}^i$=8.5,

$B_{emb}$

| | |
|---|---|
| 27.75 | 27.75 |
| 22.25 | 39.25 |

⇒ $F_{dct}$

| | |
|---|---|
| 58.5 | -8.5 |
| -3 | 8.5 |

| $L_b$ | | $U_b$ | | $bits$ |
|---|---|---|---|---|
| 50 | ⇒ | 60 | ⇒ | 11 |
| 0 | ⇒ | 10 | ⇒ | 11 |
| 0 | ⇒ | 10 | ⇒ | 01 |

Hence, recovery bits vector is obtained as follows:

---

**Algorithm 5** CPE

---

**INPUT:** $W_L$ as watermark bit sequence of the copyright binary logo($C_L$) of length $\frac{M*N}{4}$.
**OUTPUT:** $I_{cp}$ is the encoded image with copyright information of size $M \times N$.
**Ensure:**
 (1) $W_{L_i}$ is the $i^{th}$ bit of $W_L$
 (2) $R_i$ is an empty matrix of size $2 \times 2$

**for** $i \leftarrow 1$ to $\frac{M \times N}{4}$ **do** ▷ For all bits of the copyright logo
  **if** $W_{L_i} \leftarrow 0$ **then** ▷ When bit of copyright logo is 0
    $R_i \leftarrow \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$
  **else** ▷ When bit of copyright logo is 1
    $R_i \leftarrow \begin{bmatrix} 3 & 2 \\ 2 & 2 \end{bmatrix}$
  **end if**
**end for**
**return** $I_{cp}$ ▷ $I_{cp}$ is composed of non overlapping blocks $R_i$

---

## 2.3 Generation and embedding of Authentication information

The authentication of the watermarked image is done at pixel level. An authentication bit is generated for each of the pixel depending upon its intensity value ($P_{xy}$) and position co-ordinates (row($x$), column($y$)) as shown in Fig. 4. The detailed algorithm for pixel level authentication (PLA) is enlisted in the Algorithm 6.

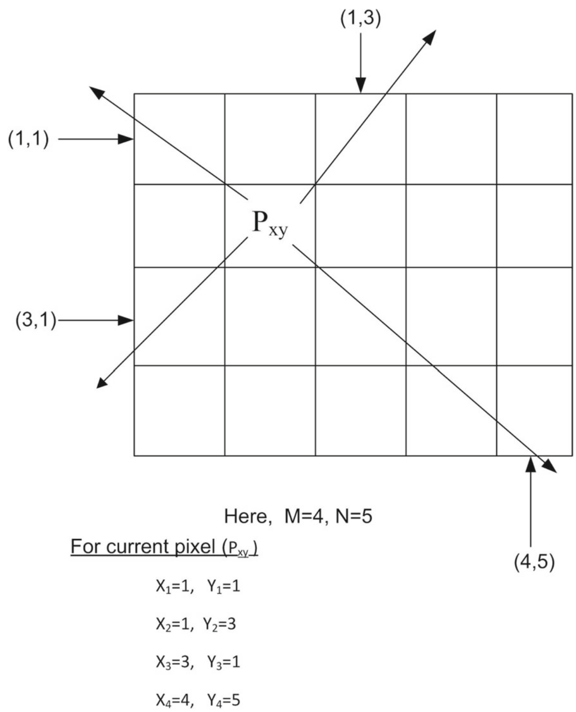



**Fig. 4** Authentication Bit for a Pixel

**Algorithm 6** PLA

**INPUT:** $P_{xy}$ is the pixel intensity at $x, y$ location, $x, y$ as the pixel co-ordinates,$M, N$ as size of cover image.
**OUTPUT:** $\psi$ is the authentication matrix of size $M \times N$.
**Ensure:**
$rr$ is a vector of size $1 \times 4$.
$\alpha$ is a empty matrix of size $4 \times 8$.
$A_s^i$ is composed of $A_s^1, A_s^2, A_s^3, A_s^4$ which are vectors of size $1 \times 4$
$\beta$ is a vector of size $1 \times 4$
$\psi$ is the authentication matrix of size $M \times N$
$\psi_{xy}$ denotes the authentication bit at x,y co-ordinate position

$x_1 \leftarrow x$ ▷ Assign current x co-ordinate
$y_1 \leftarrow y$ ▷ Assign current y co-ordinate
**while** $x_1 \neq 1$ *or* $y_1 \neq 1$ **do** ▷ Keep traversing until $x_1$ or $y_1$ co-ordinate becomes 1
  $x_1 \leftarrow x_1 - 1$ ▷ Decrement x co-ordinate by 1
  $y_1 \leftarrow y_1 - 1$ ▷ Decrement y co-ordinate by 1
**end while**
**if** $x_1 \neq 1$ **then** ▷ $rr_1$ is assigned co-ordinate value other than 1
  $rr(1) \leftarrow x_1$
**else**
  $rr(1) \leftarrow y_1$
**end if**
$x_2 \leftarrow x$ ▷ Assign current x co-ordinate
$y_2 \leftarrow y$ ▷ Assign current y co-ordinate
**while** $x_2 \neq 1$ *or* $y_2 \neq N$ **do** ▷ Keep traversing until $x_2$ is 1 or $y_2$ co-ordinate becomes $N$
  $x_2 \leftarrow x_2 - 1$ ▷ Decrement x co-ordinate by 1
  $y_2 \leftarrow y_2 + 1$ ▷ Increment y co-ordinate by 1
**end while**
**if** $x_2 \neq 1$ **then** ▷ $rr_2$ is assigned co-ordinate value other than 1 or $N$
  $rr(2) \leftarrow x_2$
**else**
  $rr(2) \leftarrow y_2$
**end if**
$x_3 \leftarrow x$ ▷ Assign current x co-ordinate
$y_3 \leftarrow y$ ▷ Assign current y co-ordinate
**while** $x_3 \neq M$ *or* $y_3 \neq 1$ **do** ▷ Keep traversing until $x_3$ is $M$ or $y_3$ co-ordinate becomes 1
  $x_3 \leftarrow x_3 + 1$ ▷ Increment x co-ordinate by 1
  $y_3 \leftarrow y_3 - 1$ ▷ Decrement x co-ordinate by 1
**end while**
**if** $x_3 \neq M$ **then** ▷ $rr_3$ is assigned co-ordinate value other than $M$ or 1
  $rr(3) \leftarrow x_3$
**else**
  $rr(3) \leftarrow y_3$
**end if**
$x_4 \leftarrow x$ ▷ Assign current x co-ordinate
$y_4 \leftarrow y$ ▷ Assign current y co-ordinate
**while** $x_4 \neq M$ *or* $y_4 \neq N$ **do** ▷ Keep traversing until $x_4$ is $M$ or $y_4$ co-ordinate becomes $N$
  $x_4 \leftarrow x_4 + 1$ ▷ Increment x co-ordinate by 1
  $y_4 \leftarrow y_4 + 1$ ▷ Decrement y co-ordinate by 1
**end while**
**if** $x_4 \neq M$ **then** ▷ $rr_4$ is assigned co-ordinate value other than $M$ or $N$
  $rr(4) \leftarrow x_4$
**else**
  $rr(4) \leftarrow y_4$
**end if**
**for** $i \leftarrow 1$ to 4 **do** ▷ Binary equivalents of the binding neighbors
  $\alpha(i) \leftarrow \lfloor rr(i)/2^u \rfloor mod2$
  where $u \leftarrow 0$ to 4
**end for** ▷ $\alpha(i)$ represents $i^{th}$ row of $\alpha$ with 8 columns
**for** $j \leftarrow 1$ to 4 **do** ▷ Bitwise Exor of the Pixel intensity value with the binding neighbors
  $A_s^j \leftarrow Exor(\alpha_{4-u}^j, P_{xy_u})$
  where $u \leftarrow 4$ to 0
**end for**
$\beta \leftarrow (A_s^1 \bigwedge A_s^2 \bigwedge A_s^3 \bigwedge A_s^4)$ ▷ $\beta$ represents result of bitwise logical AND operation of all $A_s^j$
$\psi_{xy} \leftarrow (\sum_{j=1}^{4} \beta^j) mod2$ ▷ One bit authentication bit for the current pixel
**return** $\psi$ ▷ Authentication matrix of size $M \times N$

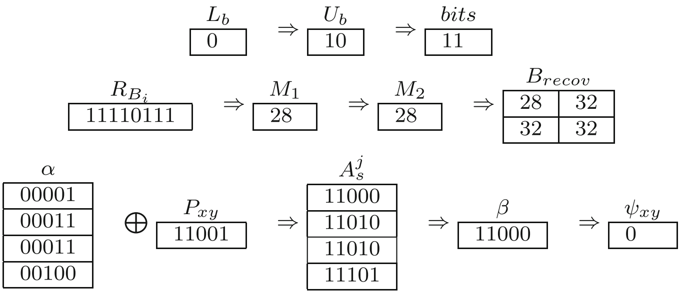
*Example 2.5* Generation of authentication bit for Pixel $P_{xy}$ using PLA Algorithm.

After obtaining the recovery information embedded image ($I_{rv}$), copyright information encoded image ($I_{cp}$) and the authentication matrix ($\psi$), all the three are coupled to obtain the final watermarked image ($I_w$) using the recovery copyright authentication ($RCA$) algorithm as Algorithm 7. The detailed procedure will follow.

**Algorithm 7** RCA

**INPUT:** $I_{rv}$, $I_{cp}$, $\psi$ are recovery embedded image, copyright information encoded image and authentication matrix of size $M \times N$.
**OUTPUT:** $I_w$ is the watermarked image of size $M \times N$.
**Ensure:**
(1) $I_w$ is eight bit watermarked image of size $M \times N$.
(2) $B_1$ is a vector of size $1 \times 5$.
(3) $B_2$ is a vector of size $1 \times 2$.

**for** $i \leftarrow 1$ to $M$ **do**
  **for** $j \leftarrow 1$ to $N$ **do**
    $B_1 \leftarrow \lfloor I_{rv}(i,j)/2^u \rfloor mod 2$
    where $u \leftarrow 0$ to $4$
    $B_2 \leftarrow \lfloor I_{cp}(i,j)/2^v \rfloor mod 2$
    where $v \leftarrow 0$ to $1$
    $I_w(i,j)^p \leftarrow B_1$
    where $p \leftarrow 8$ to $4$ ▷ Embedding into first five MSB's
    $I_w(i,j)^q \leftarrow B_2$
    where $q \leftarrow 3$ to $2$ ▷ Embedding into $3^{rd}$ and $2^{nd}$ LSB's
    $I_w(i,j)^r \leftarrow \psi(i,j)$
    where $r \leftarrow 1$ ▷ Embedding into $1^{st}$ LSB's
  **end for**
**end for**
**return** $I_w$ ▷ Watermarked Image




## 2.4 Ownership verification

The watermarked image ($I_w$) is transmitted to the receiver end. To proceed with the ownership assertion, the copyright information is extracted from the received watermarked image to obtain the copyright logo, called as extracted watermark logo ($W_{ext}$) using copyright extraction (CEX) algorithm as Algorithm 8. The rightful owner possesses the original watermark logo. If the extracted logo matches with the original one, he is proved to be the legitimate owner of the cover image. If there is a dispute, then he is the unauthorized owner and there may be a possibility of tampering of the content. To actually detect the tampered areas, one has to proceed with the $TMD$ (Tamper Detection) algorithm as Algorithm 9.

**Algorithm 8** CEX

**INPUT:** $I_w$
**OUTPUT:** $W_{ext}$ is the extracted watermark logo of size $\frac{M \times N}{4}$.
**Ensure:**
(1) $B_w$ is $2 \times 2$ sized non-overlapping block of watermarked image $I_w$
(2) $W_{bsq}$ is vector of size $1 \times \frac{M \times N}{4}$
(3)$B_{dct}$ is vector of size $1 \times 4$
(4) $B^i_{dct}$ denotes the DCT value of the $i^{th}$ pixel of the $2 \times 2$ non-overlapping block $B_w$.

```
1: ∀B_w, where w = 1, 2, ... T_b            ▷ For all blocks of watermarked image
2: B_dct ← Discrete Cosine Transform of B_w   ▷ Compute DCT of each block
3: S ← Σ_{i=1}^{4} B^i_dct                  ▷ Sum of DCT coefficients of each block
4: if S = 2 or S = 6 then
5:     bit_ext ← 1                          ▷ Append the bit_ext bit to vector W_bsq
6: else
7:     bit_ext ← 0                          ▷ Append the bit_ext bit to vector W_bsq
8: end if  ▷ W_ext is the extracted watermark by reshaping vector W_bsq to image of size (M×N)/4
9: return W_ext                             ▷ Extracted Watermark
```

## 2.5 Tamper detection

The proposed scheme performs pixel level authentication for detecting the tampered regions of the suspected watermarked image. To chalk out the tampered regions, first of all the authentication bit is calculated for each pixel using the PLA algorithm. Also, the authentication information is extracted from the suspected watermarked image received. If there is a match between the extracted authentication bit and the calculated one, then it indicates the untampered pixel signified by black region. Otherwise, it belongs to the tampered region indicated by white regions in the tamper detected image ($I_{Det}$) respectively.

**Algorithm 9** TMD

**INPUT:** $I_w$
**OUTPUT:** $I_{Det}$ is the tamper detected image of size $M \times N$.
**Ensure:**
(1)$I_{Cal}$ is the empty matrix of size $M \times N$.
(2)$I_{Temp}$ is the empty matrix of size $M \times N$.
(3)Extract is a function to extract the authentication bit from the $u^{th}$ LSB position.
(4)Compare is a function to compare pixelwise values.

```
1: I_Cal ← Call PLA(I_w) ▷ Calculate Authentication Matrix for watermarked image using PLA Algorithm
2: for i ← 1 to M do
3:     for j ← 1 to N do
4:         I_Temp(i, j) ← Extract^u(I_w(i, j))   ▷ Extract the authentication bit from each pixel of the watermarked image
5:         where u ← 1
6:     end for
7: end for
8: I_Det ← Compare(I_Cal, I_Temp)  ▷ Compare the calculated authentication matrix with the extracted one ▷ Matching pixel is treated as true(0) ▷ Non Matching pixel is treated as false(1)
9: return I_Det        ▷ Tamper Detected Image with black and white regions indicating unaltered and altered portions of image
```

## 2.6 Recovery of tampered image

After the detection of tampered areas(tampered blocks) by using the pixel level authentication (PLA) algorithm, the recovery is to be done by mapping them to their corresponding recovery information embedded blocks. Thereafter, two bits recovery information is extracted from each of the DCT block coefficients to finally build up the eight bits recovery information of the block. From the retrieved recovery information, the block elements are build up to form the recovered block. The detailed algorithm of Recovery image generation (RIG) has been detailed in Algorithm 10.

**Algorithm 10** Algorithm for RIG

**INPUT:** $I_w$,$I_{Det}$
**OUTPUT:** $I_R$ is the recovered image of size $M \times N$.
**Ensure:**
(1)Block is a function to find tampered blocks corresponding to tampered pixel locations.
(2)MapBlock is a function to map the tampered blocks with its corresponding recovery information embedded blocks.
(3)$F_{dct}$,$B_{recov}$ are empty matrices of size $2 \times 2$.
(4)$bits$ is a vector of size $1 \times 8$ that will be reinitialized for each block contained in the set of blocks $B_{emb}$.

$B_{Temp} \leftarrow Call\ Block(I_{Det})$ ▷ $B_{Temp}$ contains all tampered blocks of $I_w$
$B_{emb} \leftarrow Call\ MapBlock(B_{Temp})$ ▷ $B_{emb}$ contains all recovery information embedded blocks of $B_{Temp}$
$\forall B_{emb}$ ▷ For each recovery information embedded block
$F_{dct} \leftarrow$ Absolute Value of Discrete Cosine Transform of $B_{emb}$
**for** $i \leftarrow 1$ to 4 **do**
  $L_b \leftarrow 10 * (\lfloor F^i_{dct}/10 \rfloor)$ ▷ Lower bound of each absolute DCT coefficient of block (in multiples of 10)
  $U_b \leftarrow L_b + 10$ ▷ Upper bound of each absolute DCT coefficient of block (in multiples of 10)
  **if** $L_b \leq F^i_{dct} < L_b + 2$ **then** ▷ if DCT coefficient lies in range of lower bound and lower bound incremented by two
    $bits \leftarrow [0\ 0]$
  **end if**
  **if** $L_b + 2 \leq F^i_{dct} < L_b + 4$ **then** ▷ if DCT coefficient lies in range of lower bound incremented by two and four
    $bits \leftarrow [0\ 1]$
  **end if**
  **if** $L_b + 4 \leq F^i_{dct} < L_b + 7$ **then** ▷ if DCT coefficient lies in range of lower bound incremented by four and seven
    $bits \leftarrow [1\ 0]$
  **end if**
  **if** $L_b + 7 \leq F^i_{dct} \leq U_b$ **then** ▷ if DCT coefficient lies in range of lower bound incremented by seven and upper bound
    $bits \leftarrow [1\ 1]$
  **end if** ▷ Append the bits variable for each of the block values
**end for**
$M \leftarrow Extract^u(bits)$ ▷ Extract bits from eighth to sixth positions of $bits$ vector
where $u \leftarrow 8$ to 6
$M_D \leftarrow (M, v)2^v$ ▷ Form the decimal equivalent mean value
where $v \leftarrow 0$ to 2
$D_c \leftarrow Extract^u(bits)$ ▷ Extract bit from fifth position of $bits$ vector
where $u \leftarrow 5$
$M_2 \leftarrow (M_D + D_c) * 4$ ▷ Obtain the larger mean value as $M_2$
$M_1 \leftarrow M_D * 4$ ▷ Obtain the smaller mean value as $M_1$
**for** $i \leftarrow 1$ to 4 **do** ▷ Depending on bits in positions one to four in $bits$ vector form recovered block pixels
  **if** $Extract^u(bits) \leftarrow 1$ **then** ▷ If bit value is 1
    $B_{recov}(i) \leftarrow M_2$ ▷ Assign $M_2$ as recovered pixel value
  **else**
    $B_{recov}(i) \leftarrow M_1$ ▷ Assign $M_1$ as recovered pixel value
  **end if**
**end for**
**return** $I_R$ ▷ Recovered image is composed of $B_{recov}$

## 3 Experimental results and analysis

The proposed scheme has been simulated on a wide set of standard grayscale images using MATLAB 2013Ra. Variations of grayscale images has been tested upon, majorly categorized into three main kinds i.e. natural images, satellite images and texture images. Some of these grayscale images sized 512 × 512 has been shown in Fig. 5. To quantitatively evaluate the imperceptibility of the watermarked images, peak signal to noise ratio (PSNR) metric have been adopted with values enlisted in Table 3 and visual quality representation depicted in Fig. 6 respectively.

The Peak-Signal-to-Noise-Ratio (PSNR) metric is defined as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} (dB) \tag{8}$$

$$MSE = \frac{1}{M1 \times M2} \sum_{1}^{M1} \sum_{1}^{M2} ||C'_{i,j} - C_{i,j}|| \tag{9}$$

**Fig. 5** Cover Test Images

**Table 3** PSNR Values for Different Types of Watermarked Images

| Classification | Image | PSNR(dB) |
|---|---|---|
| Natural Images | Lena | 29.13 |
| | Barbara | 28.11 |
| | Cameraman | 28.45 |
| | Baboon | 29.20 |
| | Airplane | 30.01 |
| | Pepper | 30.11 |
| Remote Sensing Images | Satellite Image 1 | 29.63 |
| | Satellite Image 2 | 29.23 |
| | Satellite Image 3 | 30.11 |
| Texture Images | Bark | 29.10 |
| | Plastic Bubbles | 30.11 |
| | Brick Wall | 29.34 |

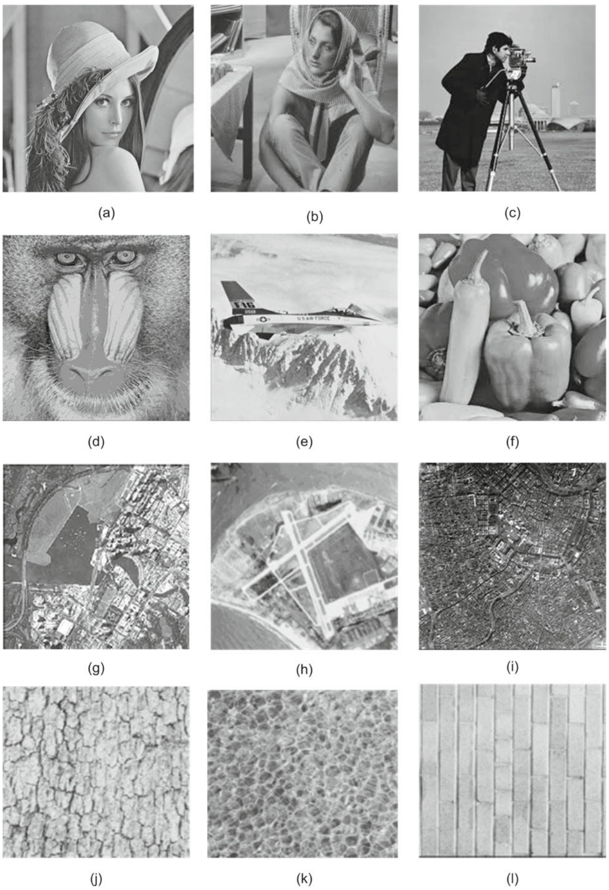



**Fig. 6** Watermarked Test Images

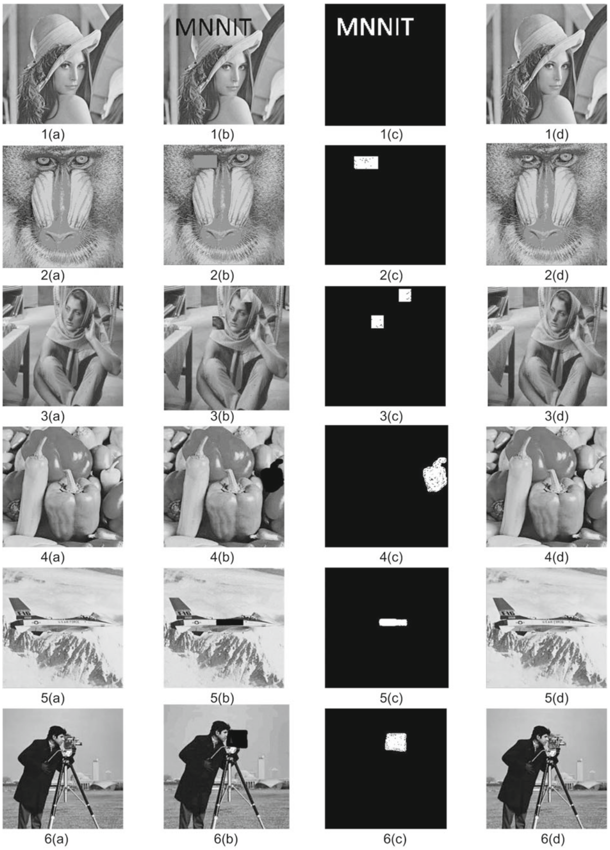



**Fig. 7** Examples of Natural Images (**a**) Watermarked Images (**b**) Tampered Images (**c**) Tamper Detected Images (**d**) Recovered images

where, $C_{i,j}$ and $C'_{i,j}$ represents pixel value of original cover image and the watermarked image of size $M1 \times M2$.

PSNR values for the three categories of grayscale images has been tabulated in Table 3. Indistinguishability and imperceptibility attained for the above three categories of watermarked images is satisfyingly enough as indicated by the PSNR values. The dual functionalities of the proposed scheme have been evaluated using various available metrics and discussed in separate sections as follows:

### 3.1 Tamper detection and recovery

To evaluate the tamper detection and recovery efficiency of the proposed scheme, following metrics have been adopted:

1. Tampering Ratio(TR)

$$r_t = \frac{100 N_T}{N} \% \tag{10}$$

where, $N$ and $N_T$ denotes the total number of blocks and the number of tampered blocks in the test cover image.

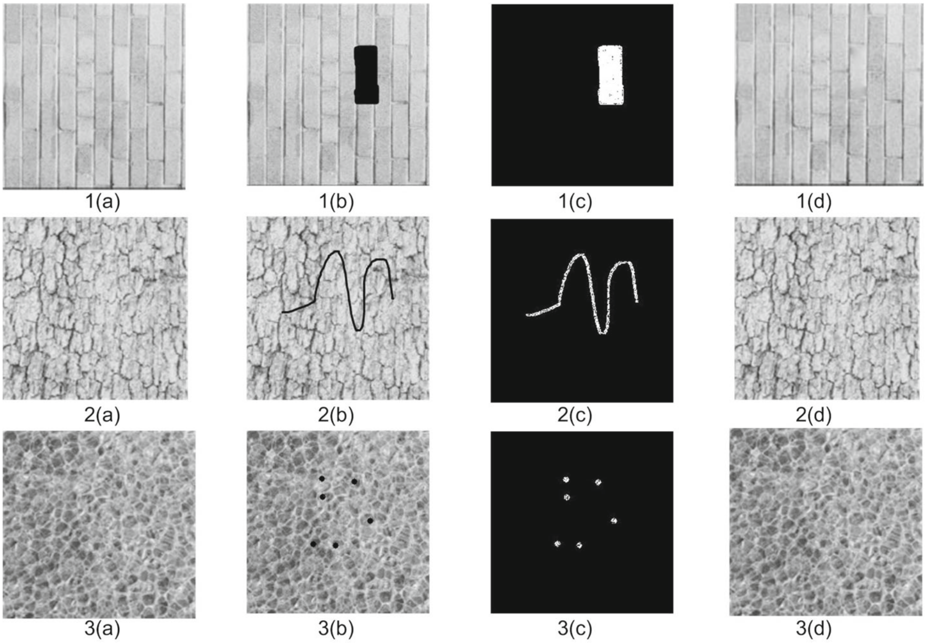



**Fig. 8** Examples of Texture Images (**a**) Watermarked Images (**b**) Tamper Images (**c**) Tampered Detected Images (**d**) Recovered images

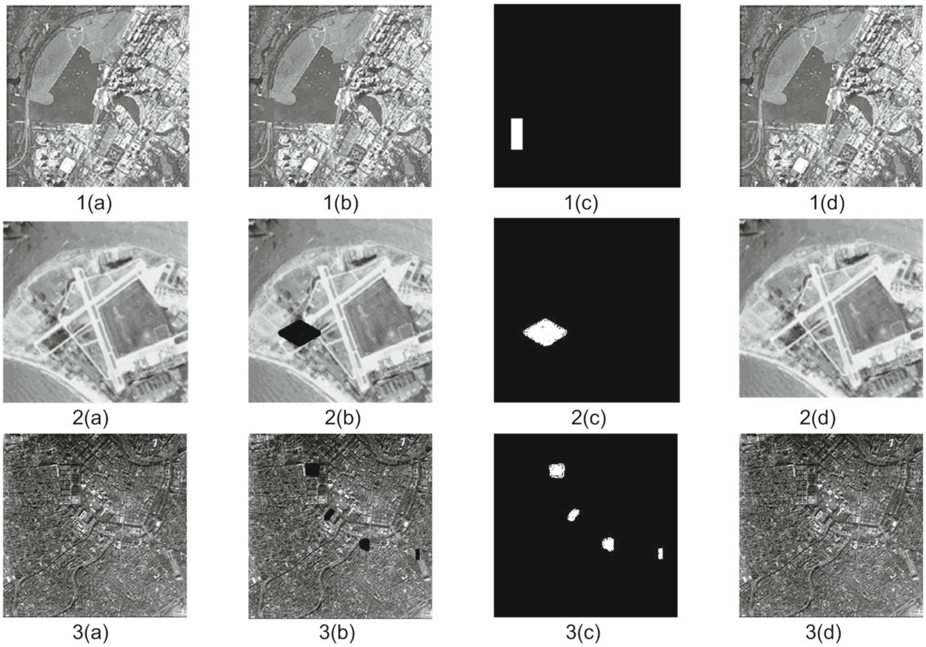



**Fig. 9** Examples of Satellite Images (**a**) Watermarked Images (**b**) Tamper Images (**c**) Tampered Detected Images (**d**) Recovered images

**Table 4** Results of Tamper Detection and Recovery for Natural Images

| Cover Image | No. of Detected Pixels | Total Pixels Altered | Detection Rate |
|---|---|---|---|
| Lena | 1988 | 2029 | 97 |
| Baboon | 812 | 823 | 98.6 |
| Barbara | 860 | 897 | 95.8 |
| Pepper | 1863 | 1902 | 97.94 |
| Airplane | 544 | 599 | 90.8 |
| Cameraman | 974 | 1016 | 95.86 |

2. Probability of False Rejection(PFR)

$$P_{fr} = \frac{100N_{ud}}{(N - N_T)}\% \tag{11}$$

where, $N_{ud}$, $N$ and $N_T$ denotes the number of valid blocks that are wrongly detected, the total number of blocks and the number of tampered blocks in the test cover image.

3. Probability of False Acceptance(PFA)

$$P_{fa} = \frac{100(N_T - N_{td})}{N_T}\% \tag{12}$$

where, $N_{td}$, $N$ and $N_T$ denotes the number of tampered blocks that are correctly detected,the total number of blocks and the number of tampered blocks in the test cover image.

To validate the efficiency of tamper detection and accurate localization of the proposed scheme for aforementioned three major categories of images: natural, texture and satellite, different attacks have been tested with few depicted in Fig. 7 for natural images, in Fig. 8 for texture images and in Fig. 9 for satellite images respectively. The detection of the altered regions have been done using Algorithm 9 and reflected by white regions in the tamper detected image($I_{Det}$) whereas untampered ones signified by black regions. The level of accuracy is quite good. In the Figs. 7, 8 and 9, column representation is as follows: (a) the original image (b)the tampered image (c)the tamper detected image and (d)the recovered image. Different kind of attacks have been applied on the watermarked images like addition of text to the image, cropping some portion of the image, exchanging different image portions, removing some detailed sensitive information of the image etc. The tamper detection results along with recovery on pixel basis are tabulated in Tables 4, 5 and 6 for natural images, texture images and remote sensing images respectively.

**Table 5** Results of Tamper Detection and Recovery for Texture Images

| Cover Image | No. of Detected Pixels | Total Pixels Altered | Detection Rate |
|---|---|---|---|
| Brick Wall | 2073 | 2115 | 98.01 |
| Bark | 1273 | 1350 | 94.64 |
| Plastic Bubbles | 902 | 226 | 92.47 |

**Table 6** Results of Tamper Detection and Recovery for Remote Sensing Images

| Cover Image | No. of Detected Pixels | Total Pixels Altered | Detection Rate |
|---|---|---|---|
| Satellite Image 1 | 550 | 560 | 98.21 |
| Satellite Image 2 | 924 | 978 | 94.47 |
| Satellite Image 3 | 700 | 753 | 92.96 |

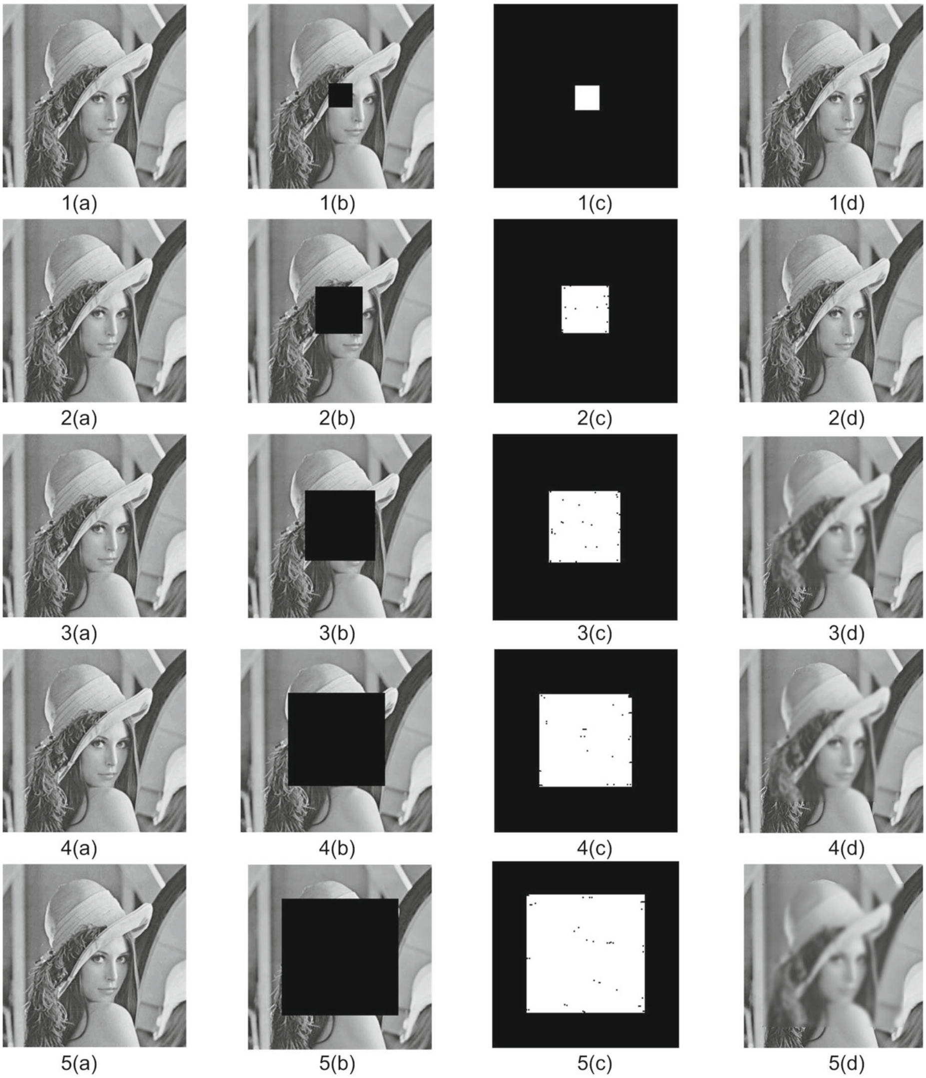



**Fig. 10** Recovery for different Tampering Ratios(TR) (**a**)10 % (**b**) 20 % (**c**)30 % (**d**)40 % (**d**)50 %

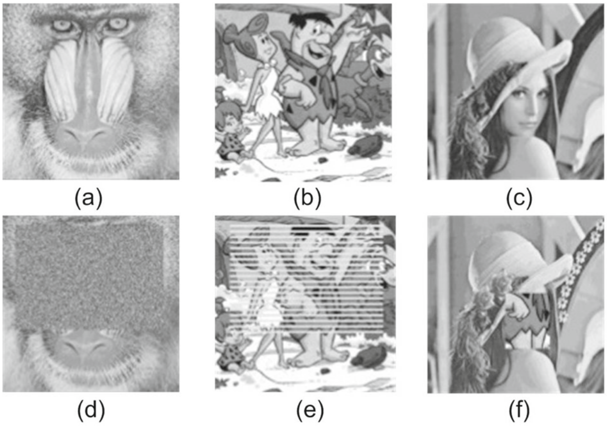



**Fig. 11** (**a**)–(**c**)Watermarked Image,(**d**)–(**f**)Tampered Image

To further demonstrate the efficacy of the proposed scheme for recovery, the cover image is tampered with varying tampering ratios ($TR$) as shown in Fig. 10 for one of the test cover lena images. The scheme is able to recover the lost content even when major portions of the image are lost although the visual quality deteriorates with increasing tampering ratio.

To illustrate the efficacy of the proposed scheme over other state of the art algorithms, following tests were performed as depicted in Fig. 11. The details of the tests are as follows:

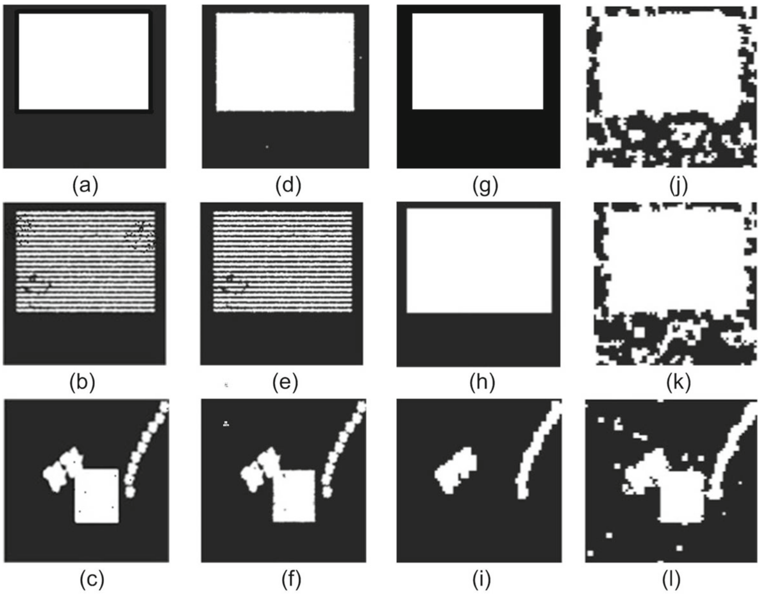



**Fig. 12** Tamper Detection Results:(**a**)–(**c**)Proposed method, (**d**)–(**f**)[11], (**g**)–(**i**) [43],(**j**)–(**l**)[21]

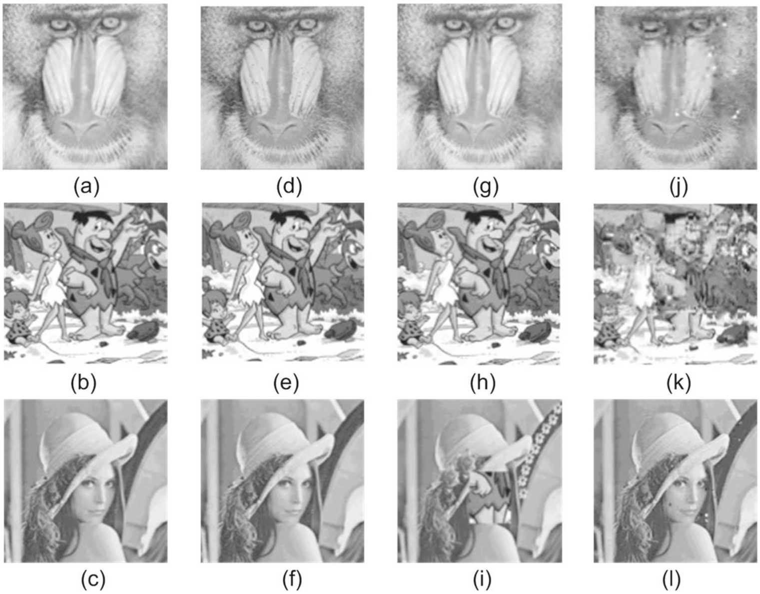



**Fig. 13** Recovery Results:(**a**)–(**c**)Proposed method, (**d**)–(**f**)[11], (**g**)–(**i**) [43],(**j**)–(**l**)[21]

**Table 7** Comparative Performance based on PFA

| Tests | Tampering Ratio | our | [30] | [11] | [43] | [21] | [8] | [40] | [27] | [35] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 48.07 | 0 | 0.3 | 0.01 | 0.00 | 0.93 | 0 | 0.1 | 0.28 | 0.05 |
| 2 | 26.48 | 0 | 0.05 | 1.50 | 0.00 | 2.34 | 0 | 0.11 | 0.34 | 0.07 |
| 3 | 13.58 | 0 | 0.01 | 1.31 | 57.56 | 1.48 | 0.01 | 0.09 | 0.36 | 0.08 |

**Table 8** Comparative Performance based on PFR

| Tests | Tampering Ratio | our | [30] | [11] | [43] | [21] | [8] | [40] | [27] | [35] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 48.07 | 0.01 | 0 | 0.36 | 4.62 | 32.94 | 1.0 | 0.01 | 0.23 | 0 |
| 2 | 26.48 | 0.01 | 0.01 | 3.14 | 37.03 | 53.95 | 1.0 | 0.001 | 0.20 | 0 |
| 3 | 13.58 | 0.02 | 0.03 | 0.16 | 2.20 | 4.89 | 0.85 | 0 | 0.15 | 0 |

**Table 9** Comparative Performance based on PSNR

| Tests | Tampering Ratio | our | [30] | [11] | [43] | [21] | [8] | [40] | [27] | [35] |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 48.07 | 16.01 | 18.2 | 24.79 | 23.95 | 20.74 | 18 | 17.5 | 12 | 12 |
| 2 | 26.48 | 21.08 | 20 | 28.72 | 21.78 | 16.13 | 19.5 | 19.0 | 13.7 | 15.9 |
| 3 | 13.58 | 22.02 | 20.2 | 36.81 | 18.31 | 32.42 | 21.5 | 20.0 | 16 | 18.7 |

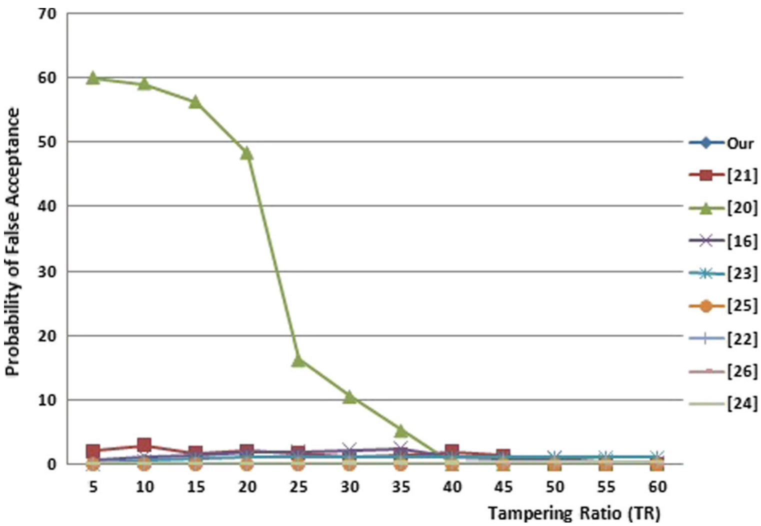



**Fig. 14** Variation of Probability of False Acceptance(PFA) with Tampering Ratio(TR)

$Test1$: A rectangular portion($300 \times 420$) of the watermarked baboon image is tampered [11(d)].

$Test2$: Watermarked Flinstones image is tampered by drawing 20 rectangles, filled with a random integer $\in [200, 223]$[11(e)].

$Test3$: The watermarked Lena image is tampered by pasting portion of the watermarked Flinstones image on it(collage attack) besides placing few small flowers and two large ones on it [11(f)].

The tamper detection efficacy of the proposed scheme is found to be quite good as pixel level authentication is done to chalk out the tampered pixels. Authentication bit for

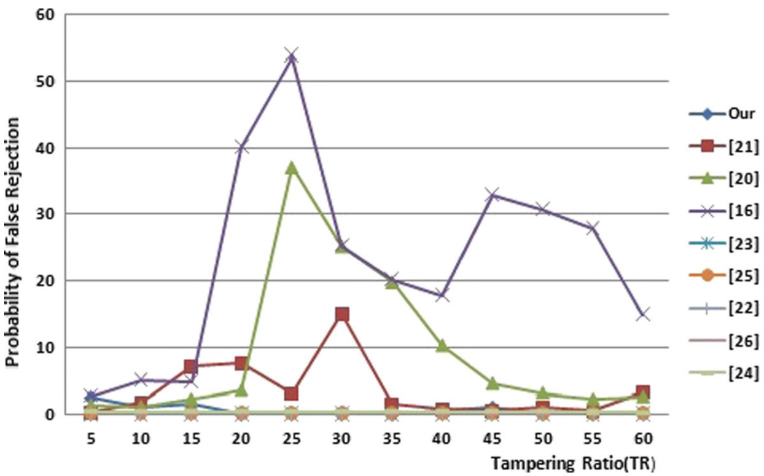



**Fig. 15** Variation of Probability of False Rejection(PFR) with Tampering Ratio(TR)

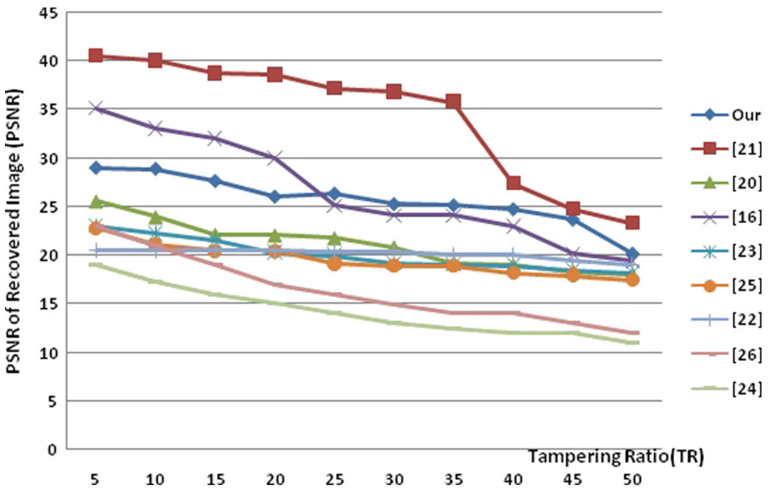



**Fig. 16** Variation of PSNR of Recovered Image with Tampering Ratio(TR)

each pixel is generated based on pixel intensity value, its location co-ordinates and boundary intersecting neighbors as shown in Fig. 4. The recovery of the altered regions is done via extracting the recovery information bits from the corresponding mapped block and rebuilding the lost content from it. The comparative results for accuracy of localization and recovery are presented in Figs. 12 and 13. Schemes [21] and [11] were able to localize the collage attack in test 3 approximately. However, [43] was not even able to detect the collaged blocks.

The probability of false acceptance (PFA), probability of false rejection (PFR) and PSNR metric values are evaluated for variable degree of alterations on the cover test images. Comparative results are tabulated in Tables 7, 8 and 9 and presented graphically in Figs. 14, 15 and 16. The PFR and PFA values of the proposed scheme are quite close to the ideal value zero in most of the analytical analysis. The imperceptibility of the recovered image decreases with the increasing tampering ratios.

### 3.2 Robustness test results

The proposed scheme provides the feature of copyright protection too. Different binary logos have been used as test watermarks. Some are shown in Fig. 17. The robustness of the scheme is tested against comprehensive set of image processing attacks. Some are enlisted

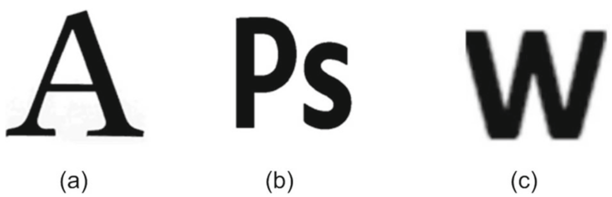
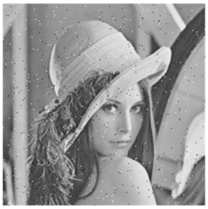



**Fig. 17** Copyright Logos

**Table 10** Robustness test results

| Attacked Watermarked image | Attack Types with parameters | Extracted Watermark | NCC value |
|---|---|---|---|
|  | Salt & Pepper Noise (var=0.05) |  | 0.98 |
|  | Histogram Equalization |  | 0.96 |
|  | Rotation(60) |  | 0.89 |
|  | Speckle Noise (var=0.05) |  | 0.91 |
|  | Motion Blur |  | 0.83 |

in Tables 10 and 11. The similarity of the extracted watermark with respect to the original one has been evaluated using the normalized cross correlation (NCC) metric value. The comparative results with other existing state of art approaches are tabulated in Tables 12 and 13. The evaluation of the extracted watermark has been further extended by using the

**Table 11** Robustness test results

| Attacked Watermarked image | Attack Types with parameters | Extracted Watermark | NCC value |
|---|---|---|---|
|  | Weiner Filter |  | 0.76 |
|  | Resizing 512 → 128 → 512 |  | 0.85 |
|  | Average Filter |  | 0.88 |
|  | Unsharp Masking |  | 0.90 |
|  | Median Filter |  | 0.86 |

various available error metrics. It is based on mutual relation between pixels, their values and their locations respectively.

Let us assume that $W_{emb}$ is the embedded watermark and $W_{ext}$ is the extracted binary watermark. The total number of true positive, false positive, true negative and false

**Table 12** Comparative Performance based on NCC

| Attacks | our | [32] | [37] | [14] | [15] | [10] | [24] | [41] | [38] |
|---|---|---|---|---|---|---|---|---|---|
| Rotation | 0.89 | 0.63 | 0.92 | 0 | 0 | 0 | 0.83 | 0.98 | 0.98 |
| Noise Addition | 0.98 | 0.75 | 0.99 | 0.82 | 0.87 | 0.89 | 0.76 | 0.98 | 0.95 |
| Median Filtering | 0.86 | 0.89 | 0.92 | 0.85 | 0.82 | 0.75 | 0.93 | 0.98 | 0.98 |
| Blurring | 0.88 | 0.77 | 0.86 | 0.79 | 0.78 | 0.79 | 0.92 | 0.98 | 0.99 |
| Sharpening | 0.96 | 0.81 | 0.98 | 0.82 | 0.93 | 0.89 | 0.81 | 0.98 | 0.99 |
| Resizing | 0.85 | 0.98 | 0.93 | 0.91 | 0.90 | 1 | 0.88 | 0.98 | 0.98 |

negative pixels with respect to $W_{emb}$ and $W_{ext}$ are indicated by $N_{TP}$, $N_{FP}$, $N_{TN}$ and $N_{FN}$ respectively [19, 42]. Some of the objective error metrics are defined as follows:

### *3.2.1 Precision*

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \tag{13}$$

For identical images value of Precision will be 1.

### *3.2.2 Recall/Sensitivity*

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{14}$$

For identical images the value of recall will be 1.

### *3.2.3 F-Measure*

$$FM = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{15}$$

For identical images the value of F-Measure will be 1.

**Table 13** Comparison with other methods

| Parameters | our | [8] | [40] | [35] | [27] | [30] |
|---|---|---|---|---|---|---|
| Security | High | Low | Low | Low | High | High |
| Localization Accuracy | High | Low | High | High | Medium | Medium |
| Recovery Quality | High | Medium | Medium | Medium | Medium | Medium |
| Robustness to JPEG | High | Low | Low | Low | Medium | High |
| Robustness to Gaussian | High | Low | Low | Medium | Low | High |
| Robustness to Rotation | High | Low | Low | Medium | Low | High |

**Table 14** Various Objective Measures for Extracted Binary Watermark against Various Attacks

| Attacks / Objective measures | Precision | Recall | F-measure | SSIM | Specificity | BCR | BER | Geometric Accuracy | NRM | DRD | MPM (x1000) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Salt & Pepper Noise | 0.9807 | 0.9756 | 97.42 | 0.9672 | 0.9692 | 0.9823 | 0.1094 | 0.830 | 0.0981 | 0.1966 | 893 |
| Hist Eq. | 0.8827 | 0.9126 | 95.12 | 0.9591 | 0.8915 | 0.8974 | 0.1594 | 0.897 | 0.1561 | 0.1939 | 816 |
| Rotation | 0.6677 | 0.8834 | 85.32 | 0.8569 | 0.7153 | 0.7819 | 0.2940 | 0.7543 | 0.3945 | 0.2871 | 501 |
| Speckle Noise | 0.7394 | 0.8936 | 87.52 | 0.8641 | 0.7955 | 0.7873 | 0.2291 | 0.7969 | 0.3297 | 0.2274 | 619 |
| Motion Blur | 0.7036 | 0.7247 | 84.42 | 0.8175 | 0.7736 | 0.7771 | 0.3964 | 0.7194 | 0.3359 | 0.3610 | 539 |
| Weiner Filter | 0.6237 | 0.7486 | 81.84 | 0.7402 | 0.7127 | 0.7962 | 0.3610 | 0.7910 | 0.3501 | 0.4820 | 489 |
| Resizing | 0.7496 | 0.7946 | 80.18 | 0.7907 | 0.7739 | 0.7583 | 0.2804 | 0.7631 | 0.3063 | 0.3492 | 703 |
| Average Filter | 0.7530 | 0.7949 | 82.86 | 0.8063 | 0.7946 | 0.8296 | 0.2075 | 0.8040 | 0.2595 | 0.2795 | 731 |
| Unsharp Masking | 0.8593 | 0.9074 | 90.73 | 0.9198 | 0.8983 | 0.8252 | 0.1793 | 0.8582 | 0.2069 | 0.1852 | 802 |
| Median Filter | 0.8183 | 0.8747 | 85.66 | 0.8280 | 0.8915 | 0.8811 | 0.1972 | 0.813 | 0.2684 | 0.2974 | 761 |
| JPEG Compression | 0.6749 | 0.7048 | 81.13 | 0.8379 | 0.8184 | 0.8709 | 0.2789 | 0.8163 | 0.3178 | 0.2804 | 691 |
| Cropping | 0.7456 | 0.7381 | 81.54 | 0.7913 | 0.7754 | 0.3682 | 0.7821 | 0.897 | 0.2699 | 0.2890 | 472 |
| Laplacian Filter | 0.7494 | 0.8671 | 87.84 | 0.8621 | 0.8071 | 0.7793 | 0.2904 | 0.891 | 0.2190 | 0.2395 | 673 |
| Gaussian Noise | 0.7183 | 0.8370 | 83.71 | 0.8591 | 0.8063 | 0.7961 | 0.2854 | 0.727 | 0.2842 | 0.2745 | 892 |

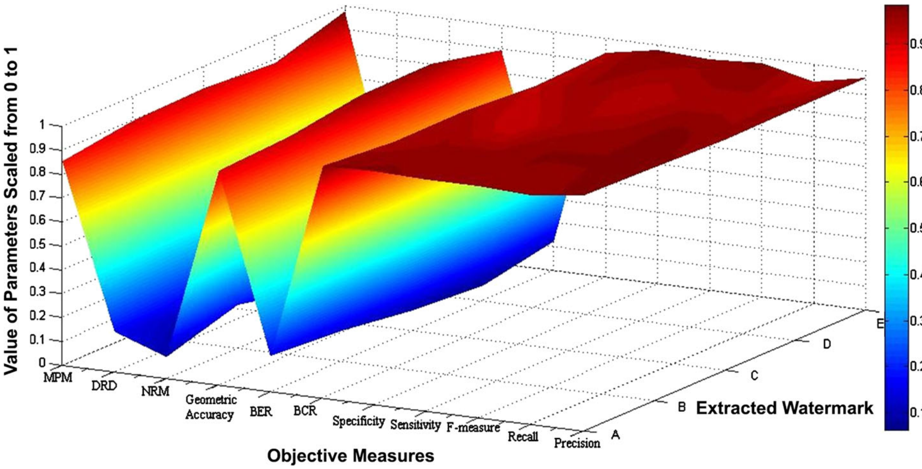



**Fig. 18** Objective Parameters for Extracted Watermark Against Salt & Pepper Noise Attack

### 3.2.4 Structural Similarity Index (SSIM)

SSIM is a Human Visual System (HVS) based evaluation metric used to measure image quality.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_1)} \quad (16)$$

Where $\mu_x$, $\mu_y$, $\sigma_x^2$, $\sigma_y^2$ and $\sigma_{xy}$ are the average, variance and covariance for $x$ and $y$ respectively. The SSIM index value ranges from -1 and 1, with value 1 is case of two identical data sets.

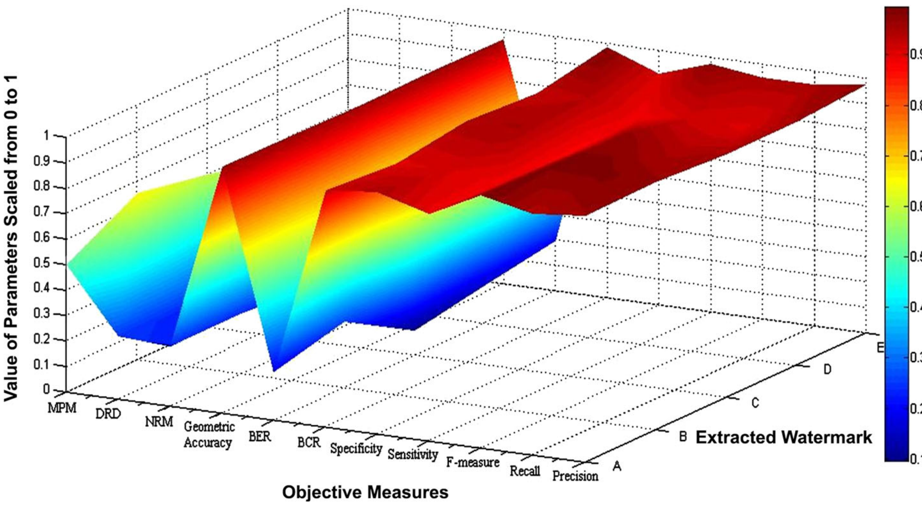



**Fig. 19** Objective Parameters for Extracted Watermark Against Histogram Equalization Attack

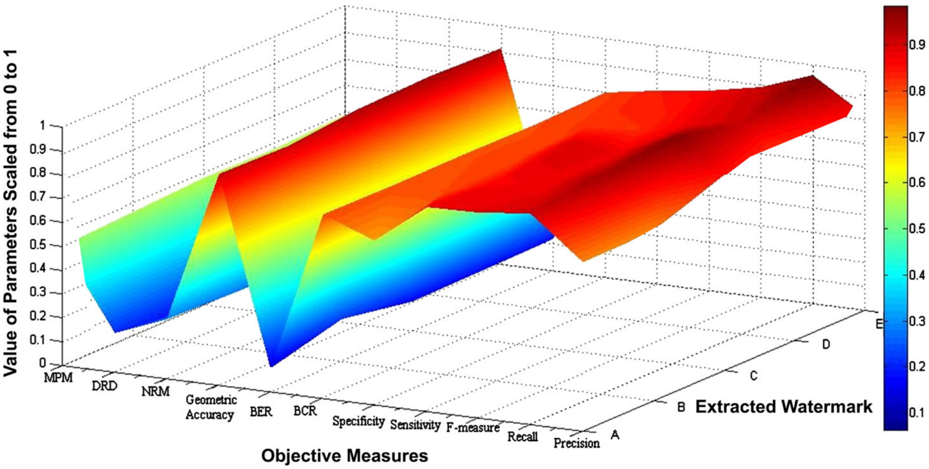



**Fig. 20** Objective Parameters for Extracted Watermark Against Rotation Attack

### *3.2.5 Specificity*

$$Specificity = \frac{N_{TN}}{N_{TN} + N_{FP}} \tag{17}$$

For identical images value of Specificity will be 1.

### *3.2.6 Balanced Classification Rate (BCR)/Area Under the Curve (AUC)*

$$BCR = 0.5 \times (Specificity + Sensitivity) \tag{18}$$

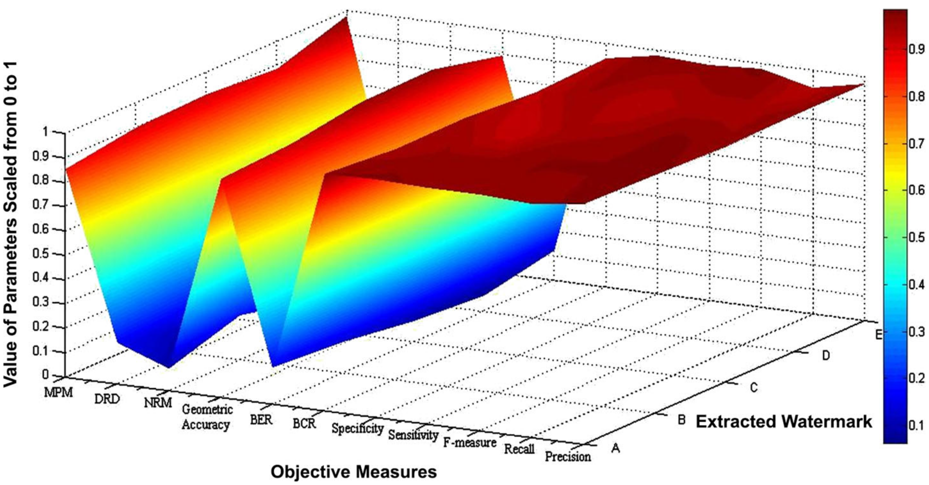



**Fig. 21** Objective Parameters for Extracted Watermark Against Speckle Noise Attack

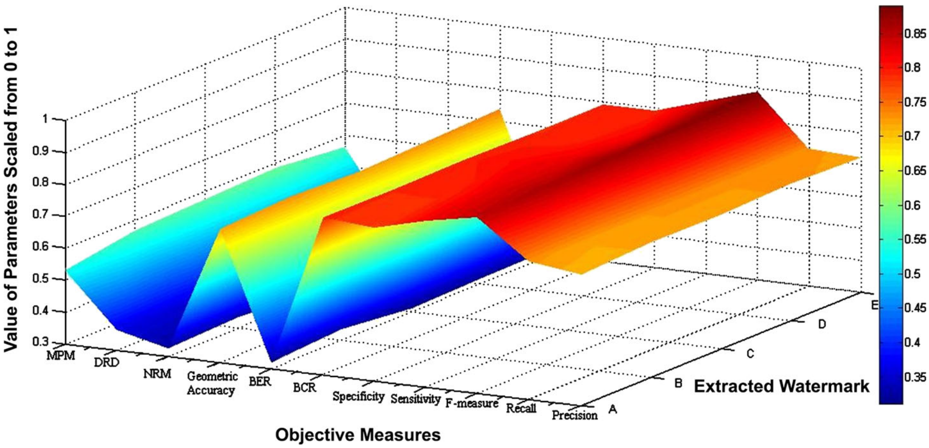



**Fig. 22** Objective Parameters for Extracted Watermark Against Motion Blur Attack

For identical images the value of BCR/AUC will be 1.

### *3.2.7 Balanced Error Rate (BER)*

$$BER = 100 \times (1 - BCR) \tag{19}$$

For identical images the value of BER will be 0.

### *3.2.8 Negative Rate Matrix (NRM)*

The NRM is based on the pixel wise mismatch between the $I$ and $G$.

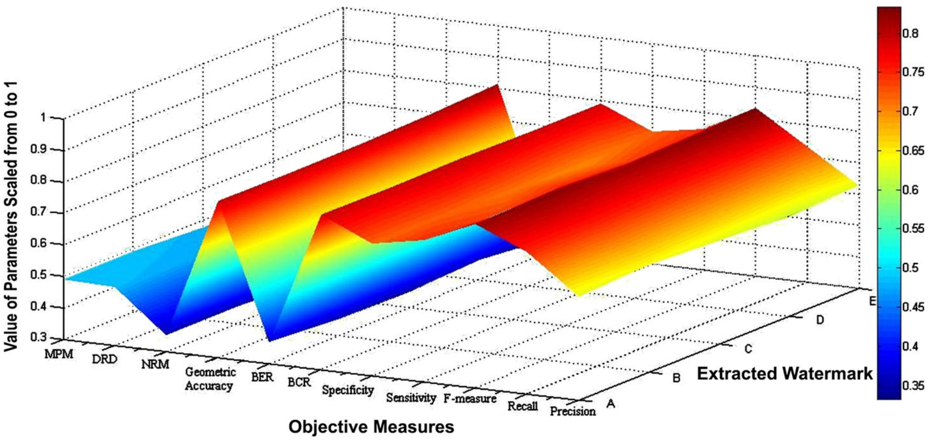



**Fig. 23** Objective Parameters for Extracted Watermark Against Weiner Filter Attack

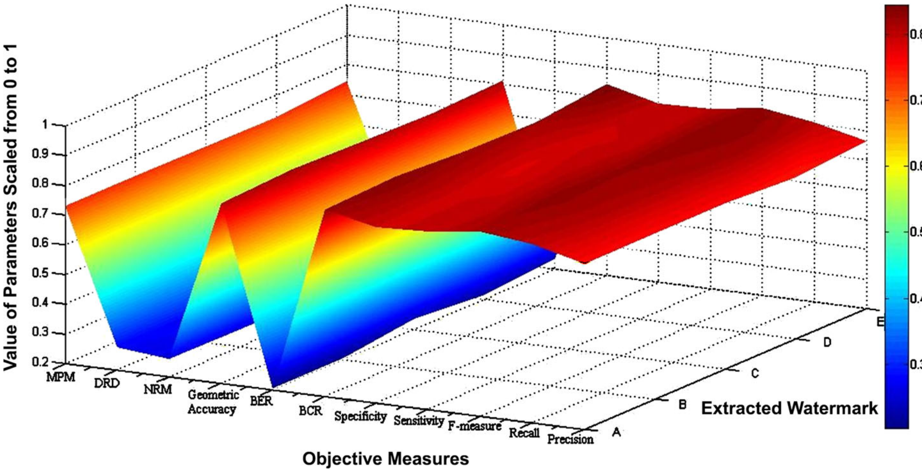



**Fig. 24** Objective Parameters for Extracted Watermark Against Average Attack

$$NRM = \frac{NR_{fn} + NR_{fp}}{2} \tag{20}$$

where

$$NR_{fn} = \frac{N_{FN}}{N_{FN} + N_{TP}} \tag{21}$$

$$NR_{fp} = \frac{N_{FP}}{N_{FP} + N_{TN}} \tag{22}$$

For identical images value of NRM will be 0.

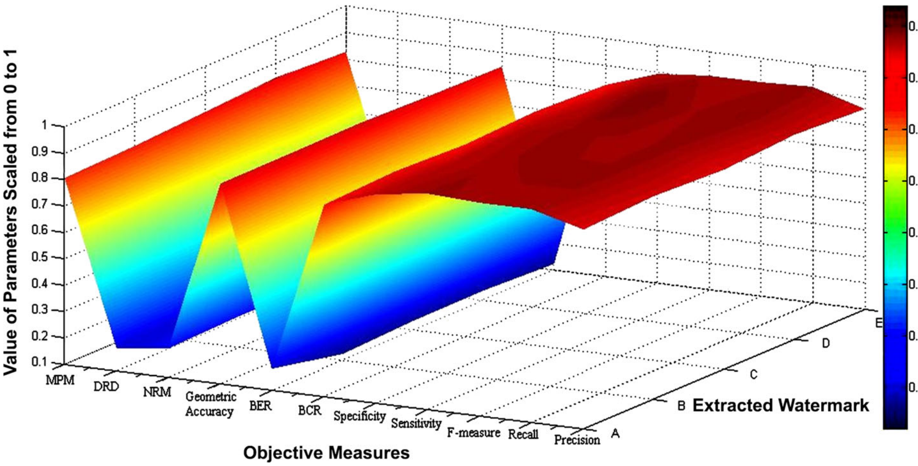



**Fig. 25** Objective Parameters for Extracted Watermark Against Unsharp Masking Attack

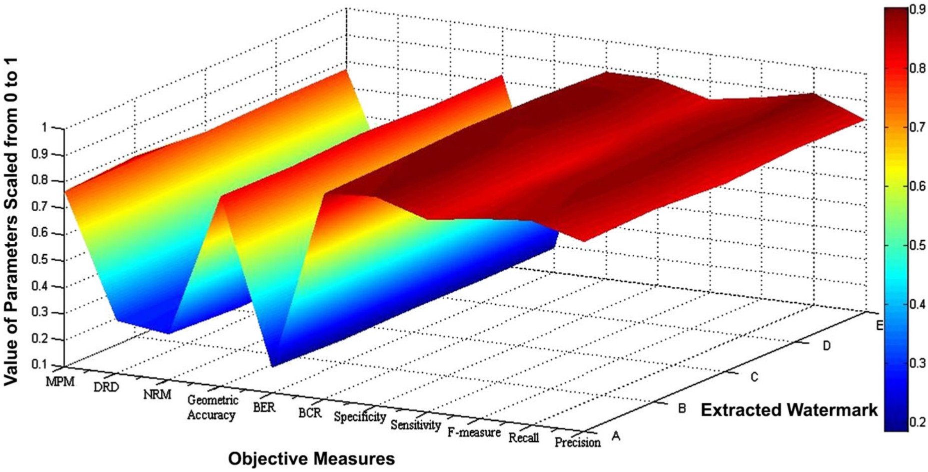



**Fig. 26** Objective Parameters for Extracted Watermark Against Median Fiter Attack

### *3.2.9 Distance-Reciprocal Distortion Measure (DRDM)*

Let $W_m$ is the weight matrix and $i_c$ and $j_c$ are the center pixel.

$$W_m(i,j) = \begin{cases} 0, & \text{if } i_c = j_c \\ \frac{1}{\sqrt{(i-i_c)^2+(j-j_c)^2}}, & \text{otherwise} \end{cases} \tag{23}$$

This matrix is Normalized by.

$$W_{Nm}(i,j) = \frac{W_m(i,j)}{\sum_{i=1}^{m}\sum_{j=1}^{m} W_m(i,j)} \tag{24}$$

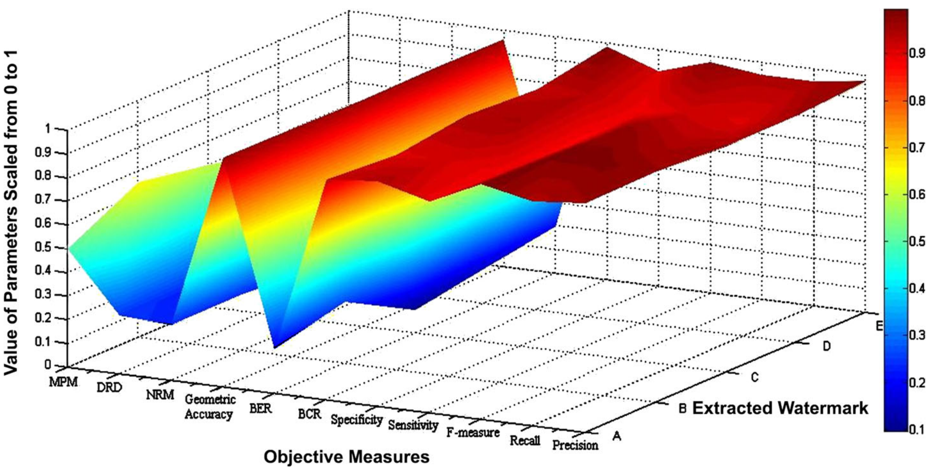



**Fig. 27** Objective Parameters for Extracted Watermark Against JPEG Compression Attack

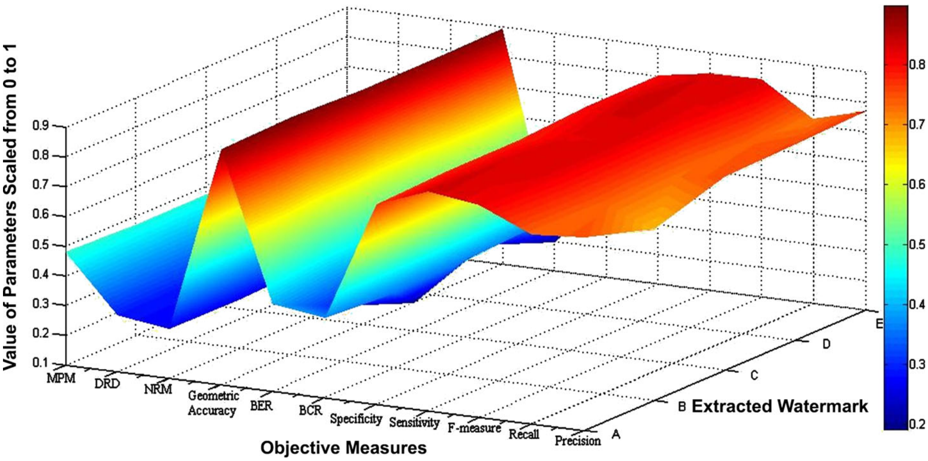



**Fig. 28** Objective Parameters for Extracted Watermark Against Cropping Attack

Now

$$DRD_k = \sum_{i,j}[D_k(i, j) \times W_{Nm}(i, j)] \tag{25}$$

Where $D_k$ is given by $(B_k(i, j) - g[(x, y)_k])$. Thus $DRD_k$ equals to he weighted sum of the pixels in the block $B_k$ of the original image.

$$DRD = \frac{\sum_{k=1}^{s} DRD_k}{NUBN} \tag{26}$$

Where NUBN is the nonuniform blocks in $F(x, y)$.

For identical Image DRDM will be 0.

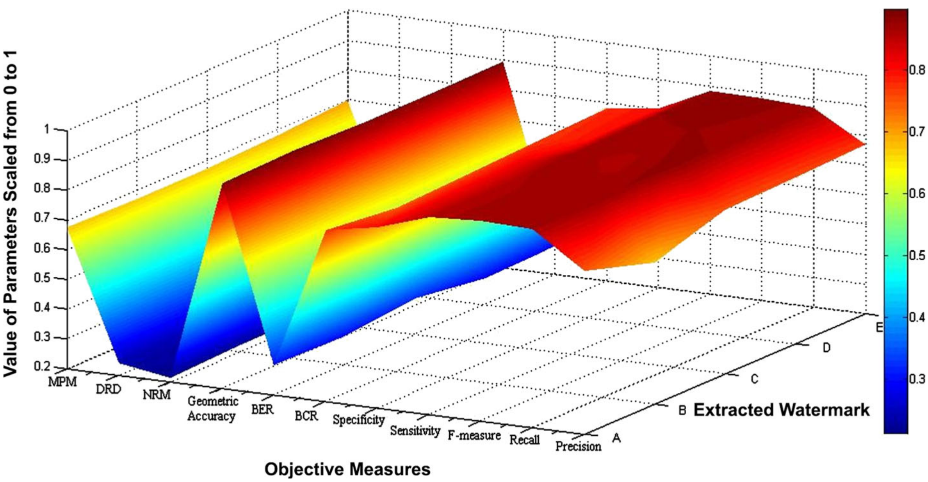



**Fig. 29** Objective Parameters for Extracted Watermark Against Laplacian Filter Attack

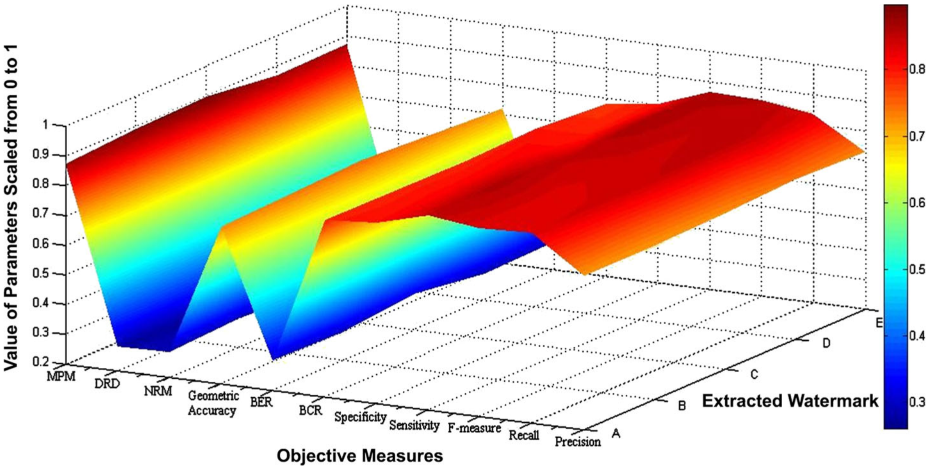



**Fig. 30** Objective Parameters for Extracted Watermark Against Gaussian Noise Attack

### *3.2.10 Misclassification Penalty Metric (MPM)*

$$MP = \frac{1}{2}(MP_{fn} + MP_{fp}) \tag{27}$$

where

$$MP_{fn} = \frac{\sum_{j=1}^{N_{fn}} d_{fn}^{j}}{D} \tag{28}$$

Represents the sum of distances of all false negatives.

$$MP_{fp} = \frac{\sum_{j=1}^{N_{fp}} d_{fp}^{j}}{D} \tag{29}$$

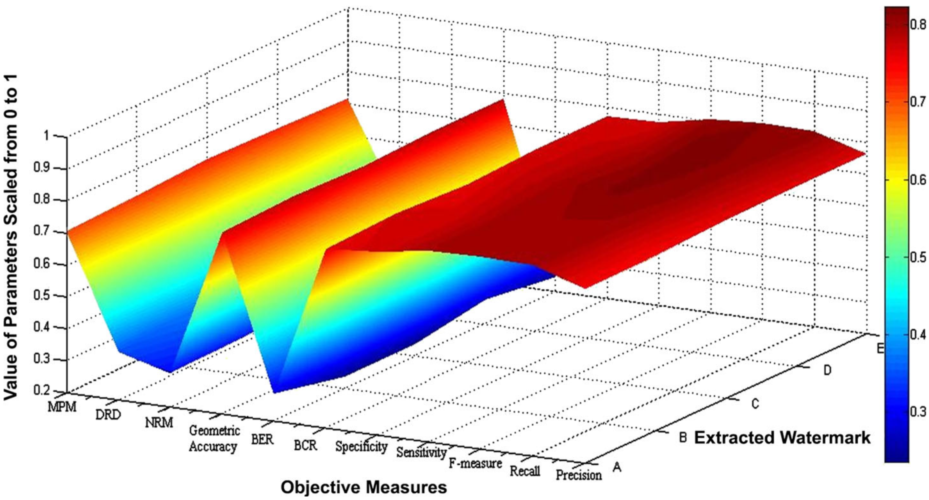



**Fig. 31** Objective Parameters for Extracted Watermark Against Resizing Attack

Represents the sum of distances of all false positives.

For identical images the value of MPM will be 100.

The objective parameters for the extracted watermark tested against the comprehensive set of attacks has been tabulated in Table 14 and their respective 3D graphs has been depicted in Figs. 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30 and 31 respectively.

## 4 Conclusion and future scope

A dual watermarking scheme incorporating both features of ownership assertion as well as integrity check has been proposed here. Recovery information of each $2 \times 2$ sized non-overlapping block was reduced to just eight bits which were further encoded to obtain only four bits and embedded in the mapping block of the cover image. The reduction in storage requirements for recovery bits was utilized for efficiently embedding the copyright information, thus adding the feature of robustness in the scheme. The scheme performed well against comprehensive set of attacks like noises, filtering, histogram equalization, rotation, jpeg compression etc. The pixel level tamper detection of the scheme could chalk out the altered areas accurately for all the three major categories of natural, texture as well as satellite images. The random chaotic mapping of blocks enhanced the efficacy of the scheme against tampers even upto 50 %. Evaluation of extracted watermark logo via variety of suitable error metrics further added to its advantages, with satisfiable PSNR values for both watermarked as well as recovered images.

## References


1. (2013) A blind image copyright protection scheme for e-government. J Visual Communication and Image Representation. 24(7): 1099–1105
2. (2012) A blind reversible method for watermarking relational databases based on a time-stamping protocol. Expert Syst Appl. 39(3): 3185–3196
3. (2014) A low cost fragile watermarking scheme in H.264/AVC compressed domain. Multimedia Tools Appl. 72(3): 2469–2495
4. (2014) An adaptive watermarking scheme for e-government document images. Multimedia Tools Appl. 72(3): 3085–3103
5. (2012) An improved SVD-based watermarking technique for copyright protection. Expert Syst Appl. 39(1): 673–689
6. (2012) Copyright Protection for E-Government Document Images. IEEE MultiMedia. 19(3): 62–73
7. Agreste S, Andaloro G (2008) A new approach to pre-processing digital image for wavelet-based watermark. J Comput Appl Math 221(2):274–283
8. Chamlawi RA, Khan A (2010) Digital image authentication and recovery: sinteger trans- form based information embedding and extraction. Inf Sci 180:4909–4928
9. Chemak C, Bouhlel MS, Lapayre JC (2007) A new scheme of robust image watermarking: the double watermarking algorithm. In: Proceedings of the 2007 summer computer simulation conference, pp 1201–1208
10. Chen B, Coatrieux G., Chen G., Sun X, Coatrieux J, Shu H (2014) Full 4-D quaternion discrete Fourier transform based watermarking for color images. Digital Signal Process 28(1):106–119
11. Chen F, He H, Tai H, Wang H (2012) Chaos-based self-embedding fragile watermarking with exible watermark payload. Multimedia Tools Appl 72:41–56
12. Cox IJ, Miller ML, Bloom JA, Fridrich J, Kalker T (2008) Digital Watermarking and Steganography, 2nd edn. Elsevier, New York
13. Cox I, Miller M, Bloom J et al (2001) Digital watermarking. Morgan Kaufmann, Burlington
14. Das C, Panigrahi S, Sharma VK, Mahapatra KK (2014) A novel blind robust image watermarking in DCT domain using inter-block coef?cient correlation. Int J Electron Commun 68(3):244–53

15. Feng L, Zheng L, Cao P (2010) A DWT-DCT based blind watermarking algorithm for copyright protection. In: Proceedings of the IEEE ICCIST, pp 455–458
16. Gonzalez RC, Woods RE (2001) Digital image processing, 2nd edn. Prentice Hall, Upper Saddle River
17. Guo J, Liu Z, Liu S (2007) Watermarking based on discrete fractional random transform. Opt Commun 272(2):344–348
18. Habib M, Sarhan S, Rajab L (2005) A robust-fragile dual watermarking system in the DCT domain. In: Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems, pp 548–553
19. Haiping Lu (2004) Distance-Reciprocal Distortion Measure for Binary Document Images. IEEE Signal Processing Letters 11(2)
20. He HJ, Zhang JS, Tai HM (2008) Block-chain based watermarking scheme with superior localization. In: IHW, pp 137–160
21. Huo Y, He H, Chen F (2012) Alterable-capacity fragile watermarking scheme with restoration capability. Opt Commun 285:1759–1766
22. Hartung F, Kutter M (1999) Multimedia watermarking techniques. Proceedings of the IEEE special issue on Protection of Multimedia Content:1062–1087. doi:10.1109/5.771066
23. Hsia CH, Guo JM (2014) Efficient modified directional lifting-based discrete wavelet transform for moving object detection. Signal Process 96(3):138–152. doi:10.1016/j.sigpro.2013.09.007
24. Hsu C, Hou Y (2005) Copyright protection scheme for digital images using visual cryptography and sampling methods. Opt Eng 44:077003
25. Katzenbeisser S, Petitcolas FAP (2000) Information hiding techniques for steganography and digital watermarking. Artech House
26. Lee TY, Lin SD (2008) Dual watermark for image tamper detection and recovery. Pattern Recogn 41(11):3497–3506. doi:10.1016/j.patcog.2008.05.003
27. Li GB, Pei SW, Chen G, Cao W, Wu B (2009) A Self-embedded watermarking scheme based on relationship function of corresponding inter-blocks DCT coe cient. In: Proceedings of the 2009 13th international conference on computer supported cooperative work in design, pp 107–112
28. Li QH, Ren GQ, Wu QZ, Zhang XY (2013) Rate pre-allocated compression for mapping image based on wavelet and rate-distortion theory. Int J Light Electron Opt 124(14):1836–1840. doi:10.1016/j.ijleo.2012.05.045
29. Li C, Wang Y, Ma B et al (2012) Tamper detection and self-recovery of biometric images using salient region-based authentication watermarking scheme. Comput Stand Interfaces 34(4):367–379. doi:10.1016/j.csi.2012.01.003
30. Li C, Zhang A, Liu Z, Liao L, Huang D (2014) Semi-fragile self-recoverable watermarking algorithm based on wavelet group quantization and double authentication. Multimedia Tools Appl:1380–7501
31. Lin WH, Horng SJ, Kao TW, Fan P, Lee CL, Pan Y (2008) An efficient watermarking method based on significant difference of wavelet coefficient quantization. IEEE Trans Multimed 10(5):746–757
32. Mohan B, Kumar S (2008) A robust digital image watermarking scheme using singular value decomposition. Journal of Multimedia 3(1):7–15
33. Nikolaidis N, Pitas I (1998) Robust image watermarking in the spatial domain. Signal Process 66(3):385–403. doi:10.1016/S0165-1684(98)00017-6
34. Niu SZ, Shu NF (2009) A digital image double watermarking algorithm based on DCT domain. J Comput Res Dev 46(4):6–10
35. Phadikar A, Maity S, Mandal M (2012) Novel wavelet-based QIM data hiding technique for tamper detection and correction of digital images. J Vis Commun Image Represent 23:454–466
36. Peng Z, Liu W (2008) Color image authentication based on spatiotemporal chaos and svd. Chaos Solitons Fractals 36(4):946–952
37. Rawat S, Raman B (2011) A chaos-based robust watermarking algorithm for rightful ownership protection. International Journal of Image and Graphics 11(4):471–493
38. Rawat S, Raman B (2013) Visual-cryptography-based blind watermarking scheme for copyright protection. In: International Journal of Signal and Imaging Systems Engineering, vol 6, no 3, pp 158–163
39. Singh P, Agarwal S (2015) An efficient fragile watermarking scheme with multilevel tamper detection and recovery based on dynamic domain selection. Multimedia Tools and Applications. doi:10.1007/s11042-015-2736-9
40. Tsai MJ, Chien C (2008) Authentication and recovery for wavelet-based semifragile watermarking. Opt Eng 47(6):19
41. Wang M, Chen W (2009) A hybrid DWT-SVD copyright protection scheme based on k-means clustering and visual cryptography. Computer Standards Interfaces 31:757–762

42. Young DP, Ferryman JM (2005) PETS Metrics: On-Line Performance Evaluation Service. In: Proceedings 2nd Joint IEEE International Workshop on VSPETS, Beijing
43. Zhang X, Wang S, Qian Z, Feng G (2011) Reference sharing mechanism for watermark self-embedding. IEEE Trans Image Process 20(2):485–495

**Priyanka Singh** received B.Tech Degree from HBTI, Kanpur, M.Tech degree from MNNIT, Allahabad and presently pursuing Ph.D. from NIT, Allahabad, India. Her area of interests include Digital Watermarking, Visual Cryptography, and Security related concepts.

**Suneeta Agarwal** received Ph.D. Degree from IIT, Kanpur is professor in the Department of Computer Science and Engineering Department in NIT, Allahabad, India. She has numerous contributions in various international and national journals. Her area of interests include Image Processing, Automata Theory, Compression, Pattern Matching and Fingerprint Recognitions.