Block-Wise Authentication and Recovery Scheme for Medical Images Focusing on Content Complexity

Faranak Tohidi[1,2,3], Manoranjan Paul[3](✉), Mohammad Reza Hooshmandasl[1,2], Subrata Chakraborty[4,5], and Biswajeet Pradhan[4,5]

[1] The Laboratory of Quantum Information Processing, Yazd University, Yazd, Iran
ftohidi@stu.yazd.ac.ir, hooshmandasl@yazd.ac.ir

[2] Department of Computer Science, Yazd University, Yazd, Iran

[3] School of Computing and Mathematics, Charles Sturt University, Bathurst, Australia
mpaul@csu.edu.au

[4] School of Information, Systems and Modelling, University of Technology Sydney, Ultimo, Australia
{Subrata.Chakraborty,Biswajeet.Pradhan}@uts.edu.au

[5] Centre for Advanced Modelling and Geospatial Information Systems, University of Technology Sydney, Ultimo, Australia

Abstract. Digital images are used to transfer most critical data in areas like medical, research, business, military, etc. The images transfer takes place over an unsecured Internet network. Therefore, there is a need for reliable security and protection for these sensitive images. Medical images play an important role in the field of Telemedicine and Tele surgery. Thus, before making any diagnostic decisions and treatments, the authenticity and the integrity of the received medical images need to be verified to avoid misdiagnosis. This paper proposes a block-wise and blind fragile watermarking mechanism for medical image authentication and recovery. By eliminating embedded insignificant data and considering different content complexity for each block during feature extraction and recovery, the capacity of data embedding without loss of quality is increased. This new embedding watermark method can embed a copy of the compressed image inside itself as a watermark to increase the recovered image quality. In our proposed hybrid scheme, the block features are utilized to improve the efficiency of data concealing for authentication and reduce tampering. Therefore, the scheme can achieve better results in terms of the recovered image quality and greater tampering protection, compared with the current schemes.

Keywords: Medical images · Image authentication · Watermarking · Tamper detection · Image recovery · Medical image security

1 Introduction

Today, Patients who live in remote areas are able to be diagnosed by experts with the help of telemedicine. However, this advancement of technology has led to some serious

J. J. Dabrowski et al. (Eds.): PSIVT 2019 Workshops, LNCS 11994, pp. 86–99, 2020.
https://doi.org/10.1007/978-3-030-39770-8_7

security concerns. Medical imagery transmission between experts and patients is an important task in this regard. During the data transfer, this data may be altered by some attacks intentionally or unintentionally. Thus, the authentication and content verification of this kind of important digital data is essential. Furthermore, recent studies show that, due to the rise of software capabilities for editing and modifying digital images, manipulation of radiography data is a serious issue; For example, modified images could be used in illegal claim for medical insurance of a patient or in publishing fraudulent results. The present security frameworks are either using encryption or steganography, or the combination of both to protect against unauthorized access. While these image encryptions are useful for protection against unauthorized access, they are unable to safeguard the authenticity and integrity of the transmitted image when the key is revealed. Furthermore, these methods are not able to reconstruct the original image when it is attacked. It is obvious that integrity and confidentiality are the main issues, because damaging of the medical image during transmission leads to serious problems in medical treatments, like the damage of decisive information, misdiagnosis by physicians and potentially calling into question the reliability of the health care center [1, 2].

Due to the high sensitivity to the modification in some images such as medical imagery, fragile watermarking schemes can be used where authentication is required. Fragile watermarking could be considered as two main groups: pixel-based and block-based schemes. In the pixel-based fragile watermarking approaches, the data pertaining to the watermark is produced utilising the host pixel values. These are then embedded into the host pixels as well. In case of the block-based fragile watermarking approaches, the host image is first segmented into multiple blocks. Each block contains individual data for the watermark, which can be used for authentication through detection and verification of the watermark data. If detection of the watermark data is unsuccessful, it indicates that the image may have been changed. Subsequently the block is then marked as tampered or invalid. From the embedding point of view, watermarking can be categorized into frequency or spatial based. The frequency-based approaches apply various transfer functions such as the *Fast Fourier Transformation* (FFT), *Discrete Cosine Transformation* (DCT), and *Discrete Wavelet Transformation* (DWT) to change pixel values from spatial domain to the coefficients of the frequency domain. Then the watermark data is hidden into those. But the spatial domain uses the pixel values to embed the watermark data directly. Spatial domain usually embed hidden data in the *Least Significant Bits* (LSBs) of pixels' value in order to avoid damaging the image [3–10].

1.1 Related Work

Self-embedding fragile watermarking can be useful in order to identify and then recover after any tampering. In this method the watermark data can be a copy of the compressed image or features of original image itself. The basic features of an image which are chosen as watermark data should include enough information to recover the original image, with higher recovery in the tamper region. A dual watermarking method has been proposed by Lee and Lin to detect tampering within an image and then to recover the original image [5]. In their method tampered area can be recovered by extracting watermarked data from the other intact blocks. This method is appropriate for minor tampering cases only.

Some of watermarking methods suffer from false image production after recovery by using a reference table, because of the block autonomous aspect of image watermarking. Those kinds of watermarking that have not involved any block dependency may be damaged with some special attacks like *Vector Quantization* (VQ) attack [6]. To overcome VQ attack, some block-wise watermarking methods are introduced, such as a fragile watermarking method for verifying and recovering medical images [6]. An image needs to be segmented into same size blocks in order to compute authentication and recovery codes by their method. Singular value decomposition is applied to attain a block authentication code for every 4 × 4 block. The recovery code is the mean value of every 2 × 2 block. Arnold transform is applied to distinguish where these codes should be embedded but embedding both codes in the same block can cause an increase in the rate of false detection. A blind image watermarking method utilising the DWT and the *Singular Value Decomposition* (SVD) has been developed by Thakkar and Srivastava [7]. They used DWT on selecting the region of interest in medical images and produced separate frequency sub-bands for decomposition of these areas. Then the results are combined by the applying SVD on the LL sub-band. Their method is robust and has produced good results in terms of watermarked image quality and in extracting watermarked data successfully, but it is not capable of recovering the medical images when it is altered.

Qin et al. [8] developed a new scheme of compressing the image, named as *Optimal Iterative Block Truncation coding* (OIBTC), which achieved better quality than the traditional *Block Truncation Coding* (BTC). They applied OIBTC to achieve recovery. They have used 4 × 4 block size and 8 × 8 block size. In higher tampering rates, the quality of a recovered image by bigger block size is higher because of more redundancy of the recovery code but in lower tampering rate the block size of 4 × 4 has higher performance, since the recovery code has not been so compressed. In most of the block-wise methods, an image is segmented into the same sized blocks and all blocks are treated equally. It is obvious that the volume of data that can be concealed in a block is limited by the size of the block. A big block size can convey more data, leading to more recovery data. But the ability of detecting and locating of the exact area is less.

Therefore, the size of block can be an important option to have efficient authentication and recovery since there is a trade-off between the size of the block and effective authentication and recovery. In addition, the features of a block can be exploited to enhance the efficiency of data concealing and authentication. It may be better to encode recovery data related to the blocks with small changes and fewer bits. Instead recovery data of the blocks with big changes could be encoded by more bits to boost the quality of the recovered image. This could mean a bigger capacity to hide the recovery data of the smooth blocks is pointless. This capacity can be reserved for hiding the recovery data of more complex blocks. In the proposed method, the complexity of the block has been used to understand the types of the blocks to design different plans of embedding and extracting data to increase the efficiency of authentication and recovery. In the other word, some blocks do not need much capacity for embedding their features, and their dedicated capacities can then be used for other purposes.

2 Proposed Method

The first step for self-embedding watermarking is obtaining the basic features from the image, then embedding this data into the image itself. Thus, an image can be recovered after tampering by extracting and using the watermarked data from intact areas of the image. On one hand, since the data is embedded into the image as watermarked data, the amount of this data should be as minimum as possible so to minimize the decrease in the watermarked image quality. On the contrary, if the amount of data entrenched into the image is larger, the recovered image will be of better quality. Therefore, there is a trade-off between the watermarked and recovered images in terms of their quality. To address this problem and have high quality for both the watermarked and the recovered images, the following steps should be considered: firstly, the selected data as watermarked data should be as efficient as possible, so that watermarked data is able to recover the tampered image with higher quality. Secondly, watermarked data should be as compressed as possible so that embedding them as watermark data into the image decreases the original image quality as little as possible.

To achieve this aim, a new hybrid method for compressing and obtaining the efficient features of an image will be introduced. This method discovers and pinpoints modifications in an image and recovers the altered areas. The information hidden in the image or the watermark data are divided into authentication code and recovery code, leading to greater accuracy. The authentication code is used to identify and trace the regions of tampered areas, and the recovery code can be used in case of tampering to recover the original image. In some cases, not only some areas of the image are destroyed but also their recovery codes may have been lost as well as a result of tampering. Therefore, these regions cannot be salvaged, and the quality of the recovered image will decrease. For this reason, as well as obtaining a better quality of a recovered image, two different copies of a compressed image will be embedded into the original image as the watermark data.

Three kinds of the watermark data should be provided for every block of size 8 × 8. The first kind of watermark data is named as the authentication code (16 bits) which can be used to identify the tampered blocks, the second and third kinds of watermark data are recovery codes, which are applied for recovery of the damaged content of the tampered image. The authentication code is entrenched inside the block itself and the recovery codes are entrenched into the mapped block of the image in order to have block dependency and being able to deal with the VQ attack. Due to the fact that replacing only two LSBs of pixels in image may not decrease the quality of the image noticeably, these two LSBs in all blocks are reserved for embedding data. Recovery codes can be achieved with the help of OIBTC and average pixels values of the block.

The Block Truncation Coding (BTC) is an effective image compressing algorithm. In this algorithm an original image with size n × n should be divided into m × m non-overlapping blocks. The average value (μ) and the standard deviation (σ) will be calculated for every block using (1, 2):

$$\mu = \frac{1}{m}\sum\nolimits_{i=1}^{m} x_i \tag{1}$$

$$\sigma = \sqrt{\frac{1}{m}\sum\nolimits_{i=1}^{m} (xi - \mu)^2} \tag{2}$$

All pixels in the block are categorized into two sets, in a way that when the intensity of a pixel is more than the mean value of the block, it is considered as the first set. Otherwise, it belongs to the other set. There is a bit map for every block as well. The corresponding bit for the pixels of the first set are zeros and for the second set pixels are ones. Any block in the image can be compressed by following above steps. Then an image block will be decompressed by substituting the ones with high reconstruction level (M_1) and the zeros by low reconstruction level (M_0) using the following Eqs. (3, 4) [11–16].

$$M_0 = \mu - \sigma\sqrt{\frac{m^+}{m^-}} \tag{3}$$

$$M_1 = \mu + \sigma\sqrt{\frac{m^-}{m^+}} \tag{4}$$

Where $m+$ is the number of pixels for which their values are greater than μ and m− is the number of pixels that are less than. To improve the visual quality of BTC-decompressed image, [8] has proposed a new OIBTC algorithm for compressing an image. In OIBTC new low and high reconstruction levels have been introduced as M_l and M_h, which can be calculated by minimizing the distortion for each block through following steps:

1. Every block is arranged in ascending order of its pixels' values, i.e.,

   $$\mathrm{S} = \{p_1, p_2, \ldots, p_m\}$$

   In which p_i are the pixels in the block and $p_1 < p_2 < \ldots < p_m$
2. Each block should be divided into two segments, and for each segment the mean value should be calculated as

   $$S_l^k = \{p_1, p_2, \ldots, p_k\}, S_h^k = \{p_{k+1}, p_{k+2}, \ldots, p_m\}$$

   In which S_l^k and S_h^k are these two segments.
3. In each block, the mean values of the two above sets $\left(M_l^k \; and \; M_h^k\right)$ are considered as low and high reconstruction levels and the distortion should be computed for the block by (5):

   $$d^k = d_l^k + d_h^k = \sum\nolimits_{i=1}^{k} (p_i - M_l^k)^2 + \sum\nolimits_{i=k+1}^{i=m} (p_i - M_h^k)^2 \tag{5}$$

   The distortion for the whole block is d^k while $d_l^k \; and \; d_h^k$ are distortion for each segment and p_i are the real amount of pixels in the block.
4. Steps 2 and 3 should be repeated to obtain minimum distortion. Where the distortion is minimum, $M_l^k \; and \; M_h^k$ can be used as the low and high reconstruction levels $(M_l \; and \; M_h)$ of the block.

After generating the recovery codes (it will be introduced in Sect. 2.1 and 2.2), these codes should be embedded in other blocks. Arnold transformation can be applied as a mapping function to find the suitable block for embedding the recovery codes. Using this function helps with distributing the recovery data into different blocks. A digital image is partitioned into blocks and each block has the address of (x, y). Arnold transform maps one block to another block using (6).

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & K_1 \\ K_2 & K_1K_2+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} modN \tag{6}$$

Where “N” is the number of all blocks in the image. K_1 and K_2 are used as keys. The embedding locations of two recovery codes of each block are different and are calculated by two keys.

2.1 Producing Authentication and Recovery Data

The first and the second LSBs of all pixels should be replaced with zero during the process of authentication code calculation, since LSBs will be substituted with watermarked data and must not be assessed. The authentication code for each block is 16 bits and can be generated through a Hash function. All 64 pixels which are inside the 8 × 8 block and the ordering numbers of them should be included in the hash function. The authentication code is then included in the block itself.

To obtain the recovery code, a distortion criteria D has been used to select which option of compression is more suitable for each block (unlike as presented in [8]). Each block has been treated differently regarding its complexity in our work. Some blocks do not need as much capacity to embed their features. These blocks are considered as smooth blocks. But some other blocks need more capacity to embed their features as they are more complex or textured. Since every smooth block can be recovered by less information, their dedicated locations can be reserved for embedding another copy related to the other blocks. For every 8 × 8 block these following four compression methods are available to choose in order (methods are arranged in order of descending compression rates):

1. An average pixels values of the 8 × 8 block
2. Four average pixels values related to four 4 × 4 blocks inside the 8 × 8 block
3. An 8 × 8 OIBTC compression
4. Four 4 × 4 OIBTC compression related to four 4 × 4 blocks inside the 8 × 8 block

In order to efficiently exploit the available capacity and to embed more data, as well as having a high-quality watermarked image, a threshold for distortion should be set. Each block should have its own limitation to extract its basic features depending on its content complexity. Hence any of the four compression methods above whose distortion is less than the distortion threshold level and having greater compression rate, should be applied for selecting the first recovery data. Thus, the option that presents the highest compression rate is the priority if its calculated distortion is less than the threshold. These kinds of blocks are very smooth and the first copy in this case is just the mean value of the 8 × 8 block. Otherwise, the distortion should be calculated for the second option in a way that the block should be divided into four 4 × 4 blocks. The average mean value for each 4 × 4 block and their distortion should be calculated and if their total distortion is not less than threshold as well, the next option is our next priority using a similar procedure. The last priority is four 4 × 4 OIBTC which may be selected when the block is quite complex.

The value of threshold can be selected according to the complexity of the image and predicted tampering rate. If the threshold is selected at a lower level the distortion for the first copy will be low. Consequently the quality of the recovered block by the first copy will be high. But it should be considered that in the higher tampering rate because of high probability of losing the first copy, we have to use the backup recovery data therefore reasonable quality for the second copy is also important. Hence enough room should be created for better backup recovery as well. In this work, in order to find the suitable threshold, a copy of the compressed image by 8 × 8 OIBTC should be calculated then average value of distortion for all 8 × 8 blocks in the image can be set as a threshold. Two bits are also allocated as indicators to demonstrate which compression method has been used. The distortion is calculated by (7) for each 8 × 8 block.

$$\mathrm{D} = \sum\nolimits_{i=1}^{i=8} \sum\nolimits_{j=1}^{j=8} (p_{i,j} - c_{i,j})^2 \tag{7}$$

Where D denotes the distortion for each 8 × 8 block, $p_{i,j}$, and $c_{i,j}$ are the original pixel value and the value of pixel after compression.

2.2 Reducing the Number of Bits for Embedding

Reducing the number of bits which are needed to embed as watermark data is possible by exploiting the differences between nearby values. Since any of M_l and M_h (low and high reconstruction levels in OIBTC compression) can be displayed by 6 bits separately and both belong to the same image block, 10 bits should be sufficient for both. Here 6 bits are required for the mean values of M_l and M_h and 4 bits for the absolute difference between their mean values and any value of M_l or M_h. Instead of real values of M_l and M_h the mean value and the absolute difference value can be embedded. Then in the receiver side, the real values for M_l and M_h can be calculated conveniently by subtracting and adding the difference value with the mean value separately. Hence, it is not required to embed all 12 bits for every block and more capacity will be remaining to embed more useful data (unlike [8]).

2.3 Watermark Embedding Process

Every 8 × 8 block has 64 pixels which watermarked data is embedded in 2 LSBs of these pixels. The 16 bits of the LSBs are earmarked for authentication purposes. Two bits of the LSBs are dedicated for distinguishing which compression method has been done. The rest of the LSBs (which are 110 bits) are reserved for recovery purposes including the first and backup recovery codes. After embedding the first copy with the help of reduced bit numbers, and considering texture of every block, there are still spaces for embedding the other copy for each block. It should be mentioned that, the type of the other copy is dependent on the first copy and how much capacity is still available for embedding more data. The total capacity in each block for embedding data is restricted to 128 bits to be able to have high quality watermarked image. The vacant capacity to embed the second copy can be calculated by considering the occupied capacity that has been used by the first copy. In this way one of the embedded copies will have better quality and the other one is more compressed in every block. Thus to efficiently use the remaining capacity of the block, there are four options as follows:

- First copy is 8 × 8 OIBTC compression, second copy should be four average pixels value of four 4 × 4 blocks.
- First copy is four 4 × 4 OIBTC compression, second copy should be an average pixels value of 8 × 8 block.
- First copy is an average pixels value of 8 × 8 block, second copy should be four 4 × 4 OIBTC compression.
- First copy is four average pixels value of four 4 × 4 blocks, second copy should be 8 × 8 OIBTC compression.

2.4 Detecting and Localizing Tampered Area

For detection of tampering and trace the location of tampered area, the image is divided into 8 × 8 blocks and the 2 LSBs of all pixels are replaced with zeros. For each of the blocks the information associated with the current block should be supplied into the hash function. Clearly all the 64 pixels which are inside the 8 × 8 block and the ordering numbers of them should be included in the hash function. The obtained authentication code from each block is compared with the amount of Hash function related to that block to recognize if the block is tampered. If this information is not identical it shows that the block has been tampered with. Since hash function is sensitive to even a one bit change of input, any modification will be detected for every block. If tampering is detected, extraction of the recovery code from destination blocks is required.

2.5 Recovery of Tampered Image

If a block is detected as tampered by comparing its authentication code with the content, it can be recovered by extracting the recovery information from the intact areas of the image. Recovery data include first and backup recovery data. As the probability of losing first recovery data related to a tampered block, there is a second opportunity to recover the tampered block with the assistance of the backup recovery data. In case of tampering, the addresses of destinations for the first recovery data and the backup recovery data can be calculated by the reverse of Arnold transformation with previous keys. Then the other authentication checks should be done to ensure that the blocks that contained the first and backup data are still intact. If both are intact in regard to the indicator bits, the copy which is more detailed will be chosen for obtaining better results. Otherwise any of the copies which is available and intact can be used. If both copies had been tampered with, the recovery of the block is done with the help of mean values of their obtainable undamaged neighbouring blocks. Through the above steps and decompression of the relevant tampered blocks pixels could be recovered. Then by combining the intact blocks and the recovered blocks the recovered image can be reconstructed.

3 Experimental Results

Performance evaluation of our proposed scheme has been conducted on the watermarked image quality and recovered image quality. The experiment has been conducted on some standard 512 × 512 images when tampering rates (t) were below 50% and the results are shown in Table 1. The watermarked image quality is more than 43 dB for all images. The quality of recovered images has been compared with the watermarked image quality with two standard quality measurements (The Structural SIMilarity (SSIM) and Peak Signal-to-Noise Ratio (PSNR)). Figures 1 2, 3 and 4 show the results of encoding. In these figures, three encoded images are presented to demonstrate that in the proposed hybrid method some useless data has been eliminated during preparation of data for the first copy in order to make room for embedding one more but different copy as backup recovery data. As it can clearly be seen in the figures, more textured blocks have more data to embed, but the dedicated capacity for a smooth block has been used by embedding one more complete backup copy related to another block. Smooth blocks in hybrid method encoded figures are shown white.

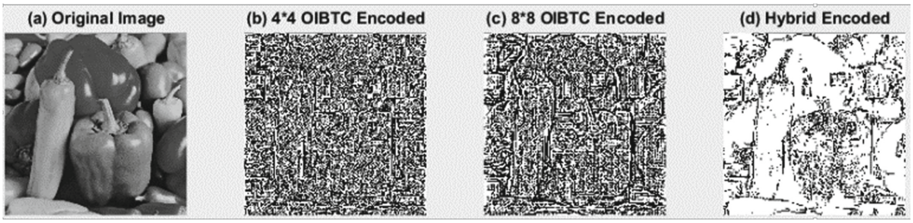

Fig. 1. (a) Pepper image, (b) OIBTC (4 × 4) encoded, (c) OIBTC (8 × 8) encoded, (d) Proposed Hybrid Scheme encoded for the first copy

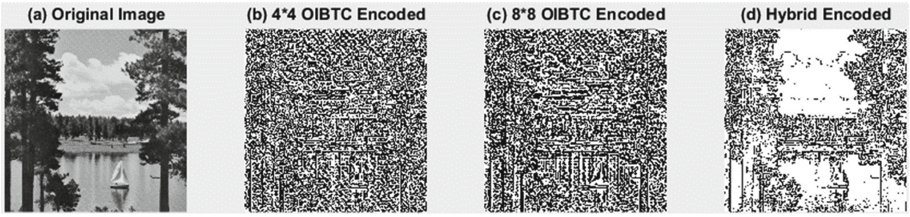

Fig. 2. (a) Lake Image, (b) OIBTC (4 × 4) encoded, (c) OIBTC (8 × 8) encoded, (d) Proposed Hybrid Scheme encoded for the first copy

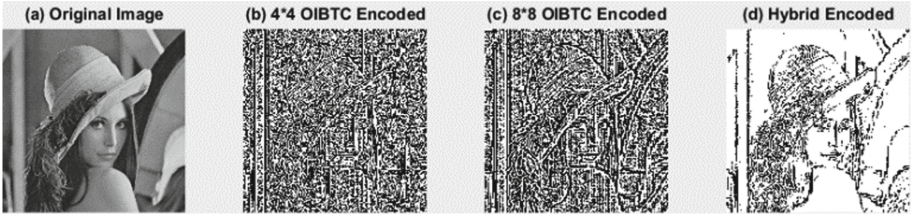

Fig. 3. (a) Lena image, (b) OIBTC (4 × 4) encoded, (c) OIBTC (8 × 8) encoded, (d) Proposed Hybrid Scheme encoded for the first copy

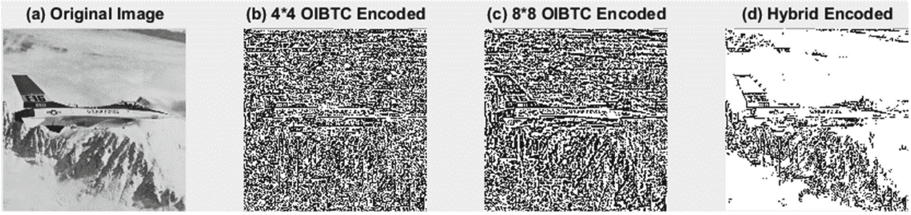

Fig. 4. (a) Plane image, (b) OIBTC (4 × 4) encoded, (c) OIBTC (8 × 8) encoded, (d) Proposed Hybrid Scheme encoded for the first copy

Figures 5 and 6 show the results of tampering detection, localization and recovery by the proposed hybrid method. The 512 × 512 standard medical images are included in our figure results also since the proposed method can work on medical images as well. Figure 7 shows the results of watermarking on the original medical image and the results of recovery after tampering using the proposed method.

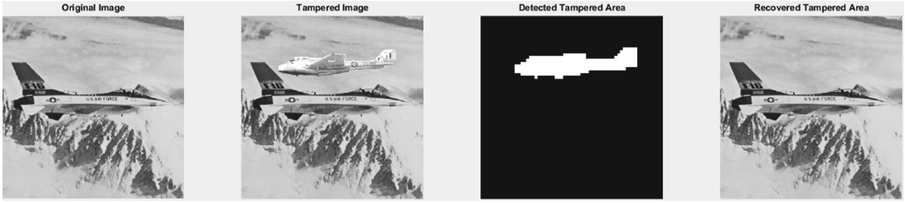

Fig. 5. (a) Original image (b) Tampered image (tampering rate = 19%), (c) Detected tampering (d) Recovered image

In the proposed hybrid method, two different copies of each block are available as the watermark data. While in 8 × 8 OIBTC method, according the amount of capacity of 2 LSBs and redundancy of data, at most one half of the blocks can have a second opportunity of another copy. In 4 × 4 OIBTC method, there is no second chance of having another copy. For this reason the method presented here could be more suitable for higher tampering rates since the probability of losing the first copy is higher. Furthermore, it can be more suitable for less textured images as presented in Table 1. Images which are more textured, e.g. Barbara and Mandril, the quality of recovered image is lower especially

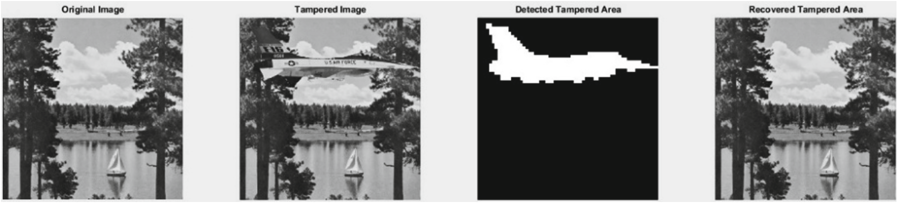

Fig. 6. (a) Original image (b) Tampered image (tampering rate = 25%), (c) Detected tampering (d) Recovered image

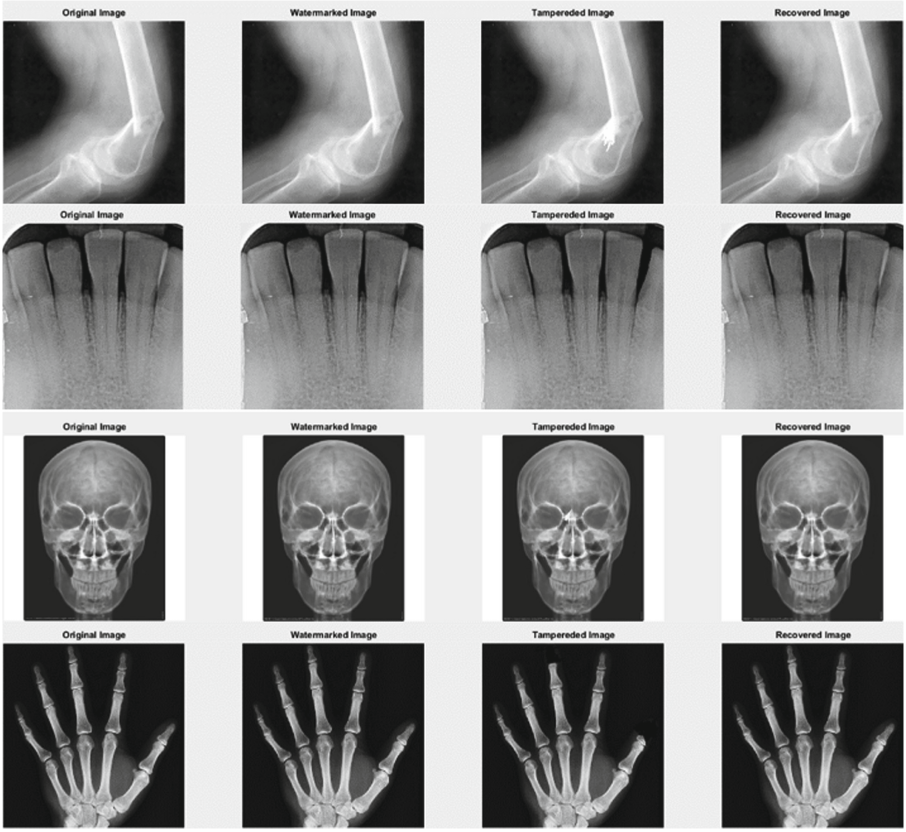

Fig. 7. (a) Original image (b) Watermarked image (c) Tampered image (d) Recovered image

when tampering rate is low compared with using just OIBTC. It is demonstrated that for most images with different tampering rates the proposed hybrid method has better performance.

Table 1. Comparison the results of Proposed Hybrid Scheme with 4 × 4 OIBTC [8] and 8 × 8 OIBTC [8] in terms of SSIM and PSNR for different standard images when tampering rates are different (the minimum values are shown for all SSIMs and PSNRs)

Standard images	8 × 8 OIBTC [8]		4 × 4 OIBTC [8]		Proposed Hybrid Scheme		(t) Tampering rate %
	SSIM	*PSNR (dB)*	*SSIM*	*PSNR (dB)*	*SSIM*	*PSNR (dB)*	
Lena	0.9036	30.16	–	–	**0.9162**	**31.84**	$45 < t < 50$
Lena	0.9534	33.92	0.9580	35.08	**0.9581**	**35.69**	$25 < t < 30$
Lena	0.9812	39.26	0.9855	41.66	**0.9839**	**42.02**	$10 < t < 12$
Barbara	0.8645	25.08	–	–	**0.8935**	**26.19**	$45 < t < 50$
Barbara	0.9384	28.98	**0.9425**	**29.14**	0.9422	29.02	$25 < t < 30$
Barbara	0.9721	32.99	**0.9766**	**33.43**	0.9701	32.24	$10 < t < 12$
Mandril	0.8474	26.66	–	–	**0.8550**	**27.01**	$45 < t < 50$
Mandril	0.9058	27.78	0.9232	28.68	**0.9288**	**28.92**	$25 < t < 30$
Mandril	0.9469	30.03	**0.9527**	**31.28**	0.9501	30.89	$10 < t < 12$
Woman-Darkhair	0.9383	35.29	–	–	**0.9521**	**38.13**	$45 < t < 50$
Woman-Darkhair	0.9673	38.29	**0.9766**	41.41	0.9759	**41.52**	$25 < t < 30$
Woman-Darkhair	0.9784	38.45	0.9816	38.14	**0.9842**	**40.15**	$10 < t < 12$
Woman-Blonde	0.8799	29.10	–	–	**0.8950**	**30.01**	$45 < t < 50$
Woman-Blonde	0.9405	33.73	0.9389	33.85	**0.9482**	**35.01**	$25 < t < 30$
Woman-Blonde	0.9651	35.09	0.9682	36.22	**0.9716**	**36.97**	$10 < t < 12$
Living room	0.8574	27.43	–	–	**0.8855**	**28.94**	$45 < t < 50$
Living room	0.9287	32.28	0.9296	32.36	**0.9416**	**33.54**	$25 < t < 30$
Living room	0.9687	37.22	0.9716	38.49	**0.9752**	**38.86**	$10 < t < 12$
Pepper	0.8883	28.53	–	–	**0.9098**	**30.43**	$45 < t < 50$
Pepper	0.9407	31.77	0.9539	33.04	**0.9543**	**33.94**	$25 < t < 30$
Pepper	0.9715	34.71	0.9789	36.21	**0.9800**	**37.65**	$10 < t < 12$
Lake	0.9475	30.80	–	–	**0.9622**	**32.65**	$45 < t < 50$
Lake	0.9737	33.98	0.9758	34.44	**0.9800**	**35.89**	$25 < t < 30$
Lake	0.9870	37.97	0.9895	38.99	**0.9912**	**40.42**	$10 < t < 12$
JetPlane	0.9582	31.45	–	–	**0.9658**	**32.62**	$45 < t < 50$
JetPlane	0.9878	42.47	0.9904	45.77	**0.9918**	**46.31**	$25 < t < 30$
JetPlane	0.9915	45.69	0.9938	47.27	**0.9940**	**48.27**	$10 < t < 12$
CameraMan	0.9610	30.43	–	–	**0.9691**	**32.15**	$45 < t < 50$
CameraMan	0.9760	32.06	**0.9816**	33.45	0.9812	**34.82**	$25 < t < 30$
CameraMan	0.9825	38.23	0.9930	43.81	**0.9932**	**43.89**	$10 < t < 12$
House	0.9507	31.87	–	–	**0.9785**	**36.84**	$45 < t < 50$
House	0.9769	34.64	0.9591	35.18	**0.9934**	**41.49**	$25 < t < 30$
House	0.9889	42.59	0.9901	45.61	**0.9972**	**47.76**	$10 < t < 12$

4 Conclusion

In this work, an image security scheme which can be applicable for sensitive medical images has been developed. This method not only provides excellent authentication detection, but also is able to recover the original image well, when it is necessary. To achieve this aim, an image is divided into set of pixel blocks, then watermarked data including authentication code and recovery codes is computed for each block. Authentication code for each block is 16 bits and is produced by a Hash function and should be hidden into the block itself. In order to authenticate an image, authentication code can be extracted and compared with the result of the hash function on the contents of the block. The OIBTC compression and the mean value are exploited for each block to generate recovery information. Another recovery code is available since there is a probability of losing one of the recovery codes as a result of tampering. Recovery codes are scrambled inside the image blocks to have better reconstruction of the image in case of tampering. The proposed method can embed two compressed copies of the image inside the image itself with high quality by applying two new ways; extracting different features depending on the types of blocks then reducing the number of needed bits for embedding as well. Experimental results demonstrate conclusively that this scheme can achieve superior performance for tampering detection, localization and recovery, especially when tampering rate is high. The proposed hybrid method uses block size of 8 × 8 for authentication code and block size of 4 × 4 or 8 × 8 for recovery code depending on the texture of the block. Although our proposed method showed good performance in recovery of image after high level of tampering, the accuracy of tamper localization could be improved further by considering adaptive block size for authentication code as well.

References

1. Singh, J., Patel, A.K.: An effective telemedicine security using wavelet based watermarking. In: IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1–6 (2016)
2. Rocek, A., Slavicek, K., Dostal, O., Javorník, M.: A new approach to fully-reversible watermarking in medical imaging with breakthrough visibility. Biomed. Sig. Process. Control **29**, 44–52 (2016)
3. Qin, C., Ji, P., Zhang, X., Dong, J., Wang, J.: Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy. Sig. Process. **138**, 280–293 (2017)
4. Dhole, V.S., Patil, N.N.: Self-embedding fragile watermarking for imagetampering detection and image recovery using self recovery blocks. In: IEEE International Conference on Computing Communication Control and Automation, (ICCUBEA) (2015)
5. Lee, T.Y., Lin, S.D.: Dual watermark for image tamper detection and recovery. Pattern Recognit. **41**(11), 3497–3506 (2008)
6. Shehab, A., et al.: Secure and robust fragile watermarking scheme for medical images. IEEE Access **6**, 10269–10278 (2018)
7. Thakkar, F.N., Kumar Srivastava, V.: A blind medical image watermarking: DWT-SVD based robust and secure approach for telemedicine applications. Multimedia Tools Appl. **76**(3), 3669–3697 (2017)

8. Qin, C., Ji, P., Chang, C.-C., Dong, J., Sun, X.: Non-uniform watermark sharing based on optimal iterative BTC for image tampering recovery. IEEE Multimedia **25**, 36–48 (2018)
9. Sing, D., Sing, S.K.: Effective self-embedding watermarking scheme for image tampered detection and localization with recovery capability. J. Vis. Commun. Image Represent. **38**, 775–789 (2016)
10. Joshi, A.M., Darji, A., Mishra, V.: Design and implementation of real-time image watermarking. In: IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), pp. 1–5 (2011)
11. Sun, W., Lu, Z.M., Wen, Y.C., Yu, F.X., Shen, R.J.: High performance reversible data hiding for block truncation coding compressed images. Sig. Image Video Process. **7**(2), 297–306 (2013)
12. Mohammad, N., Sun, X., Yang, H.: An adaptive visible watermarking algorithm for BTC compressed images. Inf. Technol. J. **13**(3), 536–541 (2014)
13. Mohammad, N., Sun, X., Yang, H.: An excellent image data hiding algorithm based on BTC. Inf. Technol. J. **10**(7), 1415–1420 (2011)
14. Tohidi, F., Abdul Manaf, A.B., Zamani, M., Jamshidi, H.: Improving the capacity of watermarking techniques by using block truncation coding. JDCTA **7**(14), 33 (2013)
15. Ji, P., Qin, C., Tang, Z.: Fragile watermarking with self-recovery capability via absolute moment block truncation coding. In: Sun, X., Liu, A., Chao, H.-C., Bertino, E. (eds.) ICCCS 2016. LNCS, vol. 10039, pp. 104–113. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48671-0_10
16. Chang, C.C., Chen, T.S., Wang, Y.K., Liu, Y.: A reversible data hiding scheme based on absolute moment block truncation coding compression using exclusive OR operator. Multimedia Tools Appl. **77**(7), 9039–9053 (2018)