Detection and Recovery of Higher Tampered Images Using Novel Feature and Compression Strategy

FARANAK TOHIDI[1,2], **MANORANJAN PAUL**[2], **(Senior Member, IEEE),**
AND MOHAMMAD REZA HOOSHMANDASL[1]
[1]Department of Computer Science, Yazd University, Yazd 89158-18411, Iran
[2]School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia

Corresponding author: Manoranjan Paul (mpaul@csu.edu.au)

ABSTRACT Due to the availability of powerful image-editing software and the growing amount of multimedia data that is transmitted via the Internet, integrity verifications and confidentiality of the data are becoming critical issues. However, currently, the accuracy of detecting and the recovery capability of the tampered images by the existing methods through watermarking strategy is still not at the required level, especially at a higher tampered rate. This paper proposes a new blind and fragile watermarking method to detect tampering and better recovery of tampered images. To improve the quality of both the watermarked and the recovered images, a new feature extraction scheme is introduced which will produce a short but comprehensive recovery code using a new compression strategy. If a block in the image tampers, the proposed embedded feature allows the original data to be extracted for recovery. To overcome tamper coincidence, every block's watermarked data contains not only the recovery code belonging to the block itself but also its neighbor's data as a second layer of recovery. Various size blocks were investigated to see the performance and compare their efficiency for recovering an image after different tampering rates. The test showed the smaller block sizes may be more suitable for locating tampering, where the bigger ones are more suitable when the tampering rate is higher. The bigger block sizes in the proposed method can recover an image even after a 60% tampering rate with high quality (more than 31 dB). The experimental results prove that the proposed method can have better efficiency for detecting tampering, and recovery of the original image, compared to the relevant existing methods.

INDEX TERMS Tampered image, image recovery, image authentication, feature extraction, watermarking, image compression.

I. INTRODUCTION

With the increased use of the Internet and the availability of signal processing technologies, integrity verification and protection of digitized information are becoming crucial issues. It can be even more significant when these images are applied in critical situations such as medical treatment or law courts [1]–[4]. Several digital signature bases have already been developed to overcome this problem in recent years. These methods can verify the integrity of the data by attaching a digital signature. However, the problem is that additional storage space is needed to attach the signature. Furthermore, digital signatures are often unable to locate

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate.

or recover the manipulations of the critical images [5]–[7]. A common method to deal with these two issues is watermarking [2] which has been developed to provide ownership authentication and integrity verification for digital media [8]–[11]. In this method, some information is inserted inside the original media signal in order to verify the credibility of the content or identify the ownership [12]–[16]. Restoring a tampered image to the original one is only possible if the basic features of the original data have been embedded inside itself called ''self-embedding'' [5].

Self-embedding watermarking means that some reference data, which hold the basic information related to the image, are generated and hidden into an image. If an image is tampered with, these data should then be extracted to recover the tampered areas [17]–[20]. The problem is that to ensure

the successful restoration of the original image, the reference code must include enough data. However, the reference code needs to be kept short, because the watermarked image quality should be preserved as larger embedded data distort the image quality by altering the original information. Therefore, the foremost problem in self-embedding watermarking is minimizing reference data to obtain a watermarked image with the highest quality possible. Simultaneously, a reference code of data should also contain enough information to achieve the highest quality recovered image. This is a challenging problem because most of the existing methods fail to detect and recover the original images especially when the tampering amount is larger. To solve the problem, a new hybrid compression scheme is introduced in this paper, which is based on various block-sizes, where we have encoded a sub-set of the original pixel intensities, and some derived information from each of them for better compression. We have followed a pattern of interpolation/extrapolation to derive the rest of the pixel intensities from the encoded data at both the sender and receiver ends. The selection of block size is critical as a large block size can conceal more recovery data, resulting in more ability to recover a good quality image in case of tampering. However, the capability of finding and locating tampered data in a precise region of the image is less accurate for larger block sizes. In this paper, we have investigated different block sizes and their corresponding patterns of interpolation/extrapolation and analyzed the performance of each, in terms of detection and recovering, at different tampering rates, with different types of images. We have found that despite being very compact, this new method of compressed data is capable of recovery of an original image with much better quality than other current methods.

A. INTRODUCING TAMPERING ATTACKS

To increase sensitivity to detect tampering, some fragile watermarking methods were introduced because watermarked data can be easily evidenced by any kind of alteration [21]–[24]. There are still some different attacks that are going to deactivate the sensitivity of the fragile watermarking. Therefore, a good fragile watermarking should be easily affected by any feasible attack. The various types of tampering or common attacks which may be imposed on a watermarked image are as follows [2], [25]–[28]:

1) GENERAL TAMPERING ATTACKS

Some items may be added, deleted, or altered on a watermarked image from another image, in order to generate a desirable, but false image. Copy-paste attack which is common also considered as a general attack that involves copying some area of an image and pasting it to the watermarked image [5]. But we looked at more professional attacks as follows:

2) COPY MOVE TAMPERING ATTACKS

These kinds of attacks copy a portion of a watermarked image afterward paste it somewhere else in the same watermarked

FIGURE 1. A sample of Copy Move Tampering attack.

(a) Original image   (b) Second image

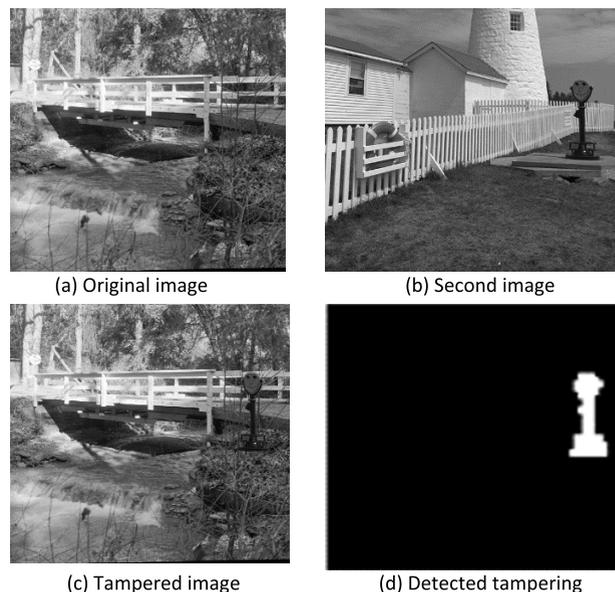(c) Tampered image   (d) Detected tampering

FIGURE 2. A sample of Collage Attack here is the insertion of this sign in exactly the same map area.

image to produce a fake image. Our results section of this paper shows the new method is effective against Copy-Move attacks. Fig. 1 shows an example of this attack.

3) PROTOCOL ATTACKS

Since most fragile watermarking methods use LSBs to hide data, there is one more attack that can deactivate watermarked data. The protocol attack or content-only attack deactivates authentication code, recovery code, or both by replacing LSBs with something else. However, our new method proves it is safe against this attack since there have been shifts by a secret key after embedding data in the proposed method.

4) COLLAGE ATTACKS AND VECTOR QUANTIZATION ATTACKS

vector quantization (VQ) and Collage attacks are two serious and problematic security counterfeiting attacks, which have similar structures to manipulate watermarked images. Both use watermarked images made by the same key.

VQ attacks copy a piece from a watermarked image, then paste it into the more desirable place in the destination

watermarked image. In collage attacks, a piece from the image which is watermarked will be copied in the same place in the second watermarked image. This means that the relative spatial locations are unchanged when a collage attack has happened. Fig. 2. illustrates how a collage attack can tamper with the original images. Our results show that our new method is also effective against these two attacks.

B. ADVANTAGES OF BLOCK-WISE AND PIXEL-WISE WATERMARKING

Targeting image authentication and recovery, there are two kinds of watermarking methods commonly used, known as block-wise and pixel-wise. A pixel-wise method generates data from the exact amounts of the pixels, then this generated data is embedded into the pixels themselves. However, in a block-wise method, firstly an image is divided into several blocks that contain a number of pixels, then watermarked data can be extracted and embedded for every block individually [2], [16].

On the one hand, block dependency can help the watermarking method to be more robust against some security attacks such as VQ or Collage attack [5], [7], [13]. On the other hand, pixel-wise recovery can lead to the higher visual quality of recovered tampered images.

To use the advantages of both schemes, the proposed hybrid method uses a block-wise procedure for authentication and a pixel-wise procedure for recovery in order to achieve higher accuracy of detecting tampering and improved recovered image quality. In the proposed method a refined block-based method has been used, where each block has its own watermark data embedded and at the same time each pixel is treated differently according to its position in the block during both feature extraction and recovery.

C. THE PROBLEM OF TAMPERING COINCIDENCE

Another Recovery issue has been tampering coincidence, which means that previous recovery methods have encountered the problem of losing the reference codes, due to damaging the area containing the recovery bits [5]. Actually, when the tampering rate exceeds 40%, consequently 40% of the watermark also will be completely lost and the quality of the recovered image is seriously affected. Embedding redundant reference codes has been tried by some researchers [15], [12], [29], [30] to overcome this issue. Having another copy of the reference code has been found helpful to increase the probability of successfully finding reference codes. However, inserting repeated data requires more space, leading to further distortion of the original image. Since most tampering rates are below 50%, obviously the probability of being unable to find the reference code as a result of tampering coincidence is less than half. However, the problem of finding the recovery code increases as the tampering area increases. To address this, we have taken advantage of the similarity between neighbor pixels and blocks to introduce a totally different logistic strategy for extracting and embedding data. The proposed method also allocated more space

for the main reference code, which is used more often and relatively less space was reserved for additional embedded features belonging to the block's neighbors, which could be used when tampering coincidence increases and the image cannot be recovered by the current methods. Thus, the proposed scheme has embedded additional essential information for recovery of the quality of an image after tampering coincidence but using less space and dual locations for embedding.

Therefore, we proposed a complete fragile, self-embedding, and block-based watermarking scheme for a wide range of tamper detection then recovery including more than 50% tampered images using a new compression strategy.

The contributions of this paper are summarized here:

- We proposed a new compression strategy where a sub-set of pixels is encoded only. We also proposed different interpolation/extrapolation patterns for different block sizes for deriving the non-encoded pixel intensities.
- We proposed a hybrid watermarking scheme by using a block-based procedure for authentication to detect different types of attacks and a pixel-wise procedure for recovery to make sure high-quality image recovery.
- We introduced different information and different sizes in each recovery code under the dual reference codes scheme to ensure can resist higher tampering rates. Therefore, it can be more robust in terms of recovery and achieve higher quality and better recovery.

The remaining part of this paper is arranged as below. Section 2 is a review of the current literature. The key stages of the proposed method will be described in Section 3; the experimental results and discussions will be outlined in Section 4; Section 5 concludes the paper.

II. BACKGROUND REVIEW

A watermarking method was proposed by Lee and Lin [12] in order to discover the tampered areas in an image and recovery the original one. In Lee's method, a tampered region is rebuilt from the data which are in all intact areas after the detection of modification. However, this method cannot be applied when there is a large modified region because it affects the quality of the recovered image. In addition, the quality of the image which is recovered depends on the location of tampering in each image.

Sing and Sing proposed a fragile watermarking method to detect modification in an image and recovery of the original one [16]. They used Discrete Cosine Transformation (DCT) to produce recovery data. Their scheme has a better quality of recovered image than the earlier methods because DCT was used to generate the recovery code. Other methods of recovery of a tampered image have been also developed using DCT or Discrete Wavelet Transformation (DWT) in transform domains [8], [26], [31]–[33]. These methods are very complex; therefore, they are limited when there is a need for real-time application. In addition, they were found to decrease the image quality [31].

Other fragile watermarking methods have been suggested using known methods of recovery, but improving the security of the image, such as Zhang *et al.* [17] using the non-linear chaotic sequences. In this method, DCT has been used to generate the image digest for any 2×2 blocks. Then the data is embedded into another block. Another method of watermarking based on a chaotic sequence has also been proposed by Tong *et al.* [24] improving security, but the quality is limited.

Dadkhah *et al.* [7] have used Singular Value Decomposition (SVD) to detect tampering in an image. In their block-wise scheme, encrypted data were produced then embedded in another block to increase the security of the watermarked image. Their method was shown to be successful against several security attacks, such as VQ attacks. However, the results of their method have shown that the amount of PSNR for the reconstructed image is limited to around 30 dB after collage attack when the tampering rate is 50%. A scheme of detecting and recovery of tampering for a medical image was developed by Shehab *et al.* [13] where an image is first divided into 4×4 blocks. Then the mean value and SVD of all blocks have been calculated to attain authentication and recovery codes. Both codes were hidden inside the Least Significant Bits (LSBs) related to pixels of the blocks. Arnold Transform [2], [13] is applied to scramble embedded watermarked data inside the image in order to improve security. This method can detect copy and paste attacks, content removal attacks, text addition attacks, and VQ attacks, but it has a problem with the accuracy of localization since both codes are embedded in the same block. The other problem of this method is that they used the trace of singular values, which is not enough for authentication and causes a big False Positive Rate (FPR) in detection. An AMBTC authentication method with efficient detection ability was proposed by Hong *et al.* in 2020 but this is not capable of recovering the original images [34].

There are some recent papers that claim that their methods can recover the original image, after tampering rates of more than 50% [22], [35], [36], but the required quality of their recovered images is not satisfied. For example, in [22], [35] the quality of their recovered images is less than 30dB even when the tampering rate is only around 30%. It is obvious for any method, the greater the tampering rate results in reduced quality for the recovered image, therefore the criteria are not met with their method beyond 30% tampering. Qin *et al.* [11] have developed a pixel-wise scheme of recovery with overlapping blocks. In their method, an image is divided into the 3×3 blocks in which every block consists of nine pixels and eight of them have been overlapped with its adjacent blocks. However, their method only detects the tampering intensity of below 45% of the image. Qin *et al.* [15] have also introduced a way to improve the quality of Block Truncation Coding (BTC) for compression of an image called Optimal Iterative Block Truncation Coding (OIBTC). They have defined new reconstruction levels for all blocks. They have used OIBTC to generate a reference code using two block sizes. In the case of a greater modification (up to 50% tampering) of an image,

a reconstructed image by a larger block size has better quality, as a result of having the redundancy of reference codes. However, the smaller block size has better performance in the lower tampering rates. These recent methods [15] of compression such as BTC, OIBTC, or Absolute Moment Block Truncation Coding (AMBTC) [9], [37] use two values for a block then generate a bit map based on those values for all pixels of the block. Later they can generate an approximation value from the encoded pixel's value. This way they get good compression of the image. However, it does not provide a high-quality image.

Some methods [38]–[42] used Interpolation-based schemes to increase quality, and also embed capacity for hidden data in an image. These schemes usually aim to achieve better rate-distortion efficiency. The existing methods that used some inter/extrapolation patterns have the advantages of having greater embedding capacity with less embedding distortion; however, they cannot recover the original images after tampering. Therefore, to address all of these problems we introduce a novel idea by exploiting interpolation/extrapolation patterns, to find a suitable recovery code that is capable of successfully recovering an image with greater quality. In addition, the proposed method also introduces dual reference codes with two different sizes, so that a good recovery is also possible at the tampering rate of more than 50%. The regions that are influenced by the tampering coincidence can be recovered using back up embedded data in every block's neighbors.

III. PROPOSED METHOD

In our proposed method, a fragile and also blind watermarking scheme is introduced in order to detect, localise tampering, then recovery of the original image. The proposed method involves two key stages that are described as follows:

On the sending side, the image is first divided into the $N \times N$ blocks with equal sizes so that watermark data which contains an authentication code and a recovery code will be derived for any block separately. The watermarked image will be obtained when authentication codes and recovery codes for all blocks are embedded. The procedure of producing watermark data on the sending side is illustrated in Fig. 3.

The authentication code is designed to find and locate the tampered region and is implanted in the image block itself. The authentication code is made by a *Hash function* [11] [15]. The first and second LSBs of any pixel intensity in the image will be exchanged with zero during computing authentication code. Since these bits of LSB will be used to embed watermark data inside them later and must not be measured initially.

The recovery code is designed to be able to reconstruct the tampered regions and we found it needs to be embedded into another block because it should be kept safe in case of tampering of the block. In order to have more security, the recovery code should also be scrambled into the image blocks. The destination block for embedding recovery code belonging to every block can be determined by the block
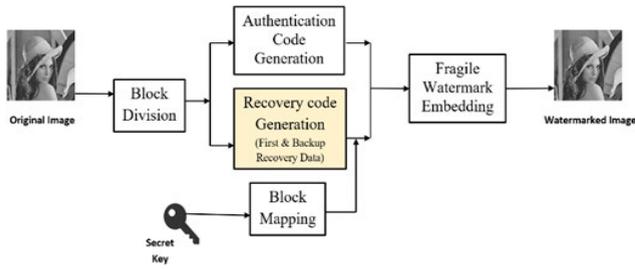
FIGURE 3. Block diagram of watermark data generation where the main contributing block is shaded.
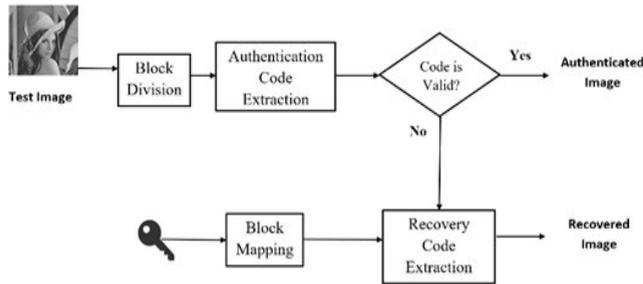

FIGURE 4. A Block diagram for the detection of Tampering and Recovery.

mapping sequence. Here, it is calculated through *Cat map transform* using secret keys [43], [44].

The recovery code is a very compact form of the image block including the first and the backup recovery data. We know that the recovery code should be efficient and compressed in a way that the watermarked image quality can remain high; in addition, the extracted information is capable of rebuilding the tampered image with superior quality simultaneously. Therefore, our proposed method describes a new compression strategy in detail, which can be used effectively for recovery code generation (shaded block in Fig. 3).

On the receiving side, to detect and locate any tampering, the authentication codes must be extracted firstly from all blocks then compared with each of those blocks' content for every block individually. If there is any mismatch between the extracted data and the regenerated authentication codes, those blocks should be tagged as tampered or modified blocks. If the image blocks are distinguished as tampered blocks, the relative recovery codes will be extracted from their mapped blocks using the previous secret keys to reconstruct the tampered areas. Our priority is using the first recovery data for the recovery of tampered blocks. Backup recovery data can be used in case of tampering coincidence. This process of detecting and recovery of an image on the receiving side is displayed in Fig. 4. The description of symbols that are used in this paper and the definitions of them have been listed in Table 1.

A. NEW COMPRESSION STRATEGY FOR RECOVERY CODES
To achieve greater quality for the reconstructed image and also a watermarked image, a new efficient image compression

TABLE 1. Main symbols used in this paper and their definitions.

P	Pixel
$\mid$	This pixel is computed by its two vertical neighbours
X	This pixel is computed by its four diagonal neighbours
$-$	This pixel is computed by its two horizontal neighbours
M	Main pixel that is shown by 6 bits
D	Difference between M and a pixel
β	block coefficient
m	The number of P for each M in a block
n	The number of M in a block
B	Backup data
Pl	Left neighbour pixel
Pr	Right neighbour pixel
Pu	Upper neighbour pixel
Pd	Down neighbour pixel
Plm	Middle pixel in the left neighbour 9×9 block
Prm	Middle pixel in the right neighbour 9×9 block
Pum	Middle pixel in the upper neighbour 9×9 block
Pdm	Middle pixel in the down neighbour 9×9 block
$B_{i,j}$	Block i,j
$W_{i,j}$	Watermark data of the block i,j
$C_{i,j}$	data produced from the Block ($B_{i,j}$)
$B_{Mi,Mj}$	Mapped block of $B_{i,j}$
S	Secret key
K	A key to shift pixels' bits
PI	Pixel Inside the block
PO	Pixel Outside the block
DR	extracted Differences (first) Recovery data
BR	extracted Backup Recovery data
Pr_{RF}	Probability of recovery for a block by first recovery data
Pr_{RB}	Probability of recovery for a block by backup recovery data
Pr_{NR}	The probability that a block cannot be recovered by watermarked data
α	Tampering rate

scheme is now introduced in the following two steps. These two steps will also be employed later for obtaining the recovery code. It should be mentioned that for the color images all the following instructions can be applied for each channel separately (e.g. RGB (3 × 8 bits)).

1) STEP ONE [FIRST COMPRESSION]
In this step, there are some pixels in the image which are eliminated in a way that they can be calculated later by their available neighboring pixels. This elimination can result in decreasing the total number of image pixels by at least one-quarter. Fig. 5 shows some of the pixels in a block of an image and illustrates how it can be possible to compress the image by decreasing the number of its total pixels so that unavailable

FIGURE 5. Generating the values of omitted pixels by using their available neighbors (Step One).

FIGURE 6. Calculating the values of Pixels by using differences with their neighbors(Step Two).

pixels will be reproduced later by the existing pixels. In Fig 5. The pixels of P_2, P_4, P_6, P_8 and P_5, for instance, are supposed to be eliminated in order to compress the image. These pixels can be calculated later by the other pixels which have been kept as follows:

•P_5: These pixels can be obtained later by the mean values of four Ps which exist around them. For example, $P_5 = (P_1 + P_3 + P_7 + P_9)/4$. Since these types of pixels are computed by their four diagonal neighbors, they are shown by the symbol of x.

•P_4 and P_6: These pixels can be obtained later by the mean values of two Ps located on the top and the bottom sides of them. For example: $P_4 = (P_1 + P_7)/2$ and $P_6 = (P_3 + P_9)/2$. Since these types of pixels are computed by their two vertical neighbors, they are shown by the sign of |.

•P_2 and P_8: These pixels can be obtained later by the mean values of two *Ps* located on the left and the right sides of them. For example: $P_2 = (P_1 + P_3)/2$ and $P_8 = (P_7 + P_9)/2$. Since these types of pixels are computed by their two horizontal neighbors, they are shown by the sign of −.

In step one, the initial stages of compression of an image are explained in such a way that recovery of 3×3 pixels is possible by having the values of four pixels only, instead of nine pixels. It should be mentioned that this compression leads to less decrease in the quality of the image than would be expected.

2) STEP TWO [FURTHER COMPRESSION OF DATA]

In this step, more compression can be made possible by making use of the amount of difference between near values of the pixels. Referring to Fig. 6, a pixel that is situated in the middle of the 5×5 block is defined as M_1. As shown obviously in Fig. 6, the pixel of M_1 is situated in the center of P_1 to P_8 and the distance between M_1 and any pixels of P_1 to P_8 are just 2 pixels.

Therefore, the differences between their values are often very small. For this reason, P_1 to P_8 can be identified just by a few bits of their differences with M_1 instead of real values of P_1 to P_8. There is also a coefficient according to

the different texture complexity for different blocks to avoid decreasing quality in the case of high texture blocks which will be defined later. The digit of 3 bits can be considered to be identified for different values separately. Any difference is shown by 2 bits to indicate how much it is, and 1 bit is also used for the sign. Therefore (the pixels from) P_1 to P_8 need 3 bits each for identifying the values of their differences with M_1 instead of 8 bits for each pixel.

When recovering the image, the real values for P_1 to P_8 can be computed simply by deducting or adding their different values with the value of M_1 individually. In Fig. 6, pixels that are located between every 5×5 pixels (in green areas) can also be calculated by the information around them. Therefore, the basic features for 6×6 pixels will be M (6 bits of its Most Significant Bits) and $8\times$ difference values (3 bits for each). This means that 36 pixels which need $36 \times 8 = 288$ bits in normal mode, can be compressed to only $6 + (8 \times 3) = 30$ bits required in the proposed compression scheme. Since compression rate is the ratio of the original data to the compressed data thus, here, the compression rate is 9.6. Fig. 6 shows how the values of *Ps* can be calculated with the help of *Ms*.

B. GENERATING WATERMARK DATA FOR EACH BLOCK

As mentioned before if watermark embedding does not involve any block dependency, it can easily be broken with particular attacks such as VQ attack [5], [7], [13]. To combat these attacks and also to have better performance of data watermarking, the images have been divided into blocks. Preparing recovery code is based on the new proposed compression strategy (described in section III. A). There is a number of steps to introduce a general algorithm of preparing watermark data for all different block sizes which are as follows:

• A hash function is used to calculate the authentication code. This function is nonreversible, and it is sensitive to

only one changed bit [11]. A hash function can convert a big number or string to a small integer. All pixels of each block and their ordering numbers must be used in the hash function to acquire the authentication code. Later, on the receiver side, this hash function will be applied again to determine if the content of the block is equal with that block on the sender side.

- Both steps of the new compression strategy are used to compressed data for every block size to acquire the recovery code. Such that at first some of the pixels in the block are selected to be removed (step 1) then depending on the size of the block there are some other pixels that are chosen to be kept (M_x). The rest of the pixels (P_x) according to their positions with the kept pixels (M_x) will be shown only by their differences (D_x) (step 2). Later, during the recovery process, all pixels can be restored only by knowing M_xs and D_xsM_xs and D_xs. Therefore, having just M_xs and D_xs as a recovery code is enough to restore a tampered block.
- Backup recovery data is defined for each block size to overcome tampering coincidence. Each block includes two kinds of recovery data in which the first recovery data belongs to itself and the second one is the backup recovery data belonging to its adjacent pixels or blocks depending on the size of the block.
- Different blocks may have different texture complexity, thus, treating all blocks equally may not be reasonable. Therefore, in order to have better quality for the reconstructed image, a coefficient should be defined for every single block in all block sizes. Here, the mean of differences between some special pixels for every block will be employed to find the value of β as a block coefficient in order to help with computing the better estimate of the value of Ps for each block separately. The amount of β is different from one block to another block depending on the complexity of each block. Five bits are allocated for the value of (β) as a block coefficient. This amount can be found by calculating the 5 most significant bits belonging to the average amount of differences. The value of β is defined as:

$$\beta = \sum_{i=1}^{i=n} \sum_{x=1}^{x=m} \left|(M_i - P_{i,x})\right| / mn \qquad (1)$$

$$\beta_t = \begin{cases} \beta_0 = 1 \, and \, \beta_t = 0, t = 1, \ldots, 4, \& \beta \leq 1 \& \\ floor \left[round(\frac{\beta}{2^{t+3}}) \right] mod \, 2, t = 0, 1, \ldots, 4, \beta > 1 \end{cases} \qquad (2)$$

The number of M_i is "n" and the number of P_i for each M_i is "m" depending on the size of the block. The floor function can give an output which is the nearest integer minus or equal to the input, the round function gives an output which is nearest to the input, and β_t ($t = 0, 1, \ldots, 4$) is five most significant bits of the coefficient of the current block.

- The first and the second LSBs of every pixel in all blocks have been selected to embed data therefore they must be changed to zero before any calculation.
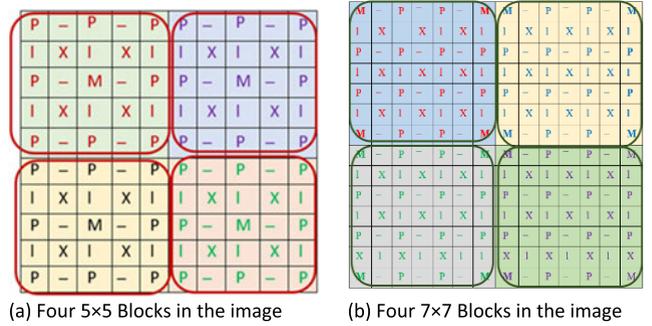

(a) Four 5×5 Blocks in the image   (b) Four 7×7 Blocks in the image

FIGURE 7. Different size of a block with its arranged pixels.

(a) Providing first recovery data including M_1, D_1 to D_8 and β (b) Providing Backup data (B_1 to B_4)
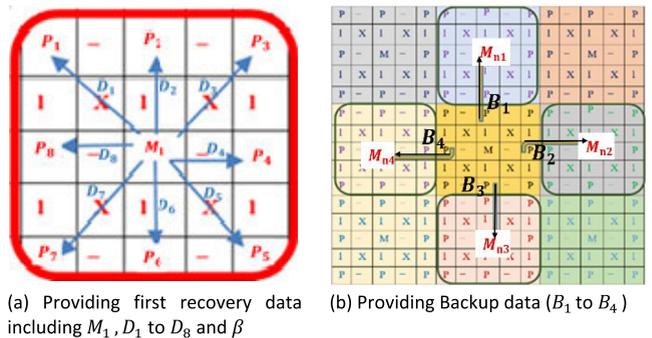
FIGURE 8. First and backup recovery data when the block size is 5 × 5.

Three sizes of blocks are being introduced to compare their efficiency for recovery of an image after different tampering rates. Fig. 7. has some examples of different block sizes to show the position of kept pixels (M_x) and those pixels (p_x) which are calculated by their differences (D_x). The rest have been eliminated and later can be found with the assist of M_x, D_x and β.

Fig. 8 (a and b)) illustrate how the first and backup recovery data are extracted when the block size is 5 × 5. Firstly, the capacity of data hiding for the block size of 5 × 5, using two *LSB*s is 50 bits. The number of allocation bits for the Authentication Code needs to be 7 bits. Thus, the rest of the capacity (43 bits) is reserved for the Recovery Code, including the first and backup information. The first Recovery Code consists of one kept pixel (M_1), which is situated in the middle of the 5 × 5 block (6 bits of MSB) and the value of β as the block coefficient, as well as the values of D_1 to D_8 (8 × 3 = 24 bits). Backup recovery data that belong to adjacent blocks include B_1 to B_4 (8 bits). When the block size is selected 7 × 7, the original image will be divided into blocks as Fig. 7.b shows. The capacity of data hiding for the block size of 7 × 7 using two LSBs is 98 bits, of which 9 bits are dedicated to the Authentication Code. Thus, the rest of the block capacity (89 bits) belongs to the recovery code including first and backup recovery data. To clarify what items the Recovery code consists of and how that can be achieved, there is an example of 7 × 7 block size in more detail, along with its formulas as follows:

1) The values of M_1 to M_4 (6 bits for each) (pixels which are situated at the corners of the block as Fig. 9 shows).
2) The value of (β) as a block coefficient (5 bits using 1 and 2).
3) The values of differences (D_{ix}) between M_i and P_{ix} (for every M there are three inside differences and for each difference 3 bits are allocated respectively) (as blue arrows are shown in Fig. 9, using 3 and 4). Where $D_{ix,t}(t = 0, 1, 2, i = 1, 2, \ldots, 4$ and $x = 1, 2, 3)$ denotes three bits of the difference between x_{th} selected P_i and M_i.

$$D_{ix,2} = \begin{cases} 1, & P_{i,x} < M_i \\ 0, & P_{i,x} \geq M_i \end{cases}$$
$$i = 1, 2, \ldots, 4, \ and \ x = 1, 2, 3 \qquad (3)$$

$$D_{ix,t} = \begin{cases} 00, & |P_{i,x} - M_i| < \beta/2 \\ 01, & \beta/2 \leq |P_{i,x} - M_i| < 3\beta/2 \\ 10, & 3\beta/2 \leq |P_{i,x} - M_i| < 5\beta/2 \\ 11, & 5\beta/2 \leq |P_{i,x} - M_i| \end{cases}$$
$$t = 0, 1, i = 1, 2, \ldots, 4, \ and \ x = 1, 2, 3 \quad (4)$$

4) The values of backup recovery data (B_{iy}) includes the differences between M_i and $Po_{i,y}$ (some outside pixels which are in the adjacent blocks) (Fig. 9, using 5 and 6). The number of 8×3 bits is allotted to adjacent pixels of every 7×7 block.

$$B_{iy,2} = \begin{cases} 1, & Po_{i,y} < M_i \\ 0, & Po_{i,y} \geq M_i \end{cases}$$
$$i = 1, 2, 3, 4 \ and \ y = 1, 2 \qquad (5)$$

$$B_{iy,t} = \begin{cases} 00, & |Po_{i,y} - M_i| < \beta/2 \\ 01, & \beta/2 \leq |Po_{i,y} - M_i| < 3\beta/2 \\ 10, & 3\beta/2 \leq |Po_{i,y} - M_i| < 5\beta/2 \\ 11, & 5\beta/2 \leq |Po_{i,y} - M_i| \end{cases}$$
$$t = 0, 1, i = 1, 2, 3, 4 \ and \ y = 1, 2 \qquad (6)$$

where $B_{iy,t}(t = 0, 1, 2 \ and \ i = 1, 2, 3, 4 \ and \ y = 1, 2)$ denotes three bits of the difference between y_{th} neighbor's pixel (two (y = 1,2) outside pixels which are situated in i_{th} adjacent block) with M_i of the current block.

Embedding more data for backup recovery information is feasible when the block size is 7×7 or bigger leading to increase quality for the reconstructed image after a greater tampering rate. There are 3 bits as backup recovery data related to some of the adjacent pixels of the block in which one of them shows whether this adjacent pixel is higher or lower than the M_i which is in the near corner of the block and the other bits indicate how many times of β its absolute value should be (as orange arrows show). The other pixels which are situated around this block (in Fig. 9 are colored yellow) can be computed by this information and the values of M (as green arrows in Fig. 9). In addition, to have a copy of every block as recovery data, the number of $14 \times 4 = 56$ pixels can be recovered by the backup data extracted from each block.
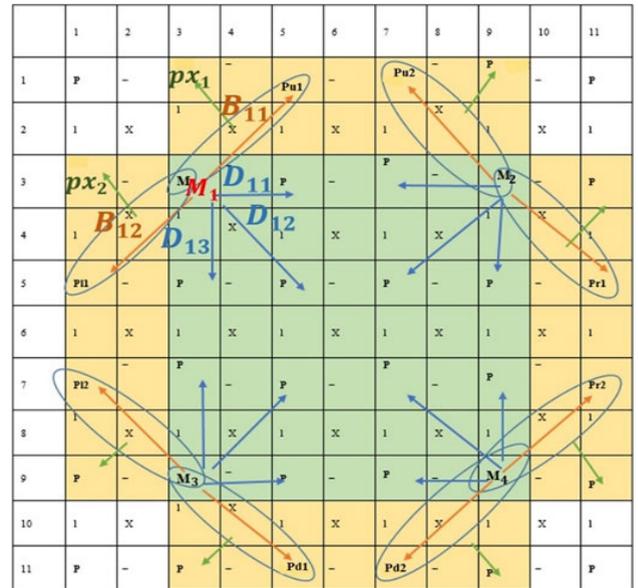

FIGURE 9. Providing the first and backup recovery data for recovery after tampering in block size 7 × 7. Pl1, Pl2, Pr1, Pr2, Pu1, Pu2, Pd1, Pd2 are adjacent pixels belonging to the left, right, up, and down neighbor blocks respectively. Each block (green area) not only can recover itself but also all pixels in yellow color can be recovered by its backup data.

Therefore, there are more than two copies of the compressed image embedded in the image itself. Consequently, the recovered image quality will be increased even if the tampering rate is high. Although more copies are embedded, watermarked image quality is still good because of embedding data in only 2 LSBs.

In Fig. 9, blue arrows show the difference between any of Ps which are inside the block and its nearest M. These values should be embedded as the first recovery data. Orange arrows show the difference between outside Ps with its nearest inside M. These values should be considered as backup data for recovery in case of damage to the first recovery data. Green Arrows also illustrate how some of the other outside Ps can be calculated by their nearby pixels. For example (7):

$$Px_1 = (Pu_1 + M_1)/2 \ and \ Px_2 = (Pl_1 + M_1)/2 \qquad (7)$$

The capacity of data hiding for the block size of 9×9 is 162 bits including 12 bits for the Authentication Code and 150 bits for the Recovery Code which contains first and backup information.

In Fig. 10 the pixels (Ps and Ms) which are inside the green area (a block size of 9×9) should be defined as the first recovery data. The pixels that are in orange regions or outside the block can be defined as backup data. The first recovery code for each block consists of M_1 to M_4, the value of (β) block, the values of differences between any of 21 inside Ps (P_1 to P_{21}), and their nearest Ms (blue arrows in Fig. 10).

For each block, there are some adjacent pixels or outside pixels that will be defined as backup data. There is a feasibility of recovering approximately two blocks by having the watermarked data belong to just one block in a block size

of 9 × 9. In Fig. 10, Pmd, Pmu, Pml, and Pmr are defined completely by 5 bits of their MSB bits since they are far from the pixels in this particular block. Some of the other outside pixels (Pu1, Pu3, Pu5, Pd1, Pd3, Pd5, Pl1, Pl3, Pl5, Pr1, Pr3, and Pr5) will be defined by 3 bits, by their differences with the nearest pixels of the block close to them. These nearest pixels are not necessary Ms and they can be calculated by any pixels (*Ps* or *Ms*) which is the nearest defined pixel to them. For example, the Orange Arrows in Fig. 10. The other Ps (Pu2, Pu4, Pd2, Pd4, Pl2, Pl4, Pr2, and Pr4) belonging to adjacent blocks can be calculated later by their adjacent Ps (as the Green Arrow show in Fig. 10). For example (8):

$$Pu2 = (P_2 + Pu1)/2 \text{ and } Pu4 = (P_4 + Pu5)/2 \quad (8)$$

C. EMBEDDING WATERMARK DATA

After merging the authentication code and recovery code, the data will be achieved for every block to be watermarked. As we have seen in the previous phases, watermarked data for each block is dependent on the content of that block and its adjacent blocks and it has been computed in a way that it only needs 2 LSBs to be hidden. To be secure when attacks happen, generated data can be encrypted by utilizing a secret key. Then it should be scrambled into the image's blocks by other secret keys using a chaotic map that can generate a mapping sequence. Embedded data should also be shifted by another secret key to be unpredictable. There are different secret keys of encryption according to the size of the block. The encryption is done by (9) then the mapped block should find through (10 and 11) to embed data. Afterward, there is a shift using the last secret key.

$$W_{i,j} = (S_1 \oplus C_{i,j}) \quad (9)$$

where $C_{i,j}$ denotes data that is produced from the $B_{i,j}$ block and $W_{i,j}$ is its watermark data. S_1 is the first secret key which is a 2 × N × N-bits. The watermarked data produced by (9) is 2 × N × N-bits encrypted data including encrypted authentication and recovery codes. Its authentication code should be embedded inside the block of $B_{i,j}$ $B_{i,j}$ but encrypted recovery code should be embedded inside the block of $B_{Mi,Mj}$ using (10).

$$B_{i,j} \rightarrow B_{Mi,Mj} \quad (10)$$

where $B_{Mi,Mj}$ is a mapped block for $B_{i,j}$ using cat map transform [39] [40] that can be calculated by (11).

$$Block_{Mi,Mj} \rightarrow \begin{cases} Mi = i + S_2 j \\ Mj = S_3 i + S_2 S_3 j + j \\ i, j = 1, 2, \ldots, m/n \end{cases} \quad (11)$$

Here, S_2 and S_3 are the other user secret keys. m/n is the number of blocks in the image The first and second LSBs of all pixels have been used to embed data. After embedding all watermarked data, there are shifts for all pixels' bits with the same key produced by (12) to avoid removing or substituting watermarked data by an attacker.

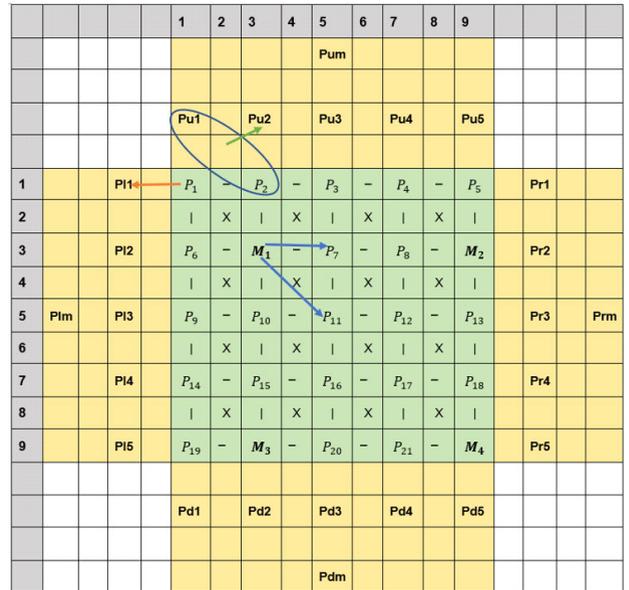$$k = S_4 \, mod \, 8 \quad (12)$$

FIGURE 10. First and backup recovery data for recovery after tampering for a Block size of 9 × 9.

where k denoted a key that is a positive integer less than 8 and it shows how many times the pixels' bits should be shifted and S_4 is the last secret key which is not divisible by 8.

D. TAMPER DETECTION AND RECOVERY

An image first should be divided into blocks on the receiver side and the size of the blocks is similar to the previous block size which has been used for extracting and embedding data on the sender side. There are reverse shifting depending on the previous key to find the exact order of bits in the pixels and extract data from two LSBs. Then decryption of extracted data should be done by (13).

$$C_{i,j} = (S_1 \oplus W_{i,j}) \quad (13)$$

The authentication code will be extracted from all blocks to compare with the blocks' content. Any block that its authentication code is not matched with the outcome of applying the hash function on its content will be tagged as a tampered block. After labeling all tampered blocks it turns to recover those blocks which labeled as tampered blocks.

In order to recover tampered blocks, the first step is discovering the block containing the recovery code using a cat map transform for tampered blocks. For example, if the block of $B_{i,j}$ has been tampered with, its first encrypted recovery data can be found in the block of $B_{Mi,Mj}$ using (11).

Then the block holding the recovery code $(B_{Mi,Mj})$ must be checked to avoid recovery by tampered data. For every tampered block if the block including its recovery code is intact, the first recovery code can be extracted and decrypted using the previous key to reconstruct the tampered block. Otherwise, we have to use the backup data which have also been watermarked in the
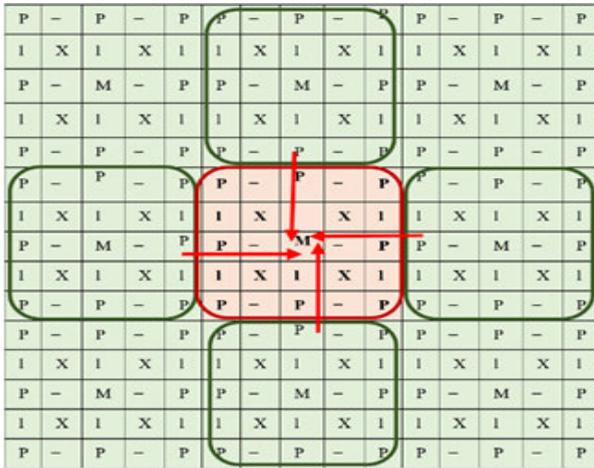
FIGURE 11. Recovery of a tampered block using backup recovery data when the first recovery data are lost for a block size of 5 × 5.

FIGURE 12. Recovery of a tampered block using backup recovery data when the first recovery data are lost for a block size of 7 × 7.

block containing its neighbors' recovery data including $\left(B_{M(i-1),Mj}\right),\left(B_{M(i+1),Mj}\right),\left(B_{Mi,M(j-1)}\right)$ and $\left(B_{Mi,M(j+1)}\right)$).

Since backup data are more compressed and deliver less quality than the first recovery data, it should be used less frequently and will be used only when the first recovery data have been damaged. It should be noted that before extracting any recovery data either first or backup data, the block containing those data should be investigated to ensure that it has not tampered. The first recovery data is dependent on the size of the block consisting of the amounts of Ms, the differences of Ms with Ps, and the block coefficient. These values can be achieved after extraction and decryption and they are able to reconstruct the block efficiently with high quality. At first, the amounts of Ps should be calculated by (14) then the other pixels can be found using these available pixels as was explained before, in the related section (III. A Step one).

In 14 formula, $DR_{x,t}t = 0, 1, 2$ and $x = 1, 2, 3, \ldots, k$ denotes three bits of differences as first recovery data for some selected inside pixels (PI_x) with their relative inside MI_x depending on the size of the block.

$$DR_{x,t} = \begin{cases} 000, & PI_x = MI_x \\ 001, & PI_x = MI_x + \beta \\ 010, & PI_x = MI_x + 2\beta \\ 011, & PI_x = MI_x + 3\beta \\ 100, & PI_x = M_x - \beta/2 \\ 101, & PI_x = MI_x - \beta \\ 110, & PI_x = MI_x - 2\beta \\ 111, & PI_x = MI_x - 3\beta \end{cases}$$
$$t = 0, 1, 2 \text{ and } x = 1, 2, 3, \ldots, k \quad (14)$$

1) RECOVERY OF TAMPERED BLOCKS WHICH HAVE LOST THEIR FIRST RECOVERY

In the proposed method, there are different ways of recovery using backup recovery data according to the size of the block. Fig. 11 illustrates how a 5 × 5 block can be recovered after
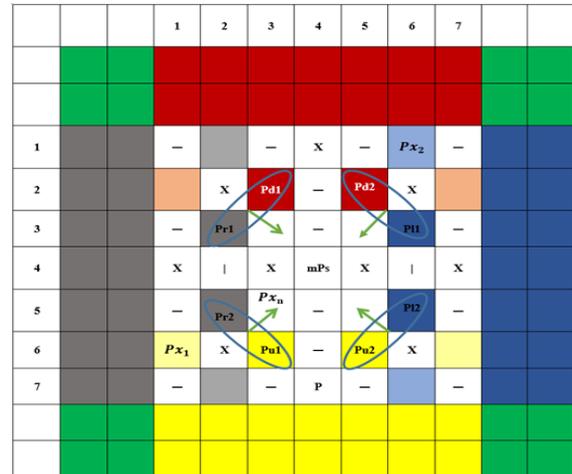
tampering when its first recovery data are also lost as a result of tampering coincidence. As shown in Fig. 11, the amount of M can be accessible by any of the recovery data related to its neighbor blocks using (15).

$$BR_{x,t} = \begin{cases} 00, & M = Pn_x \\ 01, & M = Pn_x + \beta n_x \\ 10, & M = Pn_x - \beta n_x/2 \\ 11, & M = Pn_x - \beta n_x \end{cases}$$
$$t = 0, 1 \text{ and } x = 1, 2, 3, 4 \quad (15)$$

where $BR_{x,t}(t = 0, 1$ and $x = 1, 2, 3, 4)$ denotes two bits of difference as backup recovery data for the middle pixel of the block (M) with their relative outside pixels Pn_x which are situated in their neighbors' watermarked data.

If there are more than one available and intact block containing the backup recovery code, the mean of the tampered block can be calculated through their average value. Restoring these blocks which had lost their first recovery data only by a single mean value may lead to poor quality. Their quality can be improved by replacing the value of any recovered pixels inside these blocks, with the mean of its value itself and its available neighbor values. It should be noted that the pixels that are situated around the edge of the block are replaced firstly then referring to those which are more inside the targeted block.

With this method, since backup recovery data are more comprehensive when the block size is 7 × 7 or bigger, the recovery of tampered blocks in case of tamper coincidence can be simpler and the recovered blocks have greater quality as well. Fig. 12 and Fig. 13 illustrate rebuilding a block size of 7 × 7 and a block size of 9 × 9 using backup recovery data. As can be observed from these figures, some pixels can be obtained simply from the recovery data related to adjacent blocks using (16). The other unavailable pixels will be computed with the assistance of these accessible pixels. In these two figures, having the same color for the
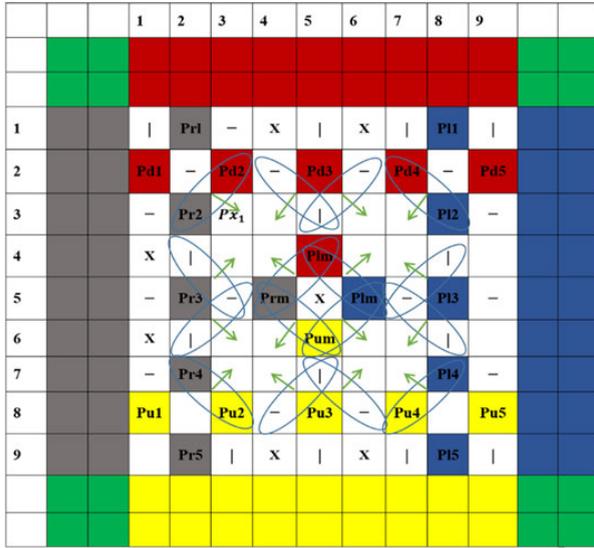
FIGURE 13. Recovery of a tampered block using backup recovery when the first recovery data are lost for a block size of 9 × 9.

FIGURE 14. (a) Original image, (b) watermarked image by 5 × 5, (c) watermarked image by 7 × 7, (d) watermarked image by 9 × 9.

pixels and blocks means that these pixels can be obtained from those outer blocks of the same color.

$$
BR_{x,t} = \begin{cases}
000, & P_x = Mn_x \\
001, & P_x = Mn_x + \beta n_x \\
010, & P_x = Mn_x + 2\beta n_x \\
011, & P_x = Mn_x + 3\beta n_x \\
100, & P_x = Mn_x - \beta n_x/2 \\
101, & P_x = Mn_x - \beta n_x \\
110, & P_x = Mn_x - 2\beta n_x \\
111, & P_x = Mn_x - 3\beta n_x \\
& t = 0, 1, 2 \ and \ x = 1, 2, 3, \ldots, k
\end{cases} \quad (16)
$$

where $BR_{x,t}$ $t = 0, 1, 2 \ and \ x = 1, 2, 3, \ldots, k$ denotes three bits of differences as backup recovery data for some selected inside pixels (P_x) with their relative outside Mn_x which are situated in their neighbors' watermarked data and k is depending on the size of the block. βn_x is the coefficient of the x_{th} block's neighbor.

Fig. 12 and Fig. 13 help to understand better with their corresponding figures Fig. 9 and Fig. 10. An example of achieving unavailable pixels by its adjacent accessible pixels can be the following example in Fig. 12: $Px_n = (Pu_1 + Pr2)/2$.

2) PROBABILITY OF DETECTING AND RECOVERY

According to the number of authentication bits that are allocated to any block size, the probability of a block being wrongly labeled or falsely detected is equal to 2^{-7}, 2^{-9}, and 2^{-12} for a block size of 5, 7 and 9 respectively. Since the recovery codes are scrambled inside the whole image by the secret keys, a chance of recovery of a block using its first or backup recovery codes or by its intact neighbor blocks depends on the rate of tampering. Therefore, when the tampering rate is low, most of the tampered blocks can

be recovered by the first recovery code. By increasing the rate of tampering the probability of the first recovery code being available being decreased, then the backup recovery code will be more engaged. The following formulas show the relationship between the rate of tampering and the rate of using different recovery data. When the tampering rate is α, it means that α is the percentage of the part of the image which is tampered with and needed to be recovered. If we assume that α is the ratio of the tampered block, then the probability that a block being recovered by the first recovery data (Pr_{RF}) can be calculated by (17).

$$
Pr_{RF} = 1 - \alpha \quad (17)
$$

where Pr_{RF} is the probability that a block is recovered by the first recovery data. The probability of recovery of a block by the backup recovery data (Pr_{RB}) can be computed by (18). This shows that the probability of recovery of a block when its first recovery data has been lost in case of tampering coincidence by its backup data.

$$
Pr_{RB} = \alpha(1 - \alpha) \quad (18)
$$

The probability that a block cannot be recovered by watermarked data (Pr_{NR}) can be found by (19).

$$
Pr_{NR} = \alpha^2 \quad (19)
$$

When both first and backup recovery data related to a block are damaged, recovery of this block can be possible by the nearest untampered block's mean value or their watermarked data. In other words, a recovered image can have approximately $(1 - \alpha)$ ratio of the intact region and $\alpha(1 - \alpha)$ ratio of recovered blocks by the first recovery data and $\alpha^2(1 - \alpha)$ ratio of recovered blocks by the backup data and α^3 ratio of unrecovered or recovered by their neighbor's mean value.

The above explanations can prove theoretically that the probability of detection is very good for all block sizes. Also, for lower tampering rates the block size of 5 is more suitable. In addition, in lower tampering rates the localization of tampering is more important, and the smaller size of the block will be more helpful. On the contrary, a block size of more than 5 can be used more beneficially when the tampering rate is high. The other reason for using a bigger block size in a higher tampering rate is that their backup recovery code is more comprehensive and can deliver a better quality of recovered blocks.
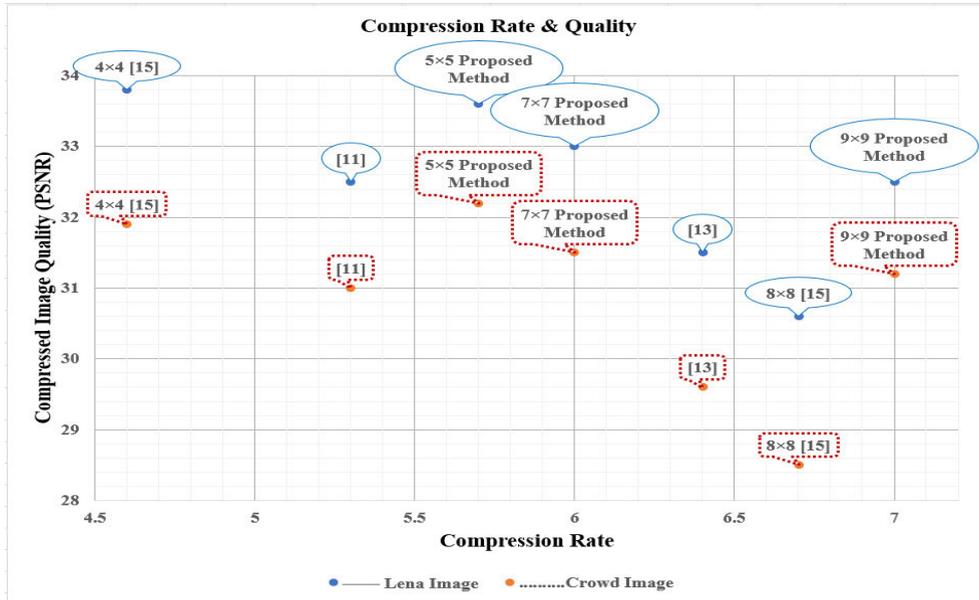
FIGURE 15. Comparison of the recovery data of the proposed method with recent other methods for the standard images of Lena and Crowd. As can be seen from the charted results the proposed method achieved better overall performance of compression than others.
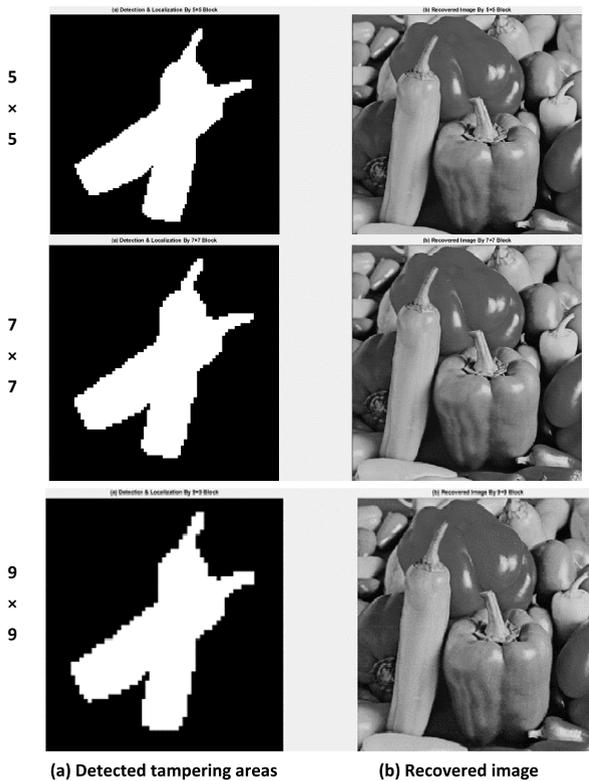

(a) Detected tampering areas   (b) Recovered image

FIGURE 16. Detecting and Recovering tampered image by 5 × 5, 7 × 7 and 9 × 9 Blocks respectively from top to bottom shows more precision in block size 5 × 5.

3) RECOVERY OF TAMPERED IMAGE

After recovery of all blocks that had been labeled as tampered blocks, all recovered blocks and all intact or untampered blocks should be merged to reconstruct the original image. To increase the quality [15] and having a better similarity with the original image, the first LSB should be changed to 0 and the second LSB should be changed to 1 for all pixels. This is the best option to minimize the value of distortion for every pixel since the distortion for every block is calculated by (20).

$$Distortion = \sum_{i=0}^{i=3}(Lsb_i - x)^2 \qquad (20)$$

This distortion caused by two LSBs and can be minimized when and $x = 10$ where Lsb_i is the original amount of the LSBs.

IV. EXPERIMENTAL RESULTS

To investigate the efficiency of the proposed method two distinct measurements and comparisons are presented in this section. We know that one of the main challenges in image tamper detection and recovery is minimizing the implanted recovery code to gain a higher quality of watermarked image, while also acquiring a good reconstruction of the tampered image. Therefore, the performance of the recovery code will be discussed first. Other measurements will also be discussed where our method has been affected by different kinds of attacks with various tampering rates. Several standard 512 × 512 digital library images: Splash, House, Bridge, Crowd, Pepper, Kiel, Lighthouse, and Lena are applied and have been imposed by different attacks to show the performance of the proposed method. Fig. 14 shows the original images and the same corresponding images after watermarking when the block size is 5, 7, and 9 separately. The watermarked image quality was found to be more than 44 dB on average, as a result of using only 2 LSBs.

A. PERFORMANCE OF THE RECOVERY CODE

Since the recovery code is usually a highly compressed version of the original image blocks, the length of the recovery
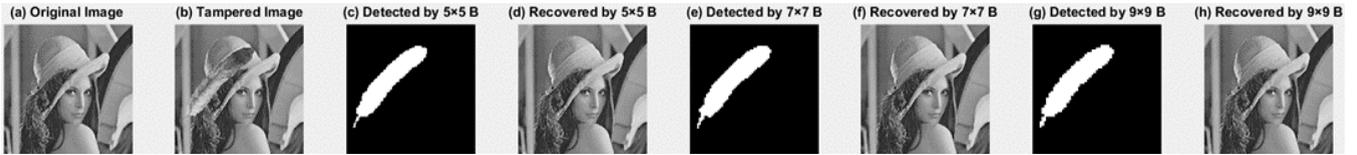
FIGURE 17. (a) Original image, (b) Copy and Paste attack from the outside of the image, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.
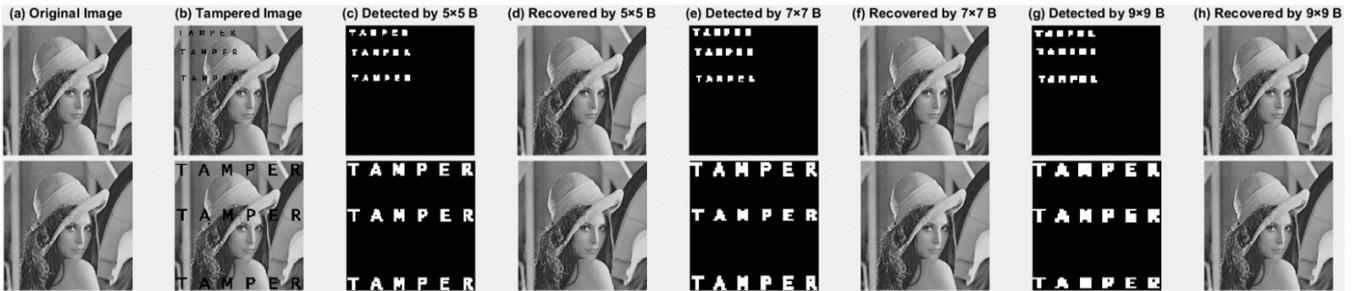

FIGURE 18. (a) Original image, (b) Content removal attack, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.
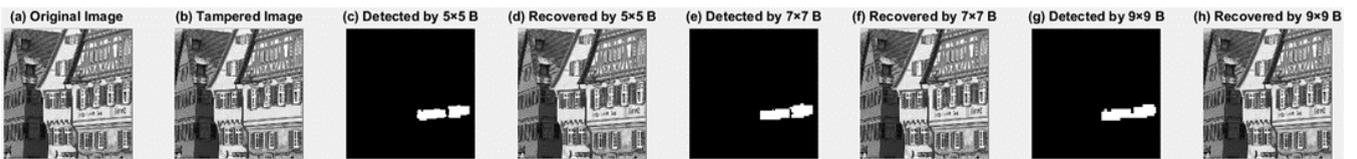

FIGURE 19. (a) Original image, (b) Copy and Paste attack from the inside and the outside of the image, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.

FIGURE 20. (a) Original image, (b) Text addition attack, (c and d) Detected and recovered image by 5 × 5B, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.

FIGURE 21. (a) Original Color Image, (b) Tampered Image, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.

code is directly proportional to the compression rate. Thus, the compression rate and quality of the decompressed version of the recovery code for all blocks were our first and second performance measurements to demonstrate that this proposed scheme can have a higher performance compared with other

recovery schemes already reviewed. Fig. 15 displays a chart of the compression rate and PSNR value related to the compressed images (Lena and Crowd Image) by the proposed method compared to some of the other recent methods where their recovery codes are produced in the spatial domain with
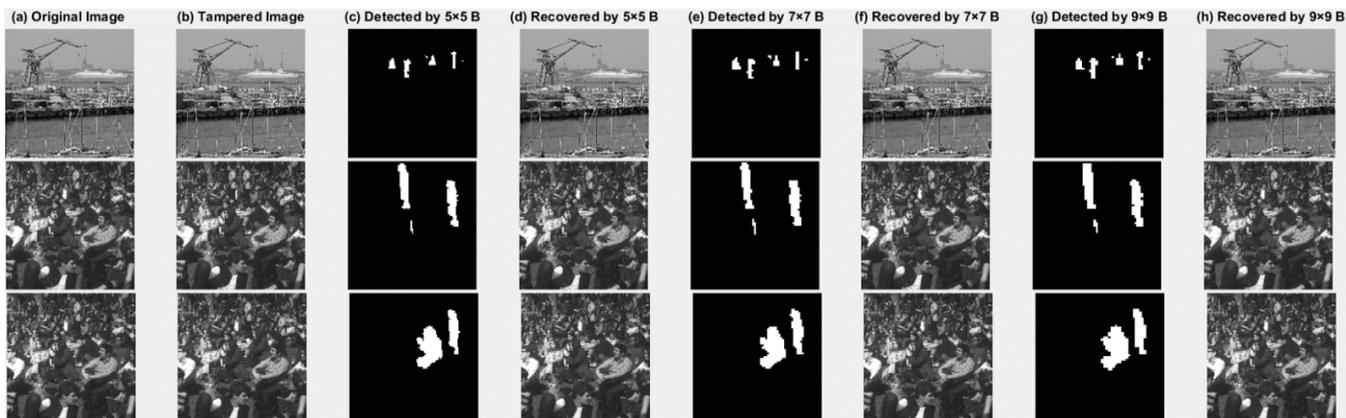
FIGURE 22. (a) Original images, (b) Copy Move attacks, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.
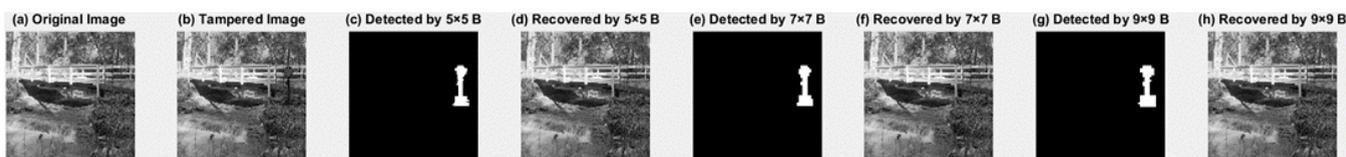

FIGURE 23. (a) Original image, (b) Collage attack, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.

FIGURE 24. (a) Original image, (b) Vector Quantization attack, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.
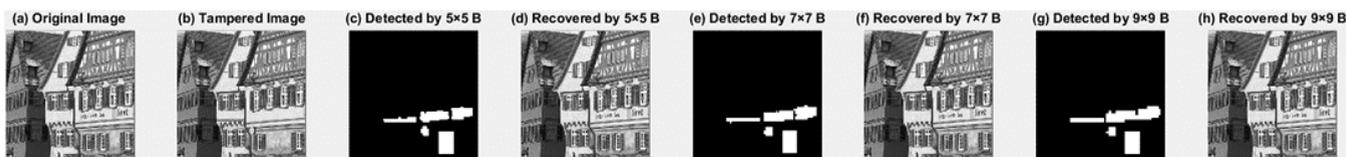

FIGURE 25. (a) Original image, (b) Multiple attacks, (c and d) Detected and recovered image by 5 × 5, (e and f) Detected and recovered image by 7 × 7, (g and h) Detected and recovered image by 9 × 9.

both good quality and high compression rate. It should be noted that the compression rate can be measured by bitrate in *bits per pixel* (bpp). However, in this paper, in order to compare different methods, the compression rate is considered the ratio of the original data to the compressed data. The chart in this figure shows the effectiveness of our method in terms of overall proficiency because both compression rate and the quality of compressed image are important factors for choosing recovery code. Since the compression rate by the proposed method was found to be better, embedding another recovery data into the image with similar quality of the watermarked image is now possible. It should be mentioned that in the proposed method for a block size of 7 or 9, compression

performance will get even better when the backup recovery data are also considered because each block also recovers some adjacent block pixels.

B. PERFORMANCE OF THE RECOVERED IMAGE AFTER DIFFERENT ATTACKS

In this section, the performance of our method is investigated after imposing different kinds of attacks which were explained in section I. A., Fig. 16. displays detection and recovery using block sizes of 5, 7, and 9, respectively. The watermarked images have been imposed by copy moved attack. This figure shows that all defined block sizes can

TABLE 2. Comparing performance of recent methods in term of recovered image quality after tampering.

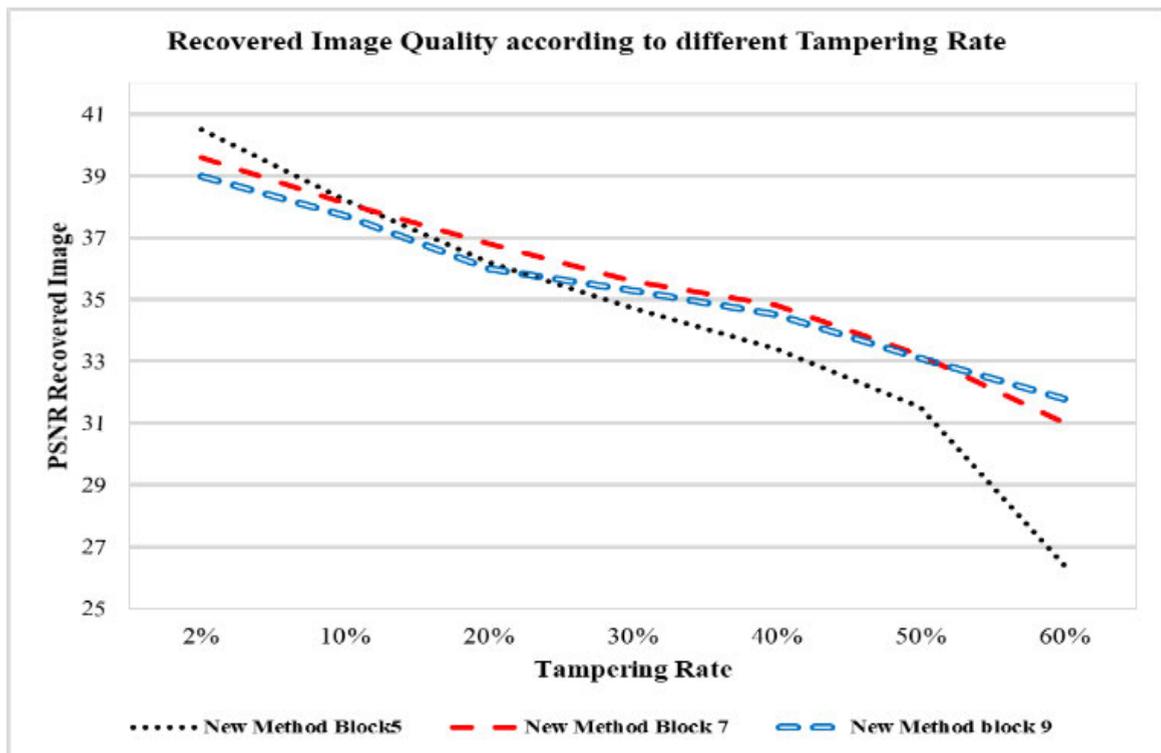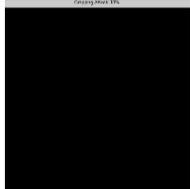| Methods	Average PSNR Of Watermarked Image	PSNR of Recovered Image [Min, Max]	Tampering Rate
4×4 Block (2018) [15]	44 dB	[33, 42]dB	$\alpha < 45\%$
8×8 Block (2018) [15]	44 dB	[31, 40]dB	$\alpha < 50\%$
(2017) [11]	42 dB	[29, 41]dB	$\alpha < 45\%$
(2018) [13]	44 dB	[29, 41]dB	$\alpha < 45\%$
2021[45]	40 dB	[36, 40]dB	$\alpha < 45\%$
2021[46]	38 dB	[32, 44]dB	$\alpha < 50\%$
5×5 Block [Proposed Method]	44 dB	[32, 42]dB	$\alpha < 55\%$
7×7 Block [Proposed Method]	44 dB	[31, 41]dB	$\alpha < 60\%$
9×9 Block [Proposed Method]	44 dB	[32, 40]dB	$\alpha < 60\%$

FIGURE 26. Recovered Image Quality according to different tampering rate.

detect and localize tampering but localizing in the smaller block sizes is more precise. Fig. 17 to 20 are results of the proposed method, after imposing some general attacks such as content removal, copy and paste from inside and outside the image, and text addition attacks. As the figures are shown, all block sizes which are affected by these kinds of attacks are able to detect and recover the image and the whole tampering is detected by several blocks. Concerning the detection of tampered areas, the smaller sizes of the block have better efficiency and they can detect tampered regions more detailed. This is more obvious when the area affected by the attack is tiny, see Fig. 18. Our method was also tested for several RGB color images (Splash, Lena, and pepper) and the results are shown in Fig. 21.

Fig. 22 to Fig. 25 show the results when images are imposed by more serious and problematic security counterfeiting attacks such as Collage attach, VQ attack, and also combining different attacks as Multiple attacks. As can be

TABLE 3. Visualized results for recovered image quality comparison of our proposed method with method [13]. The recovered images have been enlarged to show the quality is obviously higher, using our proposed method.

Original Image	Tampered Image Collage Attack $\alpha = 33\%$	Method in [13] Average PSNR=32	5×5 Block Proposed Method Average PSNR=35	7×7 Block Proposed Method Average PSNR=36	9×9 Block Proposed Method Average PSNR=36
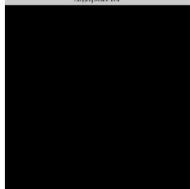
seen from these figures, all block sizes in our method also have been able to recover the original images, even when multiple attacks have been imposed. However, in terms of recovery different block sizes have different performances depending on the rate of tampering. There is a limitation, which is the rate of tampering.

Our method shows that the block sizes of 5 can recover original images, if the rate of tampering is below 50%, while the block sizes of 7 and 9 are more capable of recovery during higher tampering rates, i.e. up to 60%. Fig. 26 illustrates the PSNR curve of the average of recovered images, with respect to different tampering rates for previously mentioned Standard Images. As can be seen in this chart, larger block sizes are more efficient when the tampering rate is higher, although the block size of 5 is more suitable when the tampered region is low. This is because a block size of 5 has no access to good quality backup data when the demand for using the backup data is getting high during tampering coincidence. The strength of the proposed method is that any block's watermarked data include not only the recovery data of another block but also its neighbors' recovery data as a backup. Therefore, embedding every block's, and neighboring block's, data, results

in having better performance in the case of greater tampering rates when the probability of the first recovery code being available has been decreased because of tampering coincidence.

Better performance in comparison to other methods has been demonstrated in the results which are listed in Table 2. This table shows a comparison between the performance of the recent methods and our proposed method, in terms of comparing the quality of the watermarked image and the recovered image quality after different tampering rates on the previously mentioned standard images. [Min, Max] dB means the quality of the recovered image is between Max in best conditions (lower tampering rates) and Min dB in worst case tampering conditions. As Table 2. shows this algorithm is able to detect recovery from attacks with different and unpredictable sizes if the whole tampering rate in the image is below 50% or 60%. Thus, according to Table 2. our method can recover the original images with higher quality even in higher tampering rates comparing the current methods. Papers [45], [46] have achieved good quality for the recovered images but they have had to use three LSBs to achieve this. Therefore, the quality of their watermarked image is not high. Their methods are not also capable of recovery of the

TABLE 4. Visualized results for comparison with [15] after 48% tampering rates. (Our method can recover images even after 50% tampering). The recovered images have been enlarged to show the quality is obviously higher, using our proposed method.

Original Image	Tampered Image Collage Attack $\alpha = 48\%$	[15] Average PSNR=29.5	5×5 Block Proposed Method Average PSNR =31.5	7×7 Block Proposed Method Average PSNR =33.5	9×9 Block Proposed Method Average PSNR =33

image when the tampering rate is more than 45% and 50% respectively.

Since the existing method [13] was proposed for medical images, in Table 2. some results for medical images have been compared with our proposed method, not only to demonstrate that our method can also work on medical images but also to show some clear visual comparison of the efficiency of the two methods. It can be clearly seen from Table 3 in the enlarged images that the pixelation is much less with the proposed method. This can prove that our recovered image after 33% tampering has achieved better quality than [13].

In the recent methods which are listed in Table 2, only [15] has proposed a method that can recover the original image after higher tampering rates up to 50% with high quality of the watermarked image. To demonstrate that our method has also better performed compared to this method [15], some visual comparisons with this method were given in Table 4. In both visual comparison tables, there have been enlarged and zoomed photos to demonstrate visually that our method has achieved better quality in comparison with those methods.

Therefore, this paper presents a new combined pixel-wise and block-wise method to restore an image after high tampering rates which is applicable to a range of digital images including medical images. Successful recovery of images after problematic tampering attacks became possible by introducing a new method of image compression. Creating a strong recovery code using this new method of compression, also embedding neighboring block information as backup data to increase the chance of recovery of the image in case of tampering coincidence, results in a higher quality of recovery image even after high tampering rate compared to existing ones.

V. CONCLUSION

While it has previously been shown that hidden reference data inside an image is a way to restore an image after tampering, finding and generating efficient, but very short reference data that is capable of recovering an image with high quality is really challenging. Therefore, our hybrid method outlined in this paper has introduced a new way of compression for extracting the basic features of an image. This new

compression method has been proven to be able to achieve an efficient reference code for the recovery of a tampered image, even for different block sizes using a new extra/interpolation pattern. ∗ The proposed hybrid method uses a block-wise procedure for authentication and a pixel-wise procedure for recovery. ∗ This leads to both detecting multiple attacks more accurately and improving recovered image quality. ∗ Since the resultant recovery code is very compact, an increased chance of recovery is also provided in backup data to overcome high tampering coincidence without noticeably losing the quality of the watermarked image. ∗ Experimental results have shown that our designed hybrid method is able to detect and restore the tampered regions of an image, following a range of different attacks, for all three introduced block sizes. ∗ The smaller block sizes have proven to be more suitable when the tampering rate is low since it can localize tampering more precisely, and therefore deliver higher recovered image visual quality. ∗ The bigger block sizes can deliver better-reconstructed image quality because of having more precise backup data when the areas of tampering are large (more than 50%).

Achieving an efficient recovery code that can be implanted in images, which is firstly more compact, secondly more accurate, is possible using a novel features extraction and compression strategy, therefore delivering a much better quality of the recovered original image than was possible before, even after multiple attacks and in high tampering rates. This has been our important contribution supported by the experimental research results of the proposed method outlined in this paper. The proposed method can recover higher quality images even at close to 55% or a higher tampering rate compared with the existing methods. However, like any other existing method, it also does not provide good quality at a further higher tampering rate. In the future, we would like to find the solution for achieving a high-quality recovered image at higher rates. Moreover, we will try to reduce the size of the authentication and recovery codes without sacrificing the quality of the image recovery quality.

REFERENCES

[1] Y. Xiang, D. Xiao, H. Wang, and X. Li, "A secure image tampering detection and self-recovery scheme using POB number system over cloud," *Signal Process.*, vol. 162, pp. 282–295, Sep. 2019.

[2] L. Rakhmawati, W. Wirawan, and S. Suwadi, "A recent survey of self-embedding fragile watermarking scheme for image authentication with recovery capability," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, p. 61, Dec. 2019, doi: 10.1186/s13640-019-0462-3.

[3] B. B. Haghighi, A. H. Taherinia, and A. H. Mohajerzadeh, "TRLG: Fragile blind quad watermarking for image tamper detection and recovery by providing compact digests with optimized quality using LWT and GA," *Inf. Sci.*, vol. 486, pp. 204–230, Jun. 2019.

[4] C.-F. Lee, J.-J. Shen, Z.-R. Chen, and S. Agrawal, "Self-embedding authentication watermarking with effective tampered location detection and high-quality image recovery," *Sensors*, vol. 19, no. 10, p. 2267, May 2019.

[5] C. Wang, H. Zhang, and X. Zhou, "Review on self-embedding fragile watermarking for image authentication and self-recovery," *J. Inf. Process. Syst.*, vol. 14, no. 2, pp. 510–522, 2018, doi: 10.3745/JIPS.02.0082.

[6] M. Fan and H. Wang, "An enhanced fragile watermarking scheme to digital image protection and self-recovery," *Signal Process., Image Commun.*, vol. 66, pp. 19–29, Aug. 2018.

[7] S. Dadkhah, A. A. Manaf, Y. Hori, A. E. Hassanien, and S. Sadeghi, "An effective SVD-based image tampering detection and self-recovery using active watermarking," *Signal Process., Image Commun.*, vol. 29, no. 10, pp. 1197–1210, Nov. 2014.

[8] C.-S. Hsu and S.-F. Tu, "Image tamper detection and recovery using adaptive embedding rules," *Measurement*, vol. 88, pp. 287–296, Jun. 2016.

[9] C.-C. Lin, Y. Huang, and W.-L. Tai, "A novel hybrid image authentication scheme based on absolute moment block truncation coding," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 463–488, Jan. 2017.

[10] D. Singh and S. K. Singh, "DCT based efficient fragile watermarking scheme for image authentication and restoration," *Multimedia Tools Appl.*, vol. 76, no. 1, pp. 953–977, Jan. 2017.

[11] C. Qin, P. Ji, X. Zhang, J. Dong, and J. Wang, "Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy," *Signal Process.*, vol. 138, pp. 280–293, Sep. 2017.

[12] T.-Y. Lee and S. D. Lin, "Dual watermark for image tamper detection and recovery," *Pattern Recognit.*, vol. 41, no. 11, pp. 3497–3506, Nov. 2008.

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

[14] F. N. Thakkar and V. K. Srivastava, "A blind medical image watermarking: DWT-SVD based robust and secure approach for telemedicine applications," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 3669–3697, Feb. 2017.

[15] C. Qin, P. Ji, C.-C. Chang, J. Dong, and X. Sun, "Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery," *IEEE MultimediaMag.*, vol. 25, no. 3, pp. 36–48, Jul. 2018.

[16] D. Singh and S. K. Singh, "Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 775–789, Jul. 2016.

[17] J. Zhang, Q. Zhang, and H. Lv, "A novel image tamper localization and recovery algorithm based on watermarking technology," *Optik*, vol. 124, no. 23, pp. 6367–6371, Dec. 2013, doi: 10.1016/j.ijleo.2013.05.040.

[18] W. Sun, Z.-M. Lu, Y.-C. Wen, F.-X. Yu, and R.-J. Shen, "High performance reversible data hiding for block truncation coding compressed images," *Signal, Image Video Process.*, vol. 7, no. 2, pp. 297–306, Mar. 2013.

[19] C.-C. Chang, T.-S. Chen, Y.-K. Wang, and Y. Liu, "A reversible data hiding scheme based on absolute moment block truncation coding compression using exclusive OR operator," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 9039–9053, Apr. 2018.

[20] W.-L. Tai and Z.-J. Liao, "Image self-recovery with watermark self-embedding," *Signal Process., Image Commun.*, vol. 65, pp. 11–25, Jul. 2018.

[21] C. Qin, C.-C. Chang, and K.-N. Chen, "Adaptive self-recovery for tampered images based on VQ indexing and inpainting," *Signal Process.*, vol. 93, no. 4, pp. 933–946, Apr. 2013.

[22] J. Molina-Garcia, B. P. Garcia-Salgado, V. Ponomaryov, R. Reyes-Reyes, S. Sadovnychiy, and C. Cruz-Ramos, "An effective fragile watermarking scheme for color image tampering detection and self-recovery," *Signal Process., Image Commun.*, vol. 81, Feb. 2020, Art. no. 115725.

[23] C. Qin, X. Chen, D. Ye, J. Wang, and X. Sun, "A novel image hashing scheme with perceptual robustness using block truncation coding," *Inf. Sci.*, vols. 361–362, pp. 84–99, Sep. 2016.

[24] X. Tong, Y. Liu, M. Zhang, and Y. Chen, "A novel chaos-based fragile watermarking for image tampering detection and self-recovery," *Signal Process., Image Commun.*, vol. 28, no. 3, pp. 301–308, Mar. 2013.

[25] L. Rosales-Roldan, M. Cedillo-Hernandez, M. Nakano-Miyatake, H. Perez-Meana, and B. Kurkoski, "Watermarking-based image authentication with recovery capability using halftoning technique," *Signal Process., Image Commun.*, vol. 28, no. 1, pp. 69–83, Jan. 2013.

[26] C. Qin, C.-C. Chang, and P.-Y. Chen, "Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism," *Signal Process.*, vol. 92, no. 4, pp. 1137–1150, Apr. 2012.

[27] Y. Huo, H. He, and F. Chen, "Alterable-capacity fragile watermarking scheme with restoration capability," *Opt. Commun.*, vol. 285, no. 7, pp. 1759–1766, Apr. 2012.

[28] M. Hamid and C. Wang, "Adaptive image self-recovery based on feature extraction in the DCT domain," *IEEE Access*, vol. 6, pp. 67156–67165, 2018, doi: 10.1109/ACCESS.2018.2879404.

[29] B. B. Haghighi, A. H. Taherinia, and A. Harati, "TRLH: Fragile and blind dual watermarking for image tamper detection and self-recovery based on lifting wavelet transform and halftoning technique," *J. Vis. Commun. Image Represent.*, vol. 50, pp. 49–64, Jan. 2018.

[30] K. Sreenivas and V. Kamakshiprasad, "Improved image tamper localisation using chaotic maps and self-recovery," *J. Vis. Commun. Image Represent.*, vol. 49, pp. 164–176, Nov. 2017.

[31] A. Azeroual and K. Afdel, "Real-time image tamper localization based on fragile watermarking and faber-schauder wavelet," *AEU Int. J. Electron. Commun.*, vol. 79, pp. 207–218, Sep. 2017.

[32] R. O. Preda, "Semi-fragile watermarking for image authentication with sensitive tamper localization in the wavelet domain," *Measurement*, vol. 46, no. 1, pp. 367–373, Jan. 2013.

[33] C. Qin and X. Zhang, "Effective reversible data hiding in encrypted image with privacy protection for image content," *J. Vis. Commun. Image Represent.*, vol. 31, pp. 154–164, Aug. 2015.

[34] W. Hong, D. Li, D.-C. Lou, X. Zhou, and C.-H. Chang, "A bit toggling approach for AMBTC tamper detection scheme with high image fidelity," *PLoS ONE*, vol. 15, no. 4, Apr. 2020, Art. no. e0230997.

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

[36] H. M. Al-Otum and M. Ibrahim, "Color image watermarking for content authentication and self-restoration applications based on a dual-domain approach," *Multimedia Tools Appl.*, vol. 50, pp. 1–26, Jan. 2021, doi: 10.1007/s11042-020-10368-9.

[37] W. Hong, X. Zhou, and D.-C. Lou, "A recoverable AMBTC authentication scheme using similarity embedding strategy," *PLoS ONE*, vol. 14, no. 2, Feb. 2019, Art. no. e0212802.

[38] M. A. Wahed and H. Nyeem, "High capacity reversible data hiding with interpolation and adaptive embedding," *PLoS ONE*, vol. 14, no. 3, Mar. 2019, Art. no. e0212093, doi: 10.1371/journal.pone.0212093.

[39] X.-T. Wang, C.-C. Chang, T.-S. Nguyen, and M.-C. Li, "Reversible data hiding for high quality images exploiting interpolation and direction order mechanism," *Digit. Signal Process.*, vol. 23, no. 2, pp. 569–577, Mar. 2013.

[40] A. Malik, G. Sikka, and H. K. Verma, "An image interpolation based reversible data hiding scheme using pixel value adjusting feature," *Multimedia Tools Appl.*, vol. 76, no. 11, pp. 13025–13046, Jun. 2017.

[41] K.-H. Jung, "A survey of interpolation-based reversible data hiding methods," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 7795–7810, Apr. 2018.

[42] X. Zhang, Z. Sun, Z. Tang, C. Yu, and X. Wang, "High capacity data hiding based on interpolated image," *Multimedia Tools Appl.*, vol. 76, no. 7, pp. 9195–9218, Apr. 2017.

[43] H. Liu, B. Zhao, and L. Huang, "Quantum image encryption scheme using Arnold transform and S-box scrambling," *Entropy*, vol. 21, no. 4, p. 343, Mar. 2019, doi: 10.3390/e21040343.

[44] L. Sun, J. Xu, X. Zhang, and Y. Tian, "An image watermarking scheme using Arnold transform and fuzzy smooth support vector machine," *Math. Problems Eng.*, vol. 2015, Apr. 2015, Art. no. 931672, doi: 10.1155/2015/931672.

[45] C. Kim and C.-N. Yang, "Self-embedding fragile watermarking scheme to detect image tampering using AMBTC and OPAP approaches," *Appl. Sci.*, vol. 11, no. 3, p. 1146, Jan. 2021, doi: 10.3390/app11031146.

[46] E. Gul and S. Ozturk, "A novel pixel-wise authentication-based self-embedding fragile watermarking method," *Multimedia Syst.*, vol. 120, pp. 1–15, Feb. 2021, doi: 10.1007/s00530-021-00751-3.

• • •