Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

Effective Tamper Detection and Recovery of Images after Serious Attacks

Faranak Tohidi*[1], Mohammad Reza Hooshmandasl†[2] and Manoranjan Paul‡[1]

[1]School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW-2795, Australia

[2]Department of Computer Science, University of Mohaghegh Ardabili, Ardabil, Iran.

ABSTRACT

Confirming the integrity of transmitted sensitive digital content is a significant issue due to the evolution in communication technologies and the accessibility of image processing tools. Watermarking has been a successful method of authentication and integrity verification recently. However, several significant problems remain such as confronting some serious attacks and recovery after higher tampering rates. We propose a hybrid method to enable an image to be recovered successfully after a range of attacks. A blind watermarking approach is adopted which includes fragile authentication but robust recovery references. This is performed by

ARTICLE INFO

Article history:
Research paper
Received 20, May 2021
Received in revised form 11, October 2021
Accepted 14 November 2021
Available online 30, December 2021

Keyword: Image watermarking , Tamper detection , Image Recovery , Attack , Tamper Recovery

AMS subject Classification: 05C69.

*ftohidi@csu.edu.au

†Corresponding author:M. R. Hooshmandasl . Email: hooshmandasl@uma.ac.ir

‡mpaul@csu.edu.au

1 Abstract continued

embedding verification code as part of the watermarked data along with key features of the original image into a location that is resistant to the attack. To combat different kinds of attacks, the areas of the image have been investigated to find which area is more likely to be affected in each type of specific attack.

2 Introduction

Digital signatures and watermarking are two methods that have been applied to images in order to verify digital image contents. A digital signature includes the feature information extracted from original images that can be saved as independent authentication information. Then a digital image is verified when the extracted authentication information confirms the accuracy of the content. However, if tampering in image content is detected, a digital signature is not able to realize the location of the tampering. Whereas watermarking can not only detect but also identify exactly where the tampering is. Watermarking is a way in which extra information is embedded into an image to authenticate or verify the integrity of the image. Image watermarking has different applications, and the most important ones are copyright protection and image authentication. An invisible watermarking can be a logo or a message to indicate the owner of the image or some features of the original image to prove its integrity. Self-embedding fragile watermarking schemes have received attention recently for integrity verification and authentication of the image. In self-embedding watermarking, the basic feature of the image is hidden inside itself as watermarked data with the purpose of recovery after tampering [14, 7].
There are three kinds of watermarking: fragile, semi-fragile, and robust watermarking. This classification is made according to the level of resistance against different attacks or manipulation. Fragile watermarking is sensitive to any type of alteration therefore it is easily broken by any change in the content of the image. Fragile watermarking is used for verifying the integrity of the received images. A semi-fragile watermarking scheme is a kind of multi-purpose watermarking which is resistant to some malicious attacks but vulnerable toward some other manipulation. Robust watermarking can be applied to maintain ownership authentication and protect copyright [13, 18].
To detect and localize tampering, watermark data should include an authentication reference and it should be fragile, to be sensitive to detect any kind of attacks. However, after detection of tampering, recovery references are needed to reconstruct the tampered area of the image. Therefore, the watermark which is hidden inside the image should include recovery data and this data must be intact even after tampering. Thus, the recovery data must be robust against any type of tampering and be able to carry accurate data for restoring the tampered regions. It means that watermarked data should convey fragile authentication data and robust recovery data at the same time which makes the problem more serious. To address this problem some papers have suggested that more copies of recovery data should be hidden inside the image, so that after detection of tampering , essential recovery data can be retrieved from the intact areas of the image. This method

is useful but inserting more copies of the reference data inside the image causes a decrease in the quality of watermarked images [13, 18].
Authentication and recovery of the image can be done by using either pixel-wise or block-wise watermarking. In pixel-wise watermarking schemes, hidden data is extracted from the pixel and then embed inside a pixel. But, in block-wise schemes, at first, the image is divided into several blocks then watermarked data is extracted from each block and will be hidden inside the block. Block dependence helps the method to be more resistant against different attacks such as vector quantization (VQ) or collage attack. These two attacks utilize a watermarked image to tamper with a new but similar watermarked image [13, 4]. However, a pixel-wise scheme is able to restore the original image more accurately. In order to take advantage of both block-wise and pixel-wise watermarking methods, we propose a hybrid method that is block-wise during the authentication process and extraction of recovery data therefore it can detect different attacks more accurately. Then recovery can be processed by a second process on each pixel individually.
Although various research papers in the field of image authentication and restoration after modification, however, they all have some inefficiencies. All the earlier papers have achieved an improvement in something, but still, some other issues remain. For example, some competitive methods relevant to image authentication by watermarking have published recently [10, 15, 2, 8, 5]. However, [10, 15, 2] are not performing very well in the higher tampering rate. Although [6] can deliver more accurate tamper detection with good quality of the watermarked image, it is not capable of recovery of the original images and [8, 5] are not capable of recovery after rotation attack.
Qin et al. [11] proposed a method to compress a quality image and called it Optimal Iterative Block Truncation Coding (OIBTC). They applied OIBTC for each block to generate the recovery code. Their method has been tested and has achieved increased quality of a recovered image after most tampering except rotating. They applied two block sizes of 4×4 and 8×8. The bigger size of a block has benefited from the capability of recovery in a higher tampering rate of around 50%, but when the block size is 4×4 the quality of the recovered image is can be better only in lower tampering rates. This is because the recovery data for a block size of 8×8 is more compressed, so embedding redundant data in the watermarked image is possible. However, the big block size cannot deliver a high accuracy of localization of tampering [14, 16].
Shehab et al. [14] proposed a watermarking scheme for image verification and recovery after tampering for medical image applications. In their method, an image is divided into 4×4 blocks. Singular Value Decomposition (SVD) is calculated for any block to obtain block authentication bits and the self-recovery information is achieved by computing an average value of each 2×2 block. Authentication and recovery data of every block are embedded into the two least significant bits (LSBs) of the image pixels of the other block. Arnold transformation is also applied to determine which block is the destination block for embedding both authentication and recovery data. In the past, this method has achieved reasonable quality of the recovered images but has a problem in the accuracy of detection. The accuracy of authentication is low because they embed both authentication and recovery data in the same block. There is also a lack of ability to identify which block

is tampered with, whether the original block or the destination block which contained authentication data. Therefore, False Positive Rate (FPR) increased. Moreover, since an average pixel's values have been used for any block, the recovered image suffers from the mosaic appearance in each recovered block.
F. Tohidi et.al. [17] proposed a novel feature extraction for recovery of tampered image. Their method has achieved good results using new compression strategy to obtain recovery references. However, their method is unable to recover the original images when those images have been rotated.
In this paper, we propose to introduce a completely new method of self-embedding which benefits from new feature extraction to achieve recovery reference as well as introducing a new logistic strategy to embed data. This new method makes the recovery reference more robust against not only higher rates but different kinds of tampering. Our method is completely fragile in terms of detection and localization of any kind of tampering, because our watermark data includes highly sensitive authentication codes that are easily affected by any feasible attack. At the same time, we introduce recovery codes that are resilient enough to able to recover the original image after detection of attacks. Therefore, our method can resist higher tampering rates, using carefully placed recovery codes, so that it is also more robust in terms of restoring the original image.
Contribution of this Research

1. Achieving successful image recovery after multiple attacks.
2. Introducing an effective but short recovery code, to prepare enough capacity for embedding several copies of data.
3. Carefully embedding another copy of the recovery data as backup recovery data in case of tamper coincidence.
4. Finding those areas of the image which are less likely to be affected by each particular attack.
5. Not distributing important recovery reference across the image uniformly as it is usual but embedding in only the safe areas of the image after attack.
6. Introducing post-processing step after recovery to remove mosaic shape of the recovered blocks and achieving higher quality of the restore image.

3 Proposed method

The image is divided into 4×4 blocks and authentication and recovery references are calculated for every block individually. There are different watermarking data for each block separately including authentication reference and recovery reference. Unlike the [14] method the destination embedding block for the block's authentication reference is different from the same block's recovery reference. This is done because of achieving

greater accuracy of the detection of tampering. Authentication data will be embedded in each block itself and recovery data will be embedded into the other block. Feature extraction to obtain recovery references can be achieved with the help of a new data compression method that is explained later. Then authentication reference for each block will be computed according to the block's information and its recovery reference. To ensure security and providing better recovery capability, each block's recovery reference transforms into the other block using a secret key in a way that it can only be reversed back by the previously used key. For this reason, Arnold transformation is used to find destination blocks for embedding recovery references in such a way that watermarked data can be distributed into an image's blocks [14, 16].

To apply Arnold transforms, an image divided into non-overlapping blocks is considered as a two-unit function $f(x, y)$. A source block (x, y) is mapping to a destination block (x', y') using

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & k_1 \\ k_2 & k_1 k_2 + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} mod\ N \tag{1}$$

The parameter of "N" is the number of blocks in the image. Parameters k_1 and k_2 can be used as secret keys.

There are various types of tampering and a good fragile watermarking should be easily affected by any of them to detect. Some of the tampering attacks are Copy-paste attacks, text addition attacks, etc. More serious attacks can be as follows:

Copy Move Tampering Attack

These attacks copy a slice of a watermarked image and paste it into another area in the same watermarked image to forge that image.

Collage Attacks and Vector Quantization Attacks

Vector quantization (VQ), and Collage attacks have similar structures. Both of them use watermarked images made by the same key to copy and pase a desirable section. However, Collage attacks copy a piece from a watermarked image and paste it into the same place in the second watermarked image. Figure 1 and Figure 2 illustrate how Copy-move and College attacks attacks manipulate an original image.

Figure 1: Copy Move Tampering attack.

Rotation attack

(a) Original image (b) Second image (c) Tampered image (d) Tampered area

Figure 2: Collage Attack, here, is the insertion of this sign in exactly the same map area

Rotating of an image can be consider as intentional or unintentional attacks. However, most of tamper recovery methods are unable to recover the original image after rotating specially when the image is rotated in the same space.
In this section, there is a brief explanation about how an image is rotated:
The coordinates of a pixel (x_1, y_1) in the image when rotated by an angle α around the pixel (x_0, y_0) will become (x_2, y_2), as the 2 and 3 show [3]

$$x_2 = \cos(\alpha) \times (x_1 - x_0) + \sin(\alpha) \times (y_1 - y_1) \tag{2}$$

$$y_2 = -\sin(\alpha) \times (x_1 - x_0) + \cos(\alpha) \times (y_1 - y_1) \tag{3}$$

Users can choose how the pixels in the image can be adjusted in the cartesian coordinates after rotating. The common ways of adjusting are such as [3]

- Nearest neighbour interpolation: The value of the nearest pixel that the rotated point falls within is selected as the value of the rotated pixel.
- Bilinear interpolation: The mean value of the pixels that are in the nearest 2×2 neighborhood is selected as the value of the rotated pixel.
- Bicubic interpolation: The mean value of the pixels that are in the nearest 4×4 neighborhood is selected as the value of the rotated pixel.

The rotation of an image affects the capacity required to store rotated image data and it needs a greater space to save. Some data around the image are lost if the image is kept in the same space during rotation. The lost data are around the corner of the image and the amount of them depends on the angle of rotation. This concept is illustrated in Figure 3. The following notions are considered in the proposed method for rotating an image:

- An image is an array $(N \times N)$ of pixels. Each pixel is characterized by (x, y) coordinates and its value. The image is rotated around the center of the image $(N/2, N/2)$.

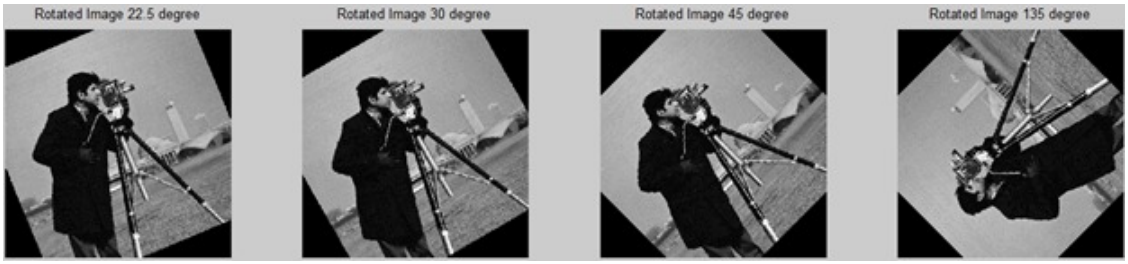

Figure 3: Rotating image with different angles and its effect on the amount of data lost.

- Nearest neighbor interpolation is selected as a method of rotating an image. The reason for choosing this is that it avoids missing the hidden data inside the Least Significant Bits (LSBs).
- The size of the image is not enlarged during and after rotation. Because we want to use rotating image in the same space to preserve the bandwidth for better performance of transferring.

3.1 Feature extraction for generating recovery reference

Feature extraction can be performed using our new compression method which has two following advantages compared with the existing ones:

1. A recovery reference provided by the proposed method takes less space and recovers a better image comparing the current ones therefore, it works better for tampering recovery. In other words, this compression method can provide a recovery reference that is able to recover the original image with higher quality while the rate of compression is also great.
2. Utilizing the proposed compression method gives an opportunity to exploit the advantages of the similarities between pixels and blocks in order to introduce a totally different logistic strategy for extracting and embedding recovery data. This new compression method can extract another recovery reference, as backup recovery information which needs much less capacity for embedding.

Thus, we have better control to embed watermarked data in terms of location, amount, and the number of copies into the compressed image. Recovery references contain two parts, the first and the reserved recovery data. The reserved references are prepared to be used in case of happening tampering coincidence when the first recovery data is also tampered with and cannot be trusted for the recovery. The first recovery is used more frequently therefore it should be selected in a way that can deliver better quality of the recovery. As a result it is more complete and needs more capacity to be hidden. The reserved recovery data belongs to the block's neighbors is chosen to be more compressed because it will be used just in case of damaging the first one. It helps to reduce the amount of watermark data and increase the quality of the watermarked image. The

recovery reference including first and reserve can be obtained for every 4×4 block as follows:

- At first the average values of any four 2×2 inside blocks is calculated by the following formula.

$$M_j = \frac{1}{4}\sum_{i=1}^{4} P_{i+4(j-1)} \quad j = 1, 2, 3, 4 \tag{4}$$

  Where $P_i\,(l = 1, 2, \ldots, 16)$ are pixels inside the block and $M_j\,(l = 1, 2, \ldots, 4)$ are related mean values (Figure 4).

- These average values arranged in ascending order according to their values, i.e.,

$$S = \{M_1, M_2, M_3, M_4\}$$

  In which M_x are the average values of any four 2×2 inside blocks and $M_1 < M_2 < M_3 < M_4$ Here, M_1 consider as a Minimum mean value of the block and M_4 is considered as the Maximum mean value of the block.

- The first mean value which is the minimum one (M_1) is chosen as a part of recovery reference. Here, the number of 5 bits is assigned for this value as its 5 Most Significant Bits (MSBs).

$$M_{1,t} = floor\left[round\left(M_1\right)/2^{t+3}\right] mod\ 2 \quad t = 0, 1, \ldots, 4 \tag{5}$$

  The function $floor\,(.)$ returns the nearest integer towards minus infinity of the input, the function $round\,(.)$ returns the nearest integer of the input, (see Figure 4), and t $(t = 0, 1, \ldots, 4)$ denote the 5 MSB bits of the M_1.

- The other mean values (M_2, M_3, M_4) can be achieved by computing and applying β. The number of 4 bits is allocated to β because of similarity between near pixels and blocks, 4 bits are enough β. It can be obtained using the following formula:

$$\frac{M_4 - M_1}{3} = \beta \tag{6}$$

$$\begin{cases} M_1 + \beta \cong M_2 \\ M_1 + 2\beta \cong M_3 \\ M_1 + 3\beta = M_4 \end{cases} \tag{7}$$

- To find which block has which M_1, we define the following codes for mb.

$$mb = \begin{cases} 00 & M_1 \\ 01 & M_2 \\ 10 & M_3 \\ 11 & M_4 \end{cases} \tag{8}$$

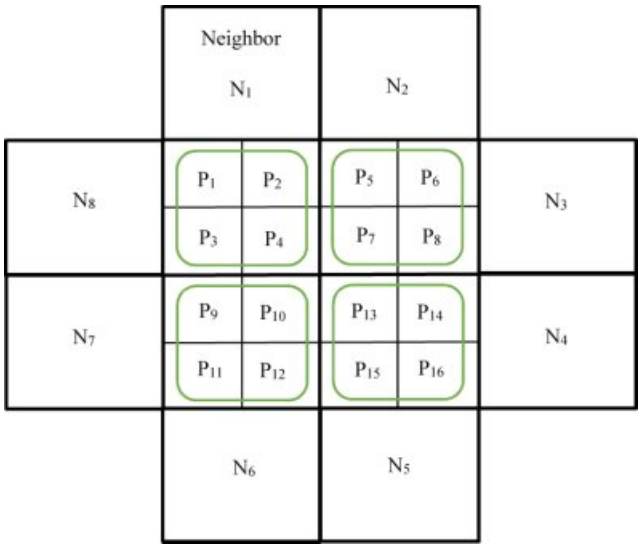

Figure 4: Extracting a block recovery reference.

- There are 8 bits allocated for 8 neighbor blocks, means that each neighbor has 1 bit that shows whether its mean value is smaller than the blocks mean value or not.

$$N_i \begin{cases} 0 & if \quad N_i < its\ neighbor \\ 1 & if \quad N_i \geq its\ neighbor \end{cases} \tag{9}$$

Block recovery reference includes: $M_1, \beta, N_i\,(i = 1, 2, \ldots, 8)$ for 8 neighbours and mb for 4 inside blocks. The number of allocated bits for the recovery data is as follows:

$$M_1\,(5bits)\,, \beta\,(4bits)\,, N_i s\,(8bits)\, and\ mbs\,(8bits)\,.$$

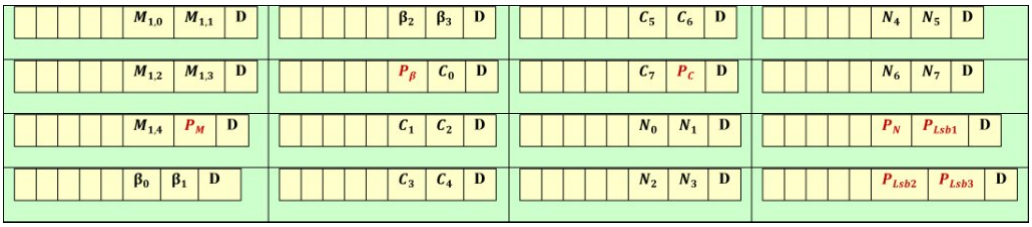

Figure 5: Embdding recovery and authentication reference.

Authentication reference for each block is calculated according to its embedded data. A parity bit for every distinct value of recovery reference including $M_1, \beta, N_i and\ mb$ will be calculated and added to them as watermark data which much be hidden inside the third and second LSBs of the block. In order to achieve more accuracy and decrease False Negative Rate (FNR), the additional three parity checks are done on the first and second, and third LSBs of the destination block to ensure that the probability of tamper detection is high enough. Therefore, the number of bits for an authentication reference of a block

size of 4 is 7 bits. Figure 5 shows how Authentication reference and recovery reference can be embedded in the LSBs of the destination block. In this figure, Ds are navigation bits that are embedded in the first LSBs of the destination block's pixels. $P_x s$ are parity checks as the block authentication reference and the rest are the block recovery references. In this Figure mbs are shown by C. Block authentication and recovery references are embedded inside the second and third LSBs of the destination block's pixels.
As noted, the authentication reference for every block is computed according to its embedded recovery data then embedded again into its own block. As a result of having seven times of checking parity, the probability of not detecting tampering is very low and is calculated by binomial distribution formula. This is because if the number of bit errors is even a single parity bit will fail to realize the error has happened. The probability of undetected error is calculated by considering the number of combinations of every two bits which can be in error.

$$P\left(undetected\right) = \binom{n}{x} p^x q^{n-x} \tag{10}$$

Here "n" is the number of bits and "x" is the number of even errors. "p" is the probability of changing (an error) for every bit and "q" is the probability of not changing. When parity checks are done on the amounts of M_1, the number of bits is 5. For example, the probability of error for parity check for $M_1 is$:

$$P\left(undetected\right)_{n=5} = \binom{5}{2}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^3 + \binom{5}{4}\left(\frac{1}{2}\right)^4\left(\frac{1}{2}\right)^1 < \frac{1}{2} \tag{11}$$

When parity check is computing on LSBs, the number of bits is 15. Therefore, the probability of error can be calculated by Formula 12.

$$\begin{aligned} P\left(undetected\right)_{n=15} &= \binom{15}{2}\left(\frac{1}{2}\right)^2\left(\frac{1}{2}\right)^{13} + \binom{15}{4}\left(\frac{1}{2}\right)^4\left(\frac{1}{2}\right)^{11} + \binom{15}{6}\left(\frac{1}{2}\right)^6\left(\frac{1}{2}\right)^9 \\ &+ \binom{15}{8}\left(\frac{1}{2}\right)^8\left(\frac{1}{2}\right)^7 + \binom{15}{10}\left(\frac{1}{2}\right)^{10}\left(\frac{1}{2}\right)^5 + \binom{15}{12}\left(\frac{1}{2}\right)^{12}\left(\frac{1}{2}\right)^3 + \binom{15}{14}\left(\frac{1}{2}\right)^{14}\left(\frac{1}{2}\right)^1 \\ &\cong \frac{1}{2} \end{aligned} \tag{12}$$

If other probabilities are computed, for all recovery data such as $N_s, C_s \; and \; B_s$it can be seen that all of them are not more than 1/2. As a result of all these parity checks for 7 times, the probability of not detecting (FNR) is far less than $\left(\frac{1}{2}\right)^7$.

3.2 Embedding data

Recovery references and calculated parity bits must be embedded inside the third and second LSBs of the block's pixels. The first LSBs of the block's pixels are reserved for the

navigation bits. To add this information to the watermarked data, all LSBs of all pixels must be changed to 1, apart from pixels that are situated on the main diameter of the image. The main diameter of the image is where the x-coordinates of the pixels are equal to y-coordinates. The first LSBs of these pixels are set at 0. This is because later we are going to use these LSBs to find the correct position of the image in case of occurring a rotation attack.

Finding the best area for embedding

Because of the difficulties with rotation attack, we are not going to hide data across the image blocks uniformly, as is the usual method of embedding. therefore, we try to embed data in a way that is more robust against rotation attack. Figure 6 shows which regions in the image are safe and which ones are not. Figure 6.a shows when an image is rotating by 45 degree the amount of data lost will be equal to the sum of the areas of 4 blue triangles. Therefore, this data lost can be calculated using the following Formula: 13

$$\frac{4 \times \left[\frac{1}{2}\left(\sqrt{2}-1\right) \times \frac{N}{2} \times \frac{N}{2}\right]}{N^2} \cong 21\% \tag{13}$$

It means that if the image is rotated 45 degrees, approximately 21% of the whole image will be lost. Figure 6.b shows that outside the circle is not a safe area for embedding. In fact, around 21.5% of the image is not safe for embedding in case of rotation attack. This amount can be calculated using the Formula 14.

$$\frac{N^2 - \pi\left(\frac{N^2}{4}\right)}{N^2} \times 100 = \left(1 - \frac{\pi}{4}\right) \cong 21.5\% \tag{14}$$

(a) (b) (c)

Figure 6: New scheme for embedding vulnerable data from corners of the image, to inside the safe circle of the image.

When an image is rotated, it needs more space to save its data therefore the corners of the image will be lost in the same space. Thus, in the proposed method, the extracted

features or the recovery references of the corners of the image are embedded inside the middle areas of the image, which is secure against rotation. Then, the proposed method can extract the features missing from the corners of the rotated image and restore them using hidden data which are embedded in the middle areas of the image. The detailed steps are:(Figure 6.c

- An image is divided into quadrants, and each quadrant is also divided into the other quadrants(4 blocks).
- We labeled the blocks in every quadrant, A B C D.
- Blocks B and C split again into B1, B2 horizontally/ vertically and C1, C2 vertically/ horizontally.
- Recovery code from each quadrant is exchanged with the diagonally opposite quadrant of the image. Recovery code for the blocks inside each quadrant is embedded in the same label and color as Figure 6.c shows. The purpose of this step is to be sure that there is a copy from the data related to the corners of the image into the middle of the image.
- Inside each block, the recovery reference is distributed using Arnold Transform to avoid predicting the position of embedding by any possible attacker.

3.3 Tamper detection, localization and recovery

In order to detect and localize the tampered areas of an image on the receiver side, firstly the image should be tested to find whether the image is rotated or not. This test can be done by checking the first LSB of all pixels. All pixels along the main diameter must be 0 and the LSBs of the other pixels need to be 1. Otherwise, those pixels have been tampered with. If there is any mismatch in the first LSBs of the image's pixels with whatever it should be, corruption has taken place. The second step is finding which corruption has occurred. To find this, it is needed to rotate the image by one degree at a time. Rotating the image will continue until most pixels that have 0 LSBs are aligned on the main diameter of the image. Those pixels which have different LSBs than they should have, have been tampered with. At this moment, the image is divided into blocks 4×4. Then for each block, parity bits will be extracted to compare with the hidden content of the block whether there is any mismatch. If there is a mismatch this block is marked as a tampered block. After doing above instructions, the image is in the right position and the marked block should be recovered. Destination for the block's recovery reference can be find using Arnold transform.
When the destination is found, another authentication test should be done to make sure the block containing the first recovery data is still intact. If the first recovery data is not valid the reserved recovery data embedded in the block's neighbor should be found. Again, another test will be done to find that the reserved data is available and intact. In this situation, a tampered block can be recovered with the help of its neighbor's watermarked

data. If both recovery data of a block are damaged, the average value of all its available intact neighbor blocks can be used to recover.

3.4 Post-processing after recovery

Mean values of all 2×2 pixels inside the tampered blocks can be recovered by the 5 Most Significant Bits. The recovered blocks may have a mosaic shape because of using the same values for all pixels of the 2×2 block. Another processing is running for achieving a better quality of the recovered image. Every pixel of the recovered blocks according to their position in the block will be calculated again.
As it can be clearly seen from Figure 7, every pixels can be affected by its four neighbors . Any pixels of the 2×2 block which is recovered by the mean of the block is affected two times by its own block mean and one time by each of two neighbor blocks mean as well. Therefore, the process after recovery can be done by Formula 15.

$$P_{xi} = [2 \times M_x + Neighbor_1\ Mean + Neighbor_2\ Mean]\,/4 \tag{15}$$

$$i = 1, 2, \ldots, 4 \quad x = 1, 2, \ldots, 4$$

Where "i" defines the number of pixels in the 2×2 block and "x" defines the number of the 2×2 block.

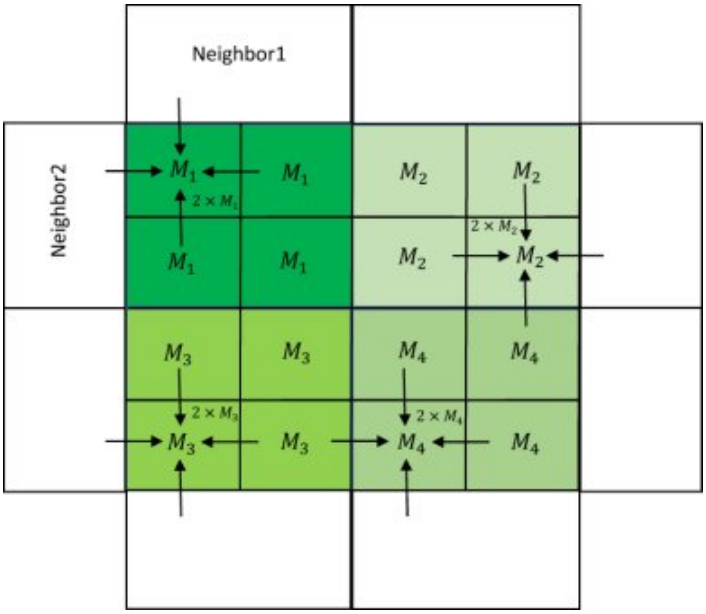

Figure 7: Improving quality of recovered block.

When all of the five Most Significant Bits (MSBs) of all tampered block's pixels are recovered, all the intact and the recovered blocks must be combined together to recover the image. To decrease distortion between the original image and the recovered image caused by the LSBs, the other process should be done. Since distortion for these three

LSBs bits is calculated by Formula 16, the binary value of 100 can be the best value for these LSBs to minimize the amount of distortion.

$$Distortion = \sum_{i=0}^{i=7} (L_i - x)^2 \tag{16}$$

This distortion defines alteration by these LSBs, L_i is the real amount of three LSBs that can be 0, 1, and 2 to 7 (in a decimal form). It can be easily calculated that the amount of 4 (100 in the binary form) can be the best value for minimizing this distortion.

4 Experimental Results

Some standard images have been applied to prove the efficiency of proposed method. Performance analysis of the proposed method has been performed on the quality of recovered images by conducting experiments in standard 512×512 images that are listed in Table1. The quality of recovered images is compared with the quality of watermarked image with standard quality measurements Peak Signal-to-Noise Ratio (PSNR)) in same conditions or similar tampering rates. The comparison results of the proposed method and [14] and [7] are shown in Table 1, when tampering rates(t) were 30% without considering rotation attack.

Table 1: Comparison the results of Proposed method with [14] and [7] in terms of PSNR for different standard images when tampering rate is 30% (Excluding rotation attack).

Standard Images	[14]	[7]	Proposed Method
Lena	35.28	33.64	35.25
Barbara	26.84	24.82	25.23
Mandril	34.32	32.09	33.78
Woman-Darkhair	43.12	42.04	44.45
Woman-Blonde	33.49	32.27	32.92
Living Room	31.36	29.55	30.68
pepper	35.37	34.17	35.72
Lake	32.90	31.47	32.92
JetPlane	37.22	35.44	37.62
CameraMan	37.52	34.41	36.40
House	44.44	43.25	46.73
Pirot	33.61	31.95	32.96

Here, there are some visual results achieved by the proposed method after different tampering attacks. Some of these attacks have been introduced in the proposed method section. Figure 8 to Figure 13.
Figure 12 and Figure 13 shows how the proposed method can cope with multiple attacks and rotation attacks. As it can be seen the position of the pixels is firstly tested then cor-

rected. The next step is identifying which areas have been tampered with. If there is any tampered region, the recovery code will be extracted from those areas that are not altered and contain embedded recovery references to restore the original image. In the proposed method, first of all, any change in the position of the pixels must be detected. Then if there is any change in the position it can be corrected. After that, the blocks' content is checked whether there is any mismatch and which ones have also been corrupted with other attacks. Therefore, unlike the current methods, if the rotation attack occurs, the proposed method is still capable of recovery of the original image. Because as mentioned before, at most 22% of the image is affected when a rotation attack happens which is the lost data related to the corner of the images.

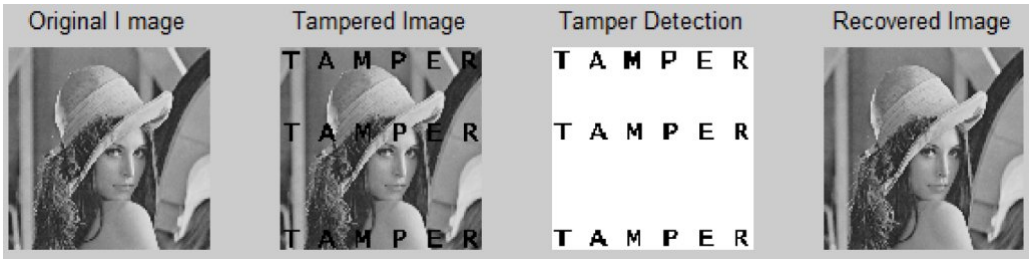

Figure 8: Results of text addition attack by the proposed method.

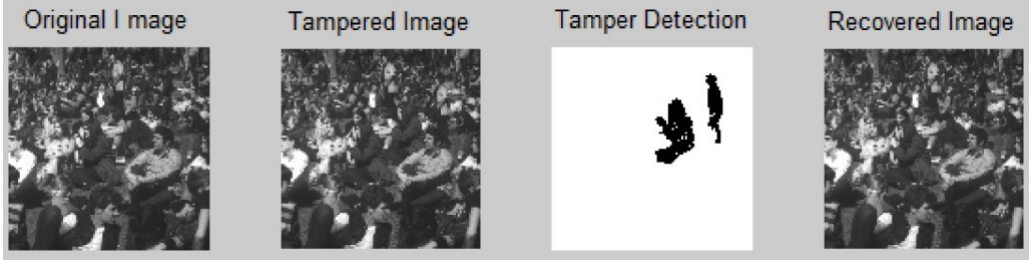

Figure 9: Results of copy paste attack by the proposed method.

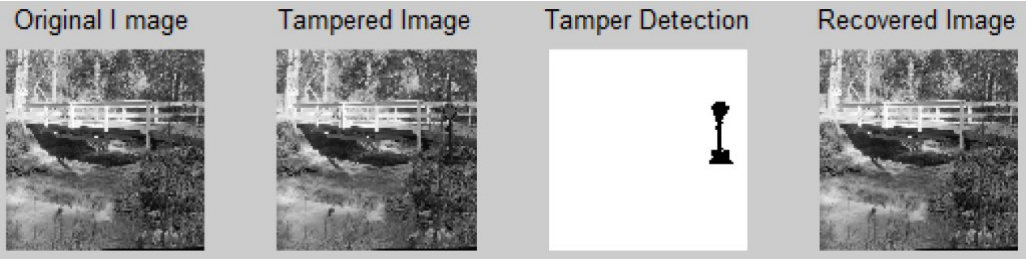

Figure 10: Results of VQ attack by the proposed method.

Since the existing method [14] has been proposed for medical images, a visual result for a medical image (Liver) is prepared to show that our method can also work on medical images. The Figure 14 shows the visual comparison between the result of our method and

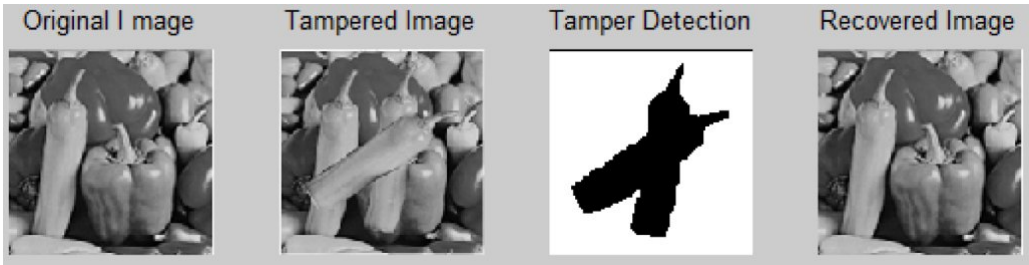

Figure 11: Results of copy-move attack by the proposed method.

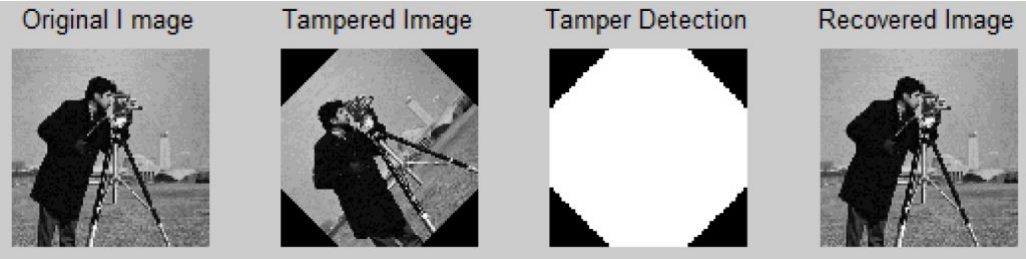

Figure 12: Results of Rotation attack by the proposed method.

Figure 13: Results of Multiple attacks by the proposed method.

[14]. It can be clearly seen from the figure that our recovered image has achieved better quality than [14]. The piece of the recovered region in the results has been enlarged in figure 15 to show the quality of our result is obviously higher and proposed method can eliminate visual mosaic shape in the recovered blocks.

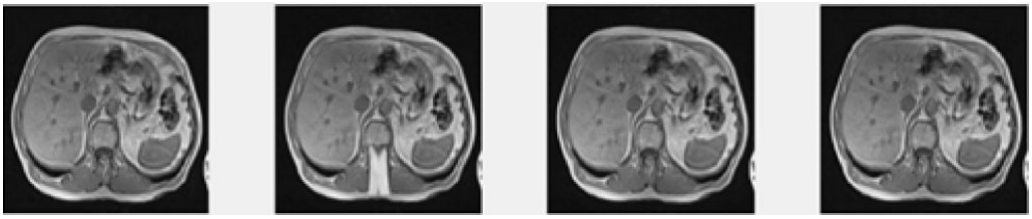

Figure 14: Original image - Tampered image - Recovered image by [14] - Recovered image by the Proposed method.

Table 2 shows the proposed scheme has higher accuracy of tamper detection because the

Figure 15: Recovered tampered area by [14] - Recovered tampered area by proposed method and eliminating mosaic visual

authentication reference is more sensitive to any kind of modification. The first reason is that the probability of detecting tampering for each block is far higher than the trace of SVD which is used in [14] as an authentication reference. Another reason is that in the proposed method, each block has its own authentication reference and recovery reference belonged to the other block, therefore, the possibility of happening False Positive Rate (FPR) decreases. This problem in the scheme of [14] occurs because the destination block includes both the authentication and the recovery reference. Thus, if the destination or source block is tampered with, their method is not able to realize which block is tampered with and where the image should be recovered exactly. The improvement can be clearly seen in Table 2 when the proposed scheme has been compared with the scheme of [14] and [7]. In this table FPR is False Positive Rate.

Table 2: Comparison the accuracy of detection of different attacks excluding rotation attack.

Paper	FNR	FPR
[14]	0.41	0.013
[7]	0.0008	0.0044
Proposed scheme	<0.0078	0.0003

The other measured quality results of the recovered images with different tampering rates are shown in Table 3. The results were recorded for tampering rate (t) up to 50% excluding rotation attack.
Tables 4 and Table 5 show the results of different methods including the proposed method in terms of the capability of detection, localization, and recovery of tampered image after rotating the image . These tables show that the accuracy of the proposed method has advantages, mainly in rotation and multiple attacks. As it can be seen from Table 4 when the rotation attack has happened, the existing methods mostly fail to detect the area of tampering. The reason is that, in fact, their method fails to detect that rotation has happened. As usual, to detect any tampering in the content they compare the extracted watermark from a block in the same position. Therefore, they are not able to detect position changes in the image as a result of rotating. They distinguish that all the blocks in the image have tampered with and cannot realize that just the position of the blocks has been changed not the blocks' contents. Considering the above, it should be reminded that although there are several current methods that are able to recover tampered images with good quality. However, they fail to restore the original image after the rotation

Table 3: Average PSNRs of the recovered image by different methods at different tampering rates excluding rotation attack.

t=Tampering rate / Methods	t =30%	t =40%	t =45%	t =50%
[12]	33.5	32.0	31.5	-
[14]	32.0	30.0	29.5	-
[11] 4×4	36.0	34.0	33.0	-
[11] 8×8	35.5	34.0	32.5	31
[16]	33.5	33	32	-
[10]	31.5	30.5	28.5	26
[15]	32	30.5	30	29
Proposed Method	35.5	34	33.5	31.5

Table 4: Comparison of the tamper localization accuracy against rotation attack with different angles ($i.e., \alpha$)

Methods	$\alpha = \pi/8$ Radians	$\alpha = \pi/4$ Radians	$\alpha = \pi/2$ Radians	$\alpha = \pi$ Radians
[12]	0.1	0.2	0	0
[14]	0.1	0.2	0	0
[11]4×4	0.1	0.2	0	0
[11]8×8	0.1	0.2	0	0
[16]	0.1	0.2	0	0
[10]	0.1	0.2	0	0
[15]	0.1	0.2	0	0
Proposed Method	0.8	0.8	0.9	0.9

attack. In their methods, all the blocks are marked as tampered after rotation and it is detected that the content of the image has changed 100%, and with 100% tampering, the whole image is not able to be restored. Table 5 can prove our mentioned claim. In this table, common attacks include general attacks, cropping, collage, and VQ attacks [4, 14]. Apart from the proposed method, there is another method [1] in Table 5 that is also able to recover the rotated image. This method [1] unlike the proposed method is only able to recover the original image if the image is rotating in the enlarged space and it cannot recover the rotated image in the same space.

5 CONCLUSION

The aim of the proposed method is to increase image resilience after combined serious attacks and being able to restore the original image even at a higher tampering rate.

Table 5: The capability of detection and recovery of the images after different tampering attacks.

Methods	Common attacks		Multiple attacks excluding rotation		Rotation attack with different angels		Multiple attacks including rotation	
	detect	recover	detect	recover	detect	recover	detect	recover
[12]	Yes	Yes	Yes	Yes	No	No	No	No
[14]	Yes	Yes	Yes	Yes	No	No	No	No
[11]4×4	Yes	Yes	Yes	Yes	No	No	No	No
[11]8×8	Yes	Yes	Yes	Yes	No	No	No	No
[11]	Yes	Yes	Yes	Yes	No	No	No	No
[8]	Yes	Yes	Yes	Yes	No	No	No	No
[5]	Yes	Yes	Yes	Yes	No	No	No	No
[9]	Yes	Yes	Yes	Yes	No	No	No	No
[1]	Yes	No	Yes	No	Yes	Yes	Yes	No
[9]	Yes	No	Yes	No	Yes	No	Yes	No
Proposed Method	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

To achieve this goal, firstly the navigation information was embedded inside the pixels which helped us to navigate the original direction of the image. Secondly, we proposed a new method of compression for achieving recovery reference that gives an opportunity to exploit the advantages of the similarities between pixels and blocks. Using this we able to extract another copy of recovery data, the reserved or backup data, which needs much less capacity for embedding using those similarities and extracting just their differences. Reserved recovery data will be used in case of damaging the first recovery data. The areas of embedding are selected specifically regarding the identified tampered areas in case of rotation attacks. These steps help to increase the robustness of the recovery code after tampering. To combat rotation attack, the recovery references belonging to the corner of the image are hidden inside the middle of the image (because inside the middle of the image is not affected by the rotation attack). The proposed hybrid method has obtained a better quality of recovered image in higher tampering rates, even after some combined attacks, including rotation, compared with other recent methods.

References

[1] Al-Maweri, N. A. A. S., Sabri, A. Q. M., and Mansoor, A. M. Automatic rotation recovery algorithm for accurate digital image and video watermarks extrac-

tion. *International Journal of Advanced Computer Science and Applications 7*, 11 (2016), 65–72.

[2] Al-Otum, H. M., and Ibrahim, M. Color image watermarking for content authentication and self-restoration applications based on a dual-domain approach. *Multimedia Tools and Applications*, 1–26.

[3] Ashtari, A. H., Nordin, M. J., and Kahaki, S. M. M. Double line image rotation. *IEEE Transactions on Image Processing 24*, 11 (2015), 3370–3385.

[4] Fan, M., and Wang, H. An enhanced fragile watermarking scheme to digital image protection and self-recovery. *Signal Processing: Image Communication 66* (2018), 19–29.

[5] Gul, E., and Ozturk, S. A novel pixel-wise authentication-based self-embedding fragile watermarking method. *Multimedia Systems* (2021), 1–15.

[6] Hong, W., Li, D., Lou, D.-C., Zhou, X., and Chang, C.-H. A bit toggling approach for ambtc tamper detection scheme with high image fidelity. *PloS one 15*, 4 (2020), e0230997.

[7] Hsu, C.-S., and Tu, S.-F. Image tamper detection and recovery using adaptive embedding rules. *Measurement 88* (2016), 287–296.

[8] Kim, C., and Yang, C.-N. Self-embedding fragile watermarking scheme to detect image tampering using ambtc and opap approaches. *Applied Sciences 11*, 3 (2021), 1146.

[9] Lei, Q., Xiao, L., Hosam, O., and Luo, H. A novel watermarking algorithm based on characteristics model of local fragmentary images. *International Journal of Embedded Systems 12*, 1 (2020), 11–21.

[10] Molina-Garcia, J., Garcia-Salgado, B. P., Ponomaryov, V., Reyes-Reyes, R., Sadovnychiy, S., and Cruz-Ramos, C. An effective fragile watermarking scheme for color image tampering detection and self-recovery. *Signal Processing: Image Communication 81* (2020), 115725.

[11] Qin, C., Ji, P., Chang, C.-C., Dong, J., and Sun, X. Non-uniform watermark sharing based on optimal iterative btc for image tampering recovery. *IEEE MultiMedia 25*, 3 (2018), 36–48.

[12] Qin, C., Ji, P., Zhang, X., Dong, J., and Wang, J. Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. *Signal processing 138* (2017), 280–293.

[13] Rakhmawati, L., Wirawan, W., and Suwadi, S. A recent survey of self-embedding fragile watermarking scheme for image authentication with recovery capability. *EURASIP Journal on Image and Video Processing 2019*, 1 (2019), 61.

[14] Shehab, A., Elhoseny, M., Muhammad, K., Sangaiah, A. K., Yang, P., Huang, H., and Hou, G. Secure and robust fragile watermarking scheme for medical images. *IEEE Access 6* (2018), 10269–10278.

[15] Sinhal, R., Ansari, I. A., and Ahn, C. W. Blind image watermarking for localization and restoration of color images. *IEEE Access 8* (2020), 200157–200169.

[16] Tohidi, F., and Paul, M. A new image watermarking scheme for efficient tamper detection, localization and recovery. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (2019), IEEE, pp. 19–24.

[17] Tohidi, F., Paul, M., and Hooshmandasl, M. R. Detection and recovery of higher tampered images using novel feature and compression strategy. *IEEE Access 9* (2021), 57510–57528.

[18] Wang, C., Zhang, H., and Zhou, X. Review on self-embedding fragile watermarking for image authentication and self-recovery. *Journal of Information Processing Systems 14*, 2 (2018).