Digital Signal Processing 73 (2018) 83–92

Contents lists available at ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Fast fragile watermark embedding and iterative mechanism with high self-restoration performance

Sergio Bravo-Solorio [a], Felix Calderon [b,*], Chang-Tsun Li [c], Asoke K. Nandi [d]

[a] Research & Innovation Department, DEIPI.COM S.A. de C.V., Mexico
[b] Univeridad Michoacana de San Nicolas de Hidalgo, División de Estudios de Posgrado, Facultad de Ingeniería Eléctrica, Santiago Tapia 403, Col. Centro, Morelia, Michoacan 58000, Mexico
[c] School of Computing and Mathematics, Charles Sturt University, Wagga Wagga, NSW 2650, Australia
[d] Department of Electronic and Computer Engineering, Brunel University London, London, UK

ARTICLE INFO

Article history:
Available online 20 November 2017

Keywords:
Fragile watermarking
Self-restoration
Image authentication
Erasure channel
Tornado codes

ABSTRACT

This paper presents a new algorithm to reduce significantly the computational cost of one of the best methods with self-recovery capabilities in the fragile watermarking literature. This is achieved by generating two sequences of reference bits associated to the 5 most significant bit-planes (MSBPs) of the image. The reference bits and some authentication bits are then allocated to the 3 least significant bit-planes (LSBPs) of the image. The receiver uses the authentication bits to localise altered pixel-blocks and then executes an iterative restoration mechanism to calculate the original value of the watermarked pixels. Experimental results demonstrate that the embedding method executes significantly faster compared to the state-of-the-art method while achieving a high restoration performance.

1. Introduction

The proliferation of powerful image editing software has raised serious concerns about the reliability of digital images, specially in application fields where altered content may lead to unacceptable consequences, e.g. law enforcement applications.

Fragile watermarking technology is aimed at exposing changes in the image content by identifying alterations on information embedded *a priori* (i.e., *watermark*). The fact that the embedded watermark undergoes the same distortions as the *host* image opens up the possibility of providing additional capabilities, such as *tampering localisation* and *self-recovery*.

Tampering localisation refers to the ability of identifying distorted regions, while verifying the integrity of the remainder of the image [1–4]. Self-recovery, on the other hand, refers to the ability of restoring the image content to its original state prior to the manipulation. The restoration can be either *approximate* or *exact*. Schemes with approximate restoration capabilities aim to recover a coarse version of the original content. For example, Qin et al. [5] embed reference bits generated from bits of vector quantisation (VQ) indices, along with some authentication bits, in the 3 LSBPs of the image. In the receiver side, manipulated pixels are located and the reference bits retrieved from unaltered pixels are used to recover a close approximation of the original content. This method is capable of restoring altered regions that extend up to 60% of the image. In [6], Qin et al. generate some reference bits from the mean value calculated from overlapping blocks of pixels, which are embedded into 1 or 2 LSBPs of the image. At the receiver end, manipulated regions are located and the mean values are reconstructed. A recovery operation is then conducted for every tampered pixel depending on its location in the overlapping blocks. The embedding strategy produce less embedding distortion and yet is capable of restoring images with manipulated regions of up to 45% of the image. Although these schemes can reconstruct considerably large tampered regions [7–18], the quality of the restored content may be insufficient for some applications.

Methods with exact restoration capabilities can recover the original content perfectly, provided that the altered area is not too extensive. In [19,20] Reed–Solomon error correction codes are used to calculate parity bits for every row and column of the cover image. The encrypted parity bits are embedded in the 2 LSBPs of the image. This scheme is capable of recovering up to 13 pixels in a single row or column and localising the distortions even if restoration is no possible. In Zhang and Wang's scheme [21], some reference bits are generated with the 5 MSBPs of the image. The receiver localises the altered regions by identifying changes in some authentication bits and then estimates the original pixels by means of exhaustive attempts. Nonetheless, the number of restored

* Corresponding author.
E-mail addresses: bravo.solorio@deipi.com (S. Bravo-Solorio), calderon@umich.mx (F. Calderon), chli@csu.edu.au (C.-T. Li), asoke.nandi@brunel.ac.uk (A.K. Nandi).

https://doi.org/10.1016/j.dsp.2017.11.005

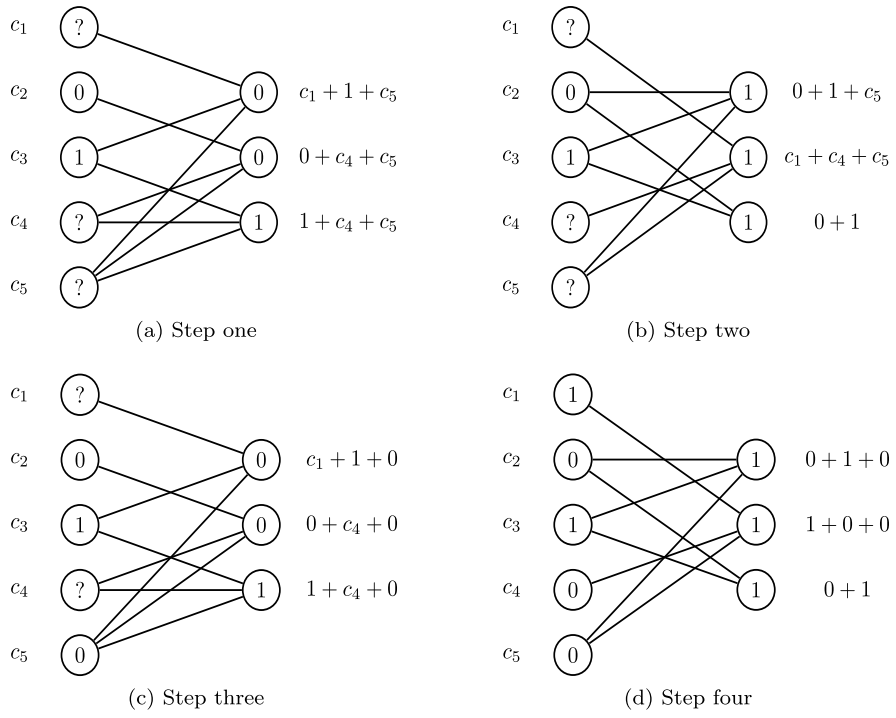


Fig. 1. Bipartite graphs used for decoding algorithm.

pixels drops sharply for tampered areas that cover above 6.6% of the image. To provide a more gradual decline of the restoration performance, an iterative approach was introduced in [22]. In [23], the reference bits and some authentication bits are reversibly embedded. At the receiver side, the surviving reference bits and the unaltered pixel bits are employed to estimate the original value of the altered pixels by means of solving some binary linear equation systems. With this method, up to 3.2% of the image can be restored. Furthermore, when the reference bits and the authentication bis are embedded with a non-reversible mechanism, up to 24%–28% of the image can be restored, depending on the initial settings and the image size [24]. However, the computational cost of the embedding algorithm may render the scheme unsuitable for real scenarios, wherein the watermark must be embedded at the time of capture [25,26].

In this paper, the problem of the embedding time consumed by the method proposed by Zhang et al. [24] is addressed. Inspired by the iterative approach of Tornado codes [27], two sequences of independent reference bits are generated as a result of associating every bit in the 5 MSBPs of the image to two different subsets. The restoration method executes an iterative mechanism to recover the altered pixels. The proposed method is detailed in Section 2. In Section 3, the restoration performance and the computational cost of the proposed approach are analysed, and some experimental and comparison results are reported in Section 4. Finally, some conclusions are given in Section 5.

2. Proposed watermark insertion method

The proposed iterative restoration approach is inspired by a class of erasure codes known as Tornado codes [27]. Typical Tornado codes are comprised of a series of random irregular bipartitie graphs. For a bipartitie graph, the left-most L nodes represent information bits which are to be transmitted reliably across the erasure channel. The nodes in all subsequent stages represent parity check bits, which form a sequence of graphs $(G_0, G_1, \cdots G_m, D)$. Assume that each stage G_i has $L\beta^i$ input bits and $L\beta^{i+1}$ output bits, for all $0 \leq i < m$ and $0 < \beta^i < 1$. Thus, the number of nodes shrinks by a factor β^i at the i-th stage except for the last one.

Unlike Tornado codes, which rely on a series of G_i bipartitie graphs to calculate the parity bits, in our approach we use a pair of graphs G_o and G_e. In the mathematical field of graph theory, a bipartite graph (or bigraph) is a graph whose vertices can be divided into two disjoint sets U and V, such that every edge connects a vertex in U to another in V. Both bigraphs will receive L information bits in parallel and calculate βL parity bits with a shrinking factor β. The L information bits and the $2\beta L$ parity bits are sent through the erasure channel.

To illustrate the rationale behind the proposed iterative restoration mechanism, consider the two example matrices $\mathbf{A}_o$ and $\mathbf{A}_e$ defined in (1).

$$\mathbf{A}_o = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{A}_e = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{1}$$

The bigraphs formed by the matrices above are shown in Fig. 1a and Fig. 1b, respectively. In the embedding process, two vectors, $\mathbf{r}_o = \mathbf{A}_o\mathbf{c}$ and $\mathbf{r}_e = \mathbf{A}_e\mathbf{c}$, are calculated using arithmetic modulo 2.

To retrieve the information, we use a process similar to the Tornado codes but alternating the solutions between the pair of graphs G_o and G_e. For example, Fig. 1 illustrates the steps followed to recover the missing information bits c_1, c_4 and c_5. Although, in the first step, it is impossible to calculate the missing values, the value of the bit c_5 can be calculated in the second step. In third step, it is possible to calculate the bits c_1 and c_4, given the bits c_2, c_3 and c_5. Finally, in the fourth step, it is verified that all the missing bits have been fully calculated and the iterative process ends.

2.1. The protected and discarded bit-planes

Given a 256 grey-scale image, sized $N_r \times N_c$ and let $p_n \in [0, 255]$ be a pixel, for $n = 1, \ldots, N$ $(N = N_r \times N_c)$. Every pixel p_n

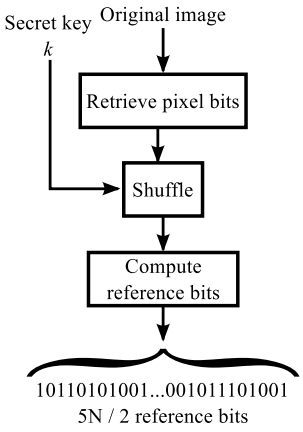


Fig. 2. A unique sequence of reference bits is computed in Zhang et al. [24].

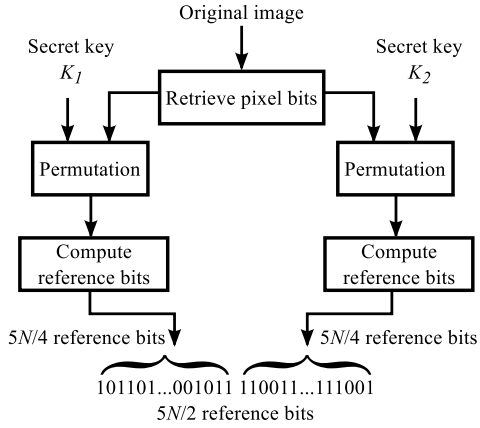


Fig. 3. In our proposed approach, the final sequence of reference bits is the concatenation of two sequences of reference bits independently generated.

can be decomposed in 8 bits, $b_{n,7}, \ldots, b_{n,0}$, where $b_{n,t} = (\lfloor p_n/2^t \rfloor \bmod 2)$, for $t = 0, \ldots, 7$. That is

$$p_n = \sum_{t=0}^{7} 2^t\, b_{n,t} \tag{2}$$

The bits $b_{n,7}, \ldots, b_{n,3}$, which make the greater contribution to the value of p_n in (2), will be referred to as the most significant bits (MSB). The bits $b_{n,2}$, $b_{n,1}$ and $b_{n,0}$ will be called the least significant bits (LSB). During the embedding process, the MSB ($5N$ in total) will remain intact, whereas the LSB ($3N$ in total) will be replaced with the watermark.

The embedding algorithm is comprised of two steps: 1) Reference bit generation and 2) Data embedding, which are detailed below.

2.2. Reference bits generation

Computing all the reference bits at once is computationally expensive in Zhang et al.'s method [24] as shown in Fig. 2. Therefore, we propose to generate two independent sequences of reference bits ($5N/4$ size), as shown in Fig. 3, and an iterative restoration mechanism at the receiver end. Thus, the redundant information about the 5 MSB is split into two strings of length $5N/4$, which are concatenated in a final reference vector with size $5N/2$.

In our approach, the $5N$ MSB are pseudo-randomly permuted in order to produce two permutation vectors, $\mathbf{c}^{(o)}$ and $\mathbf{c}^{(e)}$, with two different secret keys (K_o and K_e, respectively), as illustrated in Fig. 3. Then, each $\mathbf{c}^{(k)}$ vector, where $k \in \{o, e\}$, is divided into M subsets of L bits each; i.e., $M = (5N/L)$. Let $[\mathbf{c}^{(k)}_{m,1}, \ldots, \mathbf{c}^{(k)}_{m,L}]$ denote the bits in the m-th subset for the vector $\mathbf{c}^{(k)}$. For each subset, $(L/4)$ reference bits are generated by means of (3),

$$\begin{bmatrix} r^{(k)}_{m,1} \\ \vdots \\ r^{(k)}_{m,L/4} \end{bmatrix} = \mathbf{A}^{(k)}_m \times \begin{bmatrix} c^{(k)}_{m,1} \\ \vdots \\ c^{(k)}_{m,L} \end{bmatrix} \quad \text{for } m = 1, \ldots, M, \text{ and } k = 1, 2 \tag{3}$$

where $\mathbf{A}^{(k)}_m$ is a pseudo-random binary $(L/4) \times L$ matrix and the arithmetic in (3) is modulo-2. Note that this operation is similar to the generation of parity check bits in Hamming codes [28]. For security purposes, we assume that $\mathbf{A}^{(k)}_m$ is different for every m.

2.3. Data embedding

The two sequences of reference bits $\mathbf{r}^{(o)}$ and $\mathbf{r}^{(e)}$ are concatenated into $\mathbf{r}$, permuted upon a secret key and split into groups of 160-bits, each of which is embedded in pixel-blocks as follows.

A secure block-wise method resilient to cropping, which is a tailored version of the method in [1], was implemented to enable the detection and localisation of altered pixel-blocks. The image is divided into non-overlapping blocks of 8×8 pixels; let N_b $(= N/64)$ denote the total number of blocks. For each block, a 32-bit description code is encoded as $N_r||N_c||p$, where $0 \le p < N_b$ is the block index and $||$ denotes concatenation of bits. Observe that all the description bits share a *common prefix* (i.e., $N_r||N_c$); let Λ be the length of the common prefix. This information can be decoded by the receiver to localise manipulated blocks and, in case of cropping, restore the original dimensions of the image, while correcting possible displacements of the content. Additionally, for each block, a cryptographic hash function is fed with the 5 MSBPs of the block and the 160 reference bits to be embedded in the block. The exclusive-or operation between the description code and the resulting hash code is computed to generate a 32-bit authentication code:

$$a_{p,j} = w_{p,j} \oplus h_{p,j} \text{ for } p = 1, \ldots, N_b, \text{ and } j = 1, \ldots, 32 \ , \tag{4}$$

where $h_{p,j}$ is the j-th bit of the hash code, $a_{p,j}$ is the j-th bit of the authentication code, and $w_{p,j}$ is the j-th bit of the description code for the p-th block. The 3 LSBPs of the p-th block are replaced with the 192 bits obtained by concatenating the 160 reference bits and the encrypted version of the 32-bit hash code. To ensure robustness against the well known vector quantisation (VQ) attacks [29], a different secret key must be used for every cover image. A different block-size can be used to fulfil the requirements of different applications. However, the number of reference bits allocated in each block, and the length of the description code, should be adjusted accordingly.

2.4. Embedding distortion

To estimate the average distortion suffered by a host image, let us assume that the embedded watermark is driven by a uniform distribution. This is a reasonable assumption because of the properties of cryptographic hash codes. Because the watermark is embedded in the 3 LSBPs of the whole image, the average energy of the distortion on each pixel is given by

$$E = \frac{1}{64} \sum_{i=0}^{7} \sum_{j=0}^{7} (i-j)^2 = \frac{21}{2} \ , \tag{5}$$

so, the average Peak Signal-to-Noise Ratio (PSNR) is

$$\text{PSNR} \approx 10 \log_{10} \left(\frac{2 \times 255^2}{21} \right) = 37.9 \text{ dB} \ . \tag{6}$$

3. Proposed detection and image restoration methods

The detection process is comprised of two steps: 1) Cropping resistant tampering localisation, and 2) Iterative restoration, which are detailed below.

3.1. Cropping resistant tampering localisation

The received image, sized $N'_r \times N'_c$, is divided into non-overlapping blocks of 8×8 pixels. Let us denote the total number of blocks as N'_b. For each block, the 32-bit hash code and the 160 reference bits are extracted from the 192 bits in the 3 LSBP. The 32-bit hash code is decrypted to obtain the authentication code, meanwhile the 160 reference bits are fed to the same cryptographic hash function used at the embedding process to generate a 32-bit hash code. The description code for the p-th pixel-block is decoded by

$$w'_{p,j} = a'_{p,j} \oplus h'_{p,j} \text{ for } p = 1, \ldots, N'_b, \text{and } j = 1, \ldots, 32 \ , \tag{7}$$

where $a'_{p,j}$ is the j-th bit of the authentication code, $h'_{p,j}$ is the j-th bit of the hash code, and $w'_{p,j}$ is the j-th bit of the description code of the p-th block. Let $\mathcal{D}$ be a set of description codes, whose Λ MSBs are identical to each other. In a watermarked image, the cardinality $|\mathcal{D}|$, i.e., the number of elements in $\mathcal{D}$, is expected to be above a threshold τ_L.

If $|\mathcal{D}| \leq \tau_L$, it is possible that the left/upper-most edges of the input image had been removed by cropping. To address this problem, 64 shifted versions of the image are analysed as described above. Every shifted version is generated by displacing the image λ_i rows and λ_j columns, for $i = 0, -1, \ldots, -8$ and $j = 0, -1, \ldots, -8$. The detection process is terminated altogether if none of the shifted versions were regarded as watermarked. The probability that a non-watermarked image will be misjudged as watermarked (i.e., false positive) can be modelled by

$$P_{FP} = \left[1 - \sum_{q=0}^{\tau_L} \binom{N'_b}{q} 2^{-q\Lambda} (1 - 2^{-\Lambda})^{N'_b - q} \right] \times 64 \ , \tag{8}$$

where $\binom{N'_b}{q}$ is the binomial coefficient.

If the image is regarded as watermarked, the original dimensions of the image, N_r and N_c, are extracted from the common prefix of the description codes in $\mathcal{D}$ with higher occurrence to restore the original shape of the image in case of cropping. Finally, the block indexes retrieved from every description code are used to estimate possible *common displacements* to translate the authentic content to its original location, thereby resynchronising the watermark with the restoration mechanism detailed below. The reserved bits and pixel bits located within altered blocks will be regarded as "tampered", and as "reserved" otherwise.

3.2. Iterative restoration

The $5N/2$ reference bits are extracted, permuted back to their original location, and split into two sequences, $\mathbf{r}'^{(o)}$ and $\mathbf{r}'^{(e)}$, corresponding to the reference bits generated in the two stages at the embedding stage. The following procedure is repeated alternating the secret keys, K_o and K_e, in every successive iteration; that is K_o is used for odd iterations, while K_e is used for even iterations.

The $5N$ MSB of the image are shuffled, using the secret key as the seed of the pseudo-random permutation algorithm, to produce the vector $\mathbf{c}'^{(k)}$, for $k \in \{o, e\}$, which is divided into M subsets of L bits each. The same $(L/4) \times L$ pseudo-random matrix, $\mathbf{A}_m^{(k)}$, is generated as in the embedding stage. Let $\mathbf{r}_m'^{(k)} = [r'_1, \ldots, r'_{L/4}]$, and $\mathbf{c}_m'^{(k)} = [c'_1, \ldots, c'_L]$ be the vectors of reference bits and pixel bits of the m-th subset.

Let $N_E < (L/4)$ be the number of reference bits that were deemed "reserved" by the detection process. A system of equations similar to the one in (3) can be reordered as

$$\begin{bmatrix} \mathbf{r}'_R \\ \mathbf{r}'_T \end{bmatrix} = \begin{bmatrix} \mathbf{A}'_R \\ \mathbf{A}'_T \end{bmatrix} \mathbf{c}_m'^{(k)} \ , \tag{9}$$

where $\mathbf{r}'_R = [r'_{a_1}, \ldots, r'_{a_{N_E}}]^{\mathrm{T}}$ is the vector with the N_E "reserved" reference bits in $\mathbf{r}_m'^{(k)}$ and $\mathbf{A}'_R$ denotes their corresponding rows in $\mathbf{A}_m^{(k)}$, while $\mathbf{r}'_T = [r'_{a_{N_E+1}}, \ldots, r'_{L/4}]^{\mathrm{T}}$ is the vector with the "tampered" reference bits in $\mathbf{r}_m'^{(k)}$ and $\mathbf{A}'_T$ denotes their corresponding rows in $\mathbf{A}_m^{(k)}$.

Let $N_U < L$ be the number of "tampered" pixel bits in $\mathbf{c}_m'^{(k)}$, which can be rearranged, along with the columns in $\mathbf{A}'_R$ and $\mathbf{A}'_T$, to get them together. Thus, the system of equations can be rewritten as

$$\begin{bmatrix} \mathbf{r}'_R \\ \mathbf{r}'_T \end{bmatrix} = \begin{bmatrix} \mathbf{A}'_{R,T} & \mathbf{A}'_{R,R} \\ \mathbf{A}'_{T,T} & \mathbf{A}'_{T,R} \end{bmatrix} \begin{bmatrix} \mathbf{c}'_T \\ \mathbf{c}'_R \end{bmatrix} \ , \tag{10}$$

where $\mathbf{c}'_T = [c'_{b_1}, \ldots, c'_{b_{N_U}}]^{\mathrm{T}}$ is the vector with the N_U "tampered" pixel bits and $\mathbf{A}'_{R,T}$ and $\mathbf{A}'_{T,T}$ denote their corresponding columns in $\mathbf{A}_m^{(k)}$, while $\mathbf{c}'_R = [c'_{b_{N_U+1}}, \ldots, c'_{b_L}]^{\mathrm{T}}$ is the vector with the "reserved" pixel bits and $\mathbf{A}'_{R,R}$ and $\mathbf{A}'_{T,R}$ denote their corresponding columns in $\mathbf{A}_m^{(k)}$. From (10), it follows that,

$$\mathbf{r}'_R = \mathbf{A}'_{R,T} \mathbf{c}'_T + \mathbf{A}'_{R,R} \mathbf{c}'_R \ , \tag{11}$$

$$\mathbf{A}'_{R,T} \mathbf{c}'_T = \mathbf{r}'_R - \mathbf{A}'_{R,R} \mathbf{c}'_R \ , \tag{12}$$

where the values $\mathbf{r}'_R$, $\mathbf{A}'_{R,T}$, $\mathbf{A}'_{R,R}$ and $\mathbf{c}'_R$ are known.

The system of equations given by (12) has N_E equations and N_U unknowns, which can be solved using modulo-2 arithmetic if the system is linearly independent. In the following sub-section we calculate the probability that this system of equations will be linearly independent.

3.3. Probability of independence for the linear system

Let $0 \leq \alpha < 1$ be the ratio of bits regarded as "tampered". For a binary matrix $\mathbf{A}'_{R,T}$, of size $N_E \times N_U$, the probability that the Gaussian elimination will fail in the j-th column can be recursively determined as

$$P_{GE}(j) = P_{GE}(j-1) + [1 - P_{GE}(j-1)] 2^{-(N_E - j + 1)},$$
$$\text{for } j = 2, \cdots, N_E \ ,$$
$$P_{GE}(1) = 2^{-N_E} \ .$$

Provided that $N_E \geq N_U$, the probability that the N_E equations in $\mathbf{A}'_{R,T}$ will be linearly independent is

$$P_{LI}(\alpha, \alpha\gamma) = \sum_{i=1}^{L/4} \sum_{j=1}^{L} P_{N_E}(i, \alpha) P_{N_U}(j, \alpha\gamma) [1 - P_{GE}(i)] \ , \tag{13}$$

where P_{N_E} is the probability of the number of "reserved" reference bits in the subset (i.e., N_E), which obeys a binomial distribution given by

$$P_{N_E}(i, \alpha) = \binom{L/4}{i} (1 - \alpha)^i \alpha^{L/4 - i} \ , \tag{14}$$

and P_{N_U} is probability of the number of "tampered" pixel bits in the subset (i.e., N_U), which also obeys a binomial distribution described as,

$$P_{N_U}(j,\alpha\gamma) = \binom{L}{j}(\alpha\gamma)^j(1-\alpha\gamma)^{(L-j)} \ . \tag{15}$$

The parameter γ starts with value 1 at the first iteration. However, it is likely to decrease as long as the tampering rate is not too large; that is, the number of unknowns will decrease for every iteration. The convergence of the iterative restoration procedure is demonstrated by means of the total probability rule and the Bayes rule as follows.

Let $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_m$ be disjoint subsets of the set that contains all the "tampered" bits, $\mathcal{T}$, such that $\mathcal{T}_i \cap \mathcal{T}_j = \emptyset \ \forall i \neq j$, and, $\mathcal{T} = \bigcup_{i=1}^{m} \mathcal{T}_i$. Given $\mathcal{T} \cap \mathcal{T}_i = \mathcal{T}_i$, we can rewrite $\mathcal{T} = \bigcup_{i=1}^{m} \mathcal{T}_i$ equivalently by (16),

$$\mathcal{T} = (\mathcal{T} \cap \mathcal{T}_1) \cup (\mathcal{T} \cap \mathcal{T}_2) \cup \cdots \cup (\mathcal{T} \cap \mathcal{T}_m) \ . \tag{16}$$

Let $\widehat{\mathcal{T}}_q$ be the relative complement of the set $\mathcal{T}_q$ in $\widehat{\mathcal{T}}_{q-1}$. That is

$$\widehat{\mathcal{T}}_q = \widehat{\mathcal{T}}_{q-1} \setminus \mathcal{T}_q \ , \tag{17}$$

$$\widehat{\mathcal{T}}_0 = \mathcal{T} \ . \tag{18}$$

For every set in $\widehat{\mathcal{T}}_k$, we can easily show that $\widehat{\mathcal{T}}_{k-1} \cap \mathcal{T}_k = \mathcal{T}_k$ because $\mathcal{T}_k \subset \widehat{\mathcal{T}}_{k-1}$, and rewrite (16) as (19):

$$\begin{aligned} \mathcal{T} &= (\widehat{\mathcal{T}}_0 \cap \mathcal{T}_1) \cup (\widehat{\mathcal{T}}_1 \cap \mathcal{T}_2) \cup (\widehat{\mathcal{T}}_2 \cap \mathcal{T}_3) \cup \cdots \\ &\quad \cup (\widehat{\mathcal{T}}_{m-1} \cap \mathcal{T}_m) \ , \\ \mathcal{T} &= (\mathcal{T} \cap \mathcal{T}_1) \cup (\widehat{\mathcal{T}}_1 \cap \mathcal{T}_2) \cup (\widehat{\mathcal{T}}_2 \cap \mathcal{T}_3) \cup \cdots \\ &\quad \cup (\widehat{\mathcal{T}}_{m-1} \cap \mathcal{T}_m) \ . \end{aligned} \tag{19}$$

Applying the law of total probability and Bayes' Rule in (19), we obtain

$$\begin{aligned} P(\mathcal{T}) &= P(\mathcal{T} \cap \mathcal{T}_1) + P(\widehat{\mathcal{T}}_1 \cap \mathcal{T}_2) + P(\widehat{\mathcal{T}}_2 \cap \mathcal{T}_3) + \cdots \\ &\quad + P(\widehat{\mathcal{T}}_{m-1} \cap \mathcal{T}_m) \ , \end{aligned} \tag{20}$$

$$\begin{aligned} P(\mathcal{T}) &= P(\mathcal{T}_1|\mathcal{T})P(\mathcal{T}) + P(\mathcal{T}_2|\widehat{\mathcal{T}}_1)P(\widehat{\mathcal{T}}_1) + \cdots \\ &\quad + P(\mathcal{T}_m|\widehat{\mathcal{T}}_{m-1})P(\widehat{\mathcal{T}}_{m-1}) \ . \end{aligned} \tag{21}$$

Let $\beta_i = P(\mathcal{T}_i|\widehat{\mathcal{T}}_{i-1})$ be the likelihood of recovering the partition $\mathcal{T}_i$ given a subset of tampered bits $\widehat{\mathcal{T}}_{i-1}$. The probability of recovering the i-th subset is $P(\widehat{\mathcal{T}}_i) = \alpha\gamma_i = \alpha(1-\beta_1)(1-\beta_2)\cdots(1-\beta_i)$. Substituting these definitions in (21), it follows that

$$\begin{aligned} \alpha &= \beta_1\alpha + \beta_2\alpha(1-\beta_1) + \beta_3\alpha(1-\beta_1)(1-\beta_2) + \cdots \\ &\quad + \beta_m\alpha(1-\beta_1)(1-\beta_2)\cdots(1-\beta_{m-1}) \end{aligned} \tag{22}$$

and dividing (22) by α, we get,

$$\sum_{i=1}^{N_{iter}} \beta_i \prod_{j=1}^{i}(1-\beta_j) = 1 \ , \tag{23}$$

where $\beta_0 = 0$ and N_{iter} is the number of iterations necessary to reach the total restoration. We use (13) to estimate the probability of recovering β_i bits, given a ratio α of tampered bits:

$$\beta_i = P_{LI}(\alpha, \alpha\gamma_i) \ , \tag{24}$$

$$\gamma_i = \prod_{j=0}^{i-1}(1-\beta_j), \quad \forall i = 1, 2, \cdots, N_{iter} \ . \tag{25}$$

Algorithm 1, named Probability of Recovering Bits PRB(α), estimates the number of iterations and the percentage of recovered bits at each iteration. Figs. 4a, 4b, 4c, 4d, 4e and 4f were generated using Algorithm 1 for different tampering rates, α, namely, 0.05, 0.10, 0.15, 0.20, 0.25 and 0.27, respectively. The values of L tested in every experiment were 8, 16, 32, 64, 128 and 256. Observe that

Algorithm 1 PRB(α).

```
β_0 = 0, s_0 = 0 and i ← 0
while s_i ≤ 1 do
    i ← i + 1
    γ_i = ∏_{j=0}^{i-1} (1 − β_j)
    β_i = P_LI(α, αγ_i)
    s_i = s_{i−1} + γ_i × β_i
end while
N_iter = i
return N_iter, s
```

the number of iterations consumed by the restoration process significantly increases from $\alpha = 0.25$ to $\alpha = 0.27$ for greater values of L, e.g., $L = 256$ in Fig. 4f. On the other hand, with small values of L, the restoration process is incapable of restoring 100% of the tampered regions when the distortion is small, e.g., $\alpha = 0.05$ in Fig. 4a. This behaviour is caused by the effect of L in the probability of linear independence of the system of equations solved by the restoration process for small and large values of α. This is a key difference with Zhang et al.'s method [24], whose restoration performance improves for larger values of L, but sharply decreases for smaller values of L. For example, using a value of $L = 64$ in Zhang et al.'s method, the theoretical probability of restoring a tampered image of size 512×512 is zero, even for small tampering rates, e.g. $\alpha = 10$.

3.4. Complexity of the embedding process

The following is the analysis of the complexity of the proposed embedding approach. A matrix multiplication is computed for every one of the M subsets, which takes $\mathcal{O}(M \times L \times L/4)$. Since two sequences of reference bits are generated, the overall complexity of the algorithm is $\mathcal{O}(M \times L \times L/2)$, which is the same as in Zhang et al.'s scheme [24]. However, in Section 4, results show that the restoration performance of the proposed iterative restoration approach using $L = 52$ compares favourably to the performance of Zhang et al.'s scheme using $L = 256$. That represents a factor of 4 less computations for an equivalent restoration performance. This is particularly relevant in real applications, where watermarks must be embedded at the time of capture [25,26].

4. Experimental results

In this section, the performance of the proposed approach is tested and compared with Zhang et al.'s scheme [24]. Both schemes were implemented in C++ without any particular optimisation and the experiments were executed on an Intel Core i5 2.67 GHz processor with 4 GB RAM.

4.1. Iterative restoration

Fig. 5 illustrates the performance of the proposed cropping-resistant tampering location and the restoration methods. The image of Lena, sized 256×256, was watermarked using $L = 64$. The PSNR between the original and the watermarked images was assessed to be 34.60 dB. A tampered image, sized 231×231, was produced by removing the 25 left-most columns and the 25 bottom-most rows ($\alpha = 0.1858$). The original dimensions of the image, and the common displacement of the content, are retrieved from the description code of every unaltered pixel-block. The bottom-left image shows the watermarked image with the altered regions marked in black. The sequence of images at the bottom correspond to the images reconstructed in every subsequent iteration of the restoration algorithm. Note that all the altered pixels were successfully restored.

Fig. 6 shows two examples solved by the presented tampering localisation and restoration algorithms using $L = 64$. Fig. 6a and

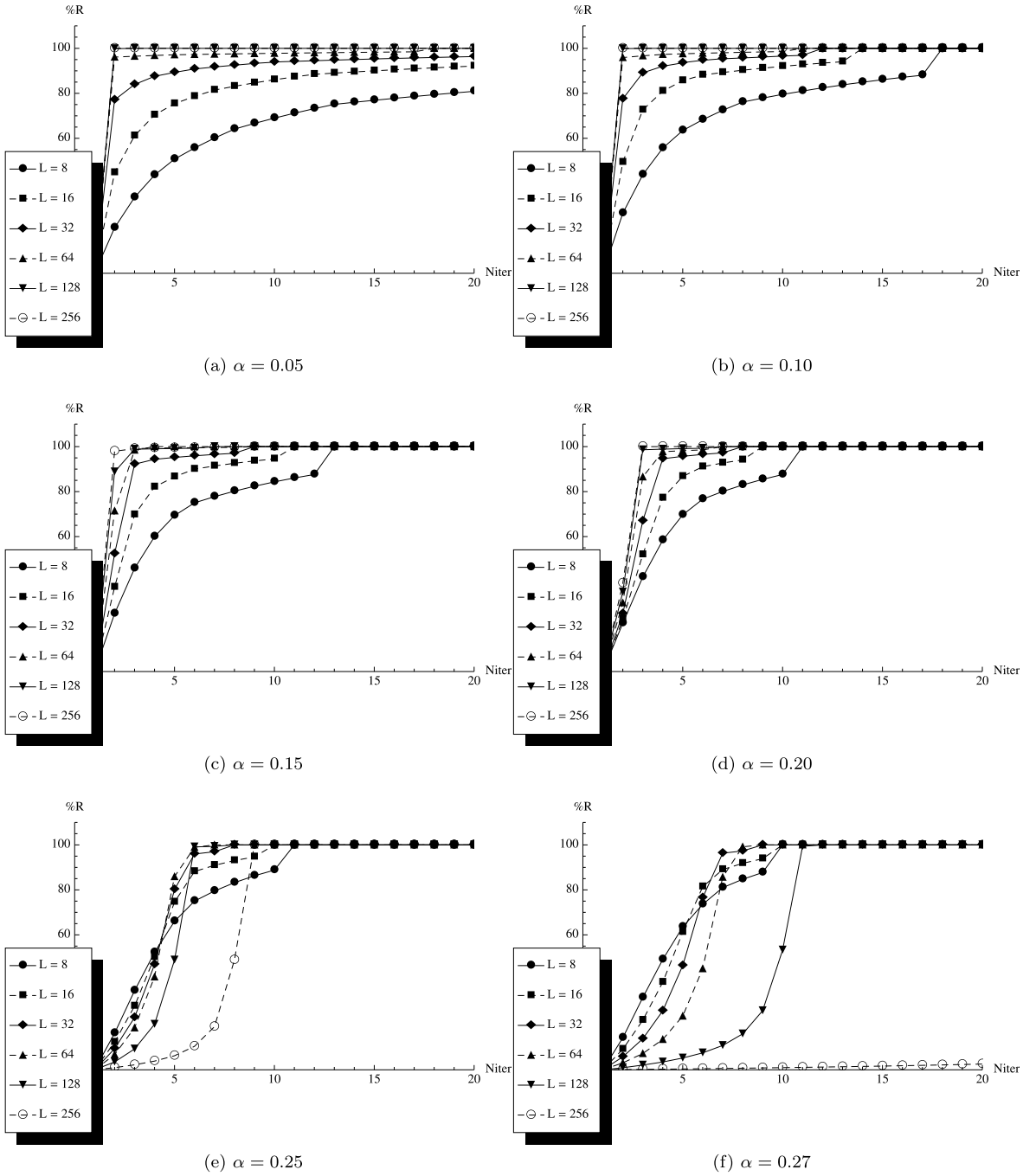


Fig. 4. Performance of the restoration process for $\alpha = 0.05$, 0.1, 0.15, 0.20, 0.25 and 0.27. For every experiment, the values of L tested were 8, 16, 32, 64, 128 and 256.

Fig. 6b show the original image of Lena, sized 256×256, and its watermarked version, respectively. Note that there is no significant difference to the naked eye. Fig. 6c shows the tampered image, which was created by copying one region of some image and pasting it on the watermarked image (Fig. 6b) using the image editor Gimp. The resulting tampering rate was assessed to be $\alpha = 0.1765$. Fig. 6f, Fig. 6g and Fig. 6h show the images reconstructed in subsequent iterations, starting with the image with all the detected tampered regions marked in black in Fig. 6e. Figs. 6i–6p show an image doctored to conceal the person that appears in the watermarked image, resulting in a tampering ratio $\alpha = 0.2263$. The image reconstructed in each subsequent iteration are shown in 6n, 6o and 6p.

The image in Fig. 7a was watermarked with the proposed method using $L = 64$, the original image was taken by a security camera in our department at the university. The embedding process took 2.9 seconds and the PSNR between the original and watermarked images was assessed to be 37.9 dB. Fig. 7b shows the manipulated version of watermarked images with a tampering ratio $\alpha = 0.2263$. The image in Fig. 7d shows the final image obtained by the iterative restoration algorithm and Fig. 7c shows the tampered regions detected by the algorithm. The cumulative percentage of restored MSB in every iteration is shown in Fig. 8 for tampered ratios $\alpha = 0.1538$ and $\alpha = 0.2263$. Observe that, in both cases, the actual restoration (AR) performance of the implemented method was very close to the theoretical estimation (TE) derived

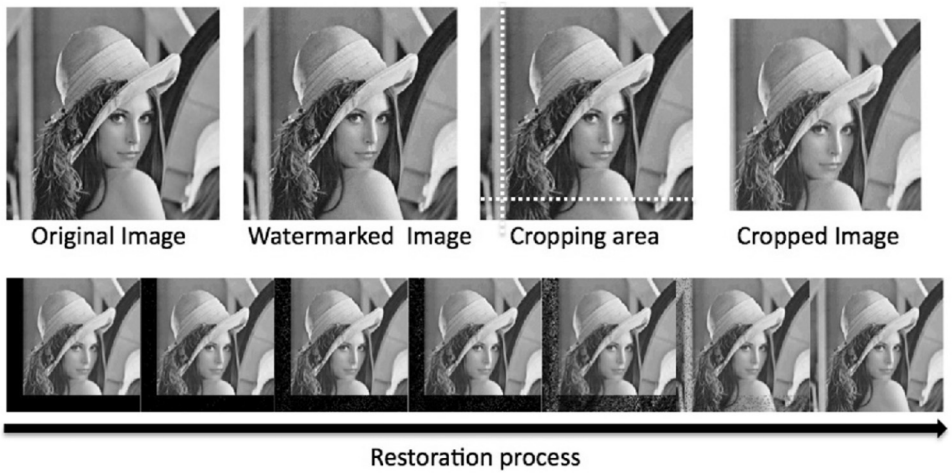


Fig. 5. Cropping-resistant tampering localisation and restoration performance of the proposed scheme. The original image, sized 256×256, was watermarked and then cropped to produce a 231×231 image. The image was completely reconstructed.

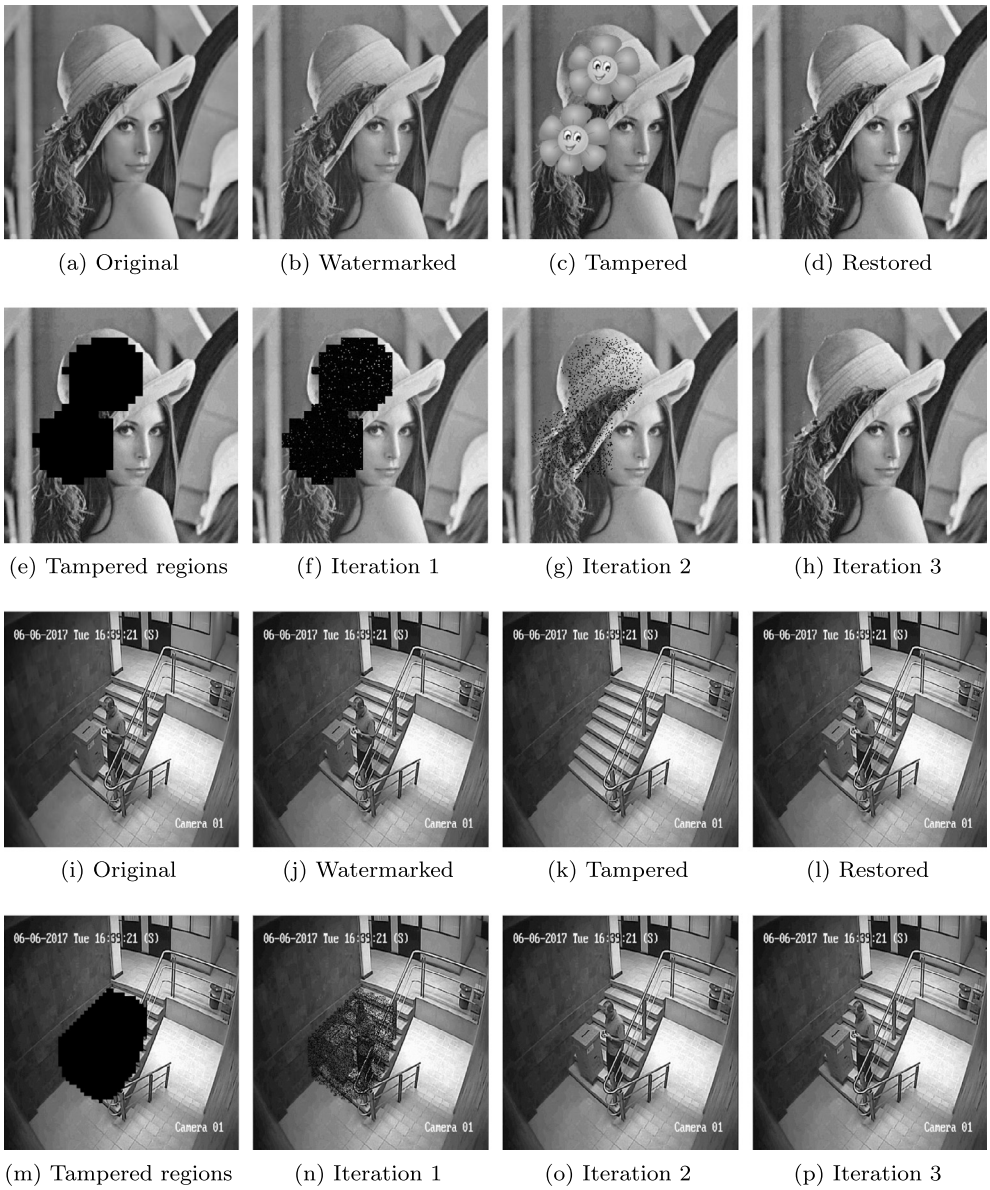
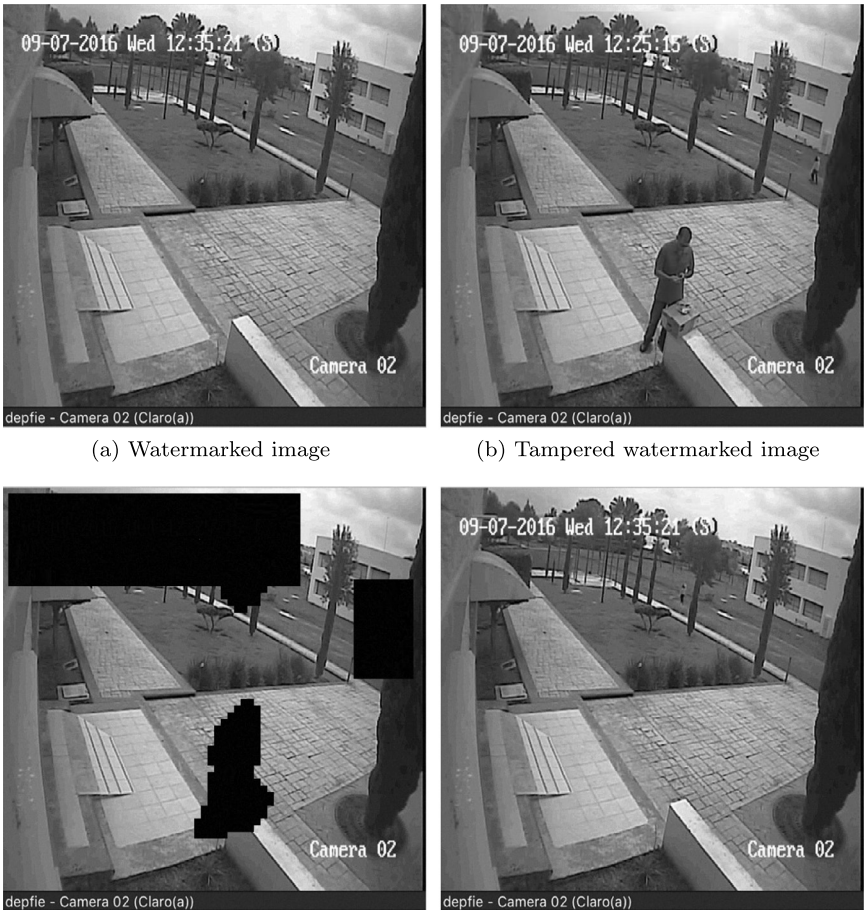


Fig. 6. Images obtained by the proposed embedding, tampering localisation and restoration methods. Upper image – size: 256×256, embedding distortion: PSNR = 34.60 dB, tampering: $\alpha = 0.1765$. Lower image – size: 512×512, embedding distortion: 37.84 dB, tampering: $\alpha = 0.2263$. In both cases $L = 64$.



(a) Watermarked image (b) Tampered watermarked image

(c) Region Tampered (d) Final image computed by Iterative Reconstruction

Fig. 7. Conventional tampering test for a security camera with image size 512×512. The tampering rate for this image was assessed to be $\alpha = 0.2263$.

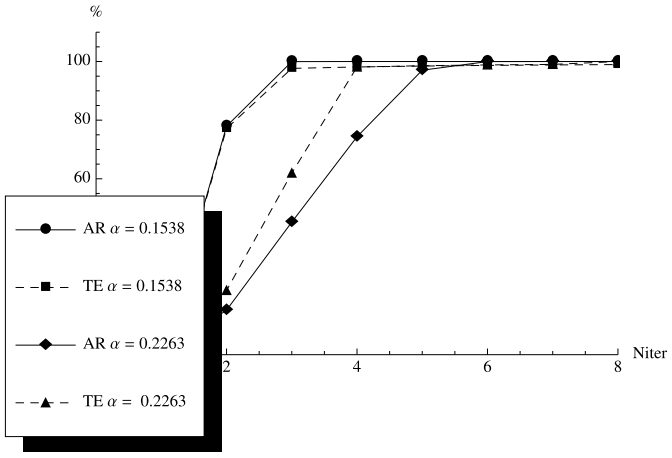


Fig. 8. Iteration for two tampered rate $\alpha = 0.1538$ and $\alpha = 0.2263$. AR (solid lines) represents the actual restoration, while TE (dashed lines) represents the theoretical estimation.

from the analysis in Section 3. Clearly, the restoration mechanism requires more iterations to recover greater portions of altered pixels.

4.2. Performance comparison

A total of 1000 images, sized 512×512, in the Caltech-256 data-set [30], were watermarked with the proposed scheme using five settings: $L = 32, 48, 52, 54$ and 64. For comparison purposes, the same experiment was conducted on images watermarked with Zhang et al.'s method [24] using $L = 64$ and 256. The following tampering ratios were tested for every image: 0.22, 0.23, 0.24, 0.25, 0.26 and 0.27. It is important to mention that Zhang et al.'s is the only watermarking scheme capable of exactly reconstruct the 5 MSBPs of tampered areas that extend beyond 28% of the image.

The results, summarised in Table 1, show that the iterative mechanism provides the best restoration performance using $L = 52$. In fact, it compares favourably to Zhang et al.'s scheme with $L = 256$ and yet the proposed embedding method executes almost 4 times faster. Observe that, in the proposed method, the execution time of the restoration process is worse for $L = 52$ than the rest of the values of L. This is because more iterations are required to restore a higher percentage of tampered pixels. An embedding time comparable to the one achieved with the proposed approach can be obtained using $L = 64$ in Zhang et al.'s method, but with a very poor restoration performance from their method. It is important to mention that Zhang et al. method is capable of restoring images with tampering rates up to $\alpha = 0.28$ using $L = 512$. However, with these settings, their embedding process takes about 25 seconds, which may be utterly impractical for real scenarios, where the watermark must be embedded at the time of capture [25,26].

The discrete nature of digital images causes the divergence between the actual and the theoretical restoration performance. For example, in our theoretical analysis, the best restoration performance was expected to be achieved with $L = 64$. However, in real images, the algorithm managed to restore a higher percentage of pixels with $L = 52$ for $\alpha = 0.26$. Moreover, our theoretical results

Table 1
Performance comparison between our proposed iterative approach and Zhang et al.'s method [24].

Method	L	Avg. time (s)		Restored pixels (%) at tampering ratio (α)					
		E	R	0.22	0.23	0.24	0.25	0.26	0.27
Iterative approach	32	2.7	26.9	100	100	93.4	17.1	0	0
	48	3.0	22.7	100	100	100	100	90.8	0
	52	**3.2**	**27.8**	**100**	**100**	**100**	**100**	**95.5**	**0**
	56	3.3	24.4	100	100	100	100	86.3	0
	64	4.4	23.1	100	100	100	100	19.7	0
Zhang et al. [24]	64	4.1	5.13	0	0	0	0	0	0
	256	15.8	25.0	100	100	100	100	0	0

Table 2
Performance comparison between our proposed iterative approach and the methods in [23,24,6,18].

Method	Average distortion (PSNR)		Max. tampering
	Embeddin*	Restoration**	
Qin et al. [6]	[38, 46] dB	[29, 41] dB	$\alpha < 0.45$
Qin et al. [18]	[37.91, 51.15] dB	[40.72, 51.14] dB	$\alpha < 0.20$
Zhang et al. [24] (2)	37.9 dB	[20, 40] dB	$\alpha < 0.66$
Zhang et al. [24] (1)	37.9 dB	$+\infty$	$\alpha < 0.28$
Zhang and Wang [23]	28.7 dB	$+\infty$	$\alpha < 3.2$
Proposed scheme	37.9 dB	$+\infty$	$\alpha < 0.26$

* Embedding distortion between the watermarked and the original images.
** Restoration distortion between the restored and the watermarked images.

showed that the proposed method should be capable of restoring images with tampering rates $\alpha \approx 0.28$. Nonetheless, in real images, the amount of pixels restored in the first iterations is insufficient to achieve the exponential growth shown in Fig. 4f. So, we empirically found that the proposed scheme is capable of restoring around 95% of altered pixels in images with tampering rates of up to 26%.

A performance comparison between the proposed scheme and five of the most relevant watermarking methods with self-restoration capabilities is presented in Table 2. Observe that the exact content of the watermarked image is successfully restored by the methods in [23], [24] (scheme 1) and the proposed scheme. The methods in [6] and [24] (scheme 2) manage to reconstruct images with higher tampering rates. However, the reconstruction is not exact, but an approximation of the original content. The restoration achieved by the method in [18] is approximate as well. However, the parameters of the algorithm can be adjusted to cause a significantly lesser embedding distortion at the expense of limiting the restoration capabilities to tampered areas that extend up to 20% of the image.

5. Conclusions

In this paper, a new algorithm has been proposed to reduce the computational cost of the method proposed by Zhang et al. [24]. Enlightened by Tornado codes, two sequences of pseudo-random reference bits are generated using different secret keys. The receiver end executes an iterative restoration mechanism, whereby the reference bits and MSB that belong to unaltered pixel-blocks are used to calculate the original value of the tampered MSB. We have presented a complete analysis of the iterative restoration mechanism. Furthermore, experimental results demonstrated that, using $L = 52$, the restoration performance of the proposed iterative mechanism compares favourably to the one achieved by Zhang et al.'s scheme using $L = 256$. The proposed embedding mechanism of the proposed scheme is faster by a factor of 4, yet it manages to restore up to 26% of the images. These results are particularly important for real scenarios, where the watermarks must be embedded at the time of capture. In fact, future work includes an optimised implementation of the proposed embedding scheme for video surveillance applications with low frame rates. Based on the presented results, we strongly believe that the computation time might be reduced by using more than two sequences of reference bits, at the expense of further decreasing the performance of the restoration process. Nonetheless, the theoretical and empirical boundaries for greater numbers of bit sequences is an open question that we are currently investigating.

References

[1] J. Fridrich, Security of fragile authentication watermarks with localization, in: Proc. of Security and Watermarking of Multimedia Contents, vol. 4675, SPIE, CA, USA, 2002, pp. 691–700.
[2] M.U. Celik, G. Sharma, E. Saber, A.M. Tekalp, Hierarchical watermarking for secure image authentication with localization, IEEE Trans. Image Process. 11 (6) (2002) 585–595.
[3] C.-T. Li, H. Si, Wavelet-based fragile watermarking scheme for image authentication, J. Electron. Imaging 16 (1) (2007) 1–9.
[4] S. Bravo-Solorio, A.K. Nandi, Two-layer fragilewatermarking method for enhanced tampering localisation, in: Proc. of ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing, 2012, pp. 2245–2248.
[5] C. Qin, P. Ji, J. Wang, C.-C. Chang, Fragile image watermarking scheme based on VQ index sharing and self-embedding, Multimed. Tools Appl. 76 (2) (2017) 2267–2287.
[6] C. Qin, P. Ji, X. Zhang, J. Dong, J. Wang, Fragile image watermarking with pixel-wise recovery based on overlapping embedding strategy, Signal Process. 138 (2017) 280–293.
[7] P.-L. Lin, C.-K. Hsie, P.-W. Huang, A hierarchical digital watermarking method for image tamper detection and recovery, Pattern Recognit. 38 (12) (2005) 2519–2529.
[8] X. Zhang, S. Wang, G. Feng, Fragile watermarking scheme with extensive content restoration capability, in: Proc. of IWDW – International Workshop on Digital Watermarking, 2009, pp. 268–278.
[9] H.J. He, J.S. Zhang, F. Chen, A self-recovery fragile watermarking scheme for image authentication with superior localization, Sci. China, Ser. F, Inf. Sci. 51 (10) (2008) 1487–1507.
[10] X. Zhang, S. Wang, Z. Qian, G. Feng, Self-embedding watermark with flexible restoration quality, Multimed. Tools Appl. 54 (2) (2010) 385–395.
[11] S. Bravo-Solorio, C.-T. Li, A.K. Nandi, Watermarking with low embedding distortion and self-propagating restoration capabilities, in: Proc. of ICIP – IEEE International Conference on Image Processing, 2012, pp. 2197–2200.
[12] H. He, F. Chen, H.-M. Tai, T. Kalker, J. Zhang, Performance analysis of a block-neighborhood-based self-recovery fragile watermarking scheme, IEEE Trans. Inf. Forensics Secur. 7 (1) (2012) 185–196.
[13] P. Korus, J. Bialas, A. Dziech, Iterative filtering for semi-fragile self-recovery, in: Proc. of WIFS – IEEE International Workshop on Information Forensics and Security, 2014, pp. 36–41.
[14] X. Li, X. Sun, Q. Liu, Image integrity authentication scheme based on fixed point theory, IEEE Trans. Image Process. 24 (2) (2015) 632–645.

[15] S. Sarreshtedari, M.A. Akhaee, A source-channel coding approach to digital image protection and self-recovery, IEEE Trans. Image Process. 24 (7) (2015) 2266–2277.
[16] H. Cai, H. Liu, M. Steinebach, X. Wang, A ROI-based self-embedding method with high recovery capability, in: Proc. of ICASSP – IEEE International Conference on Acoustics, Speech and Signal Processing, 2015, pp. 1722–1726.
[17] C. Qin, C.-C. Chang, P.-Y. Chen, Self-embedding fragile watermarking with restoration capability based on adaptive bit allocation mechanism, Signal Process. 92 (4) (2012) 1137–1150.
[18] C. Qin, H. Wang, X. Zhang, X. Sun, Self-embedding fragile watermarking based on reference-data interleaving and adaptive selection of embedding mode, Inf. Sci. 373 (2016) 233–250.
[19] J. Lee, C.S. Won, Authentication and correction of digital watermarking images, Electron. Lett. 35 (11) (1999) 886–887.
[20] J. Lee, C.S. Won, Image integrity and correction using parities of error control coding, in: Proc. of ICME – IEEE International Conference on Multimedia and Expo, vol. 3, 2000, pp. 1297–1300.
[21] X. Zhang, S. Wang, Fragile watermarking scheme using a hierarchical mechanism, Signal Process. 89 (4) (2009) 675–679.
[22] S. Bravo-Solorio, A.K. Nandi, Secure fragile watermarking method for image authentication with improved tampering localisation and self-recovery capabilities, Signal Process. 91 (4) (2011) 728–739.
[23] X. Zhang, S. Wang, Fragile watermarking with error-free restoration capability, IEEE Trans. Multimed. 10 (8) (2008) 1490–1499.
[24] X. Zhang, S. Wang, Z. Qian, G. Feng, Reference sharing mechanism for watermark self-embedding, IEEE Trans. Image Process. 20 (2) (2011) 485–495.
[25] H. Farid, A survey of image forgery detection, IEEE Signal Process. Mag. 2 (26) (2009) 16–25.
[26] J. Fridrich, D. Soukal, J. Lukáš, Detection of copy-move forgery in digital images, in: Proc. of Digital Forensic Research Workshop, 2003, pp. 55–61.
[27] M. Luby, Tornado codes: practical erasure codes based on random irregular graphs, in: Randomization and Approximation Techniques in Computer Science, Second International Workshop, Proceedings, RANDOM'98, Barcelona, Spain, October 8–10, 1998, 1998, p. 171.
[28] R. Hamming, Error detecting and error correcting codes, Bell Syst. Tech. J. 26 (2) (1950) 147–160.
[29] M. Holliman, N. Memon, Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes, IEEE Trans. Image Process. 9 (3) (2000) 432–441.
[30] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, Tech. Rep. 7694, California Institute of Technology, 2007, http://authors.library.caltech.edu/7694.

Sergio Bravo Solorio received a BEng degree in computer systems from the Morelia Institute of Technology, Mexico, in 2003, and an MSc degree in electrical and computer engineering from the University of Michoacan, Mexico, in 2005. He holds a PhD degree from the Faculty of Science and Engineering of the University of Liverpool, UK, awarded in 2011. The same year, he joined the Digital Forensics Research Group at the University of Warwick, UK, as an academic visitor, and, in 2012, he was granted a Research Fellow position to collaborate in the Milamber Digital project, supported by the Technology Strategy Board. In 2013, he was a member of the Editorial Board of the International Journal of Digital Crime and Forensics. He joined DEIPI.com, in 2013, as head of the Department of Research and Innovation where he has successfully steered numerous R&D projects from their conception to their implementation. His research interests include multimedia forensics and security, pattern recognition and image processing.

Felix Calderon is an Electrical Engineer graduated from the School of Electrical Engineering of the Michocana University of San Nicolas de Hidalgo (1987), Master in Computer Science graduated in the Technological of Toluca Mexico (1993) and Doctor in Computer Science graduated from the Research Center in Mathematics (CIMAT) in Guanajuato, Guanajuato Mexico (2001). His areas of interest are digital image processing and optimization. He is currently a member of the National System of Researchers by the National Council of Science and Technology CONACyT and professor and researcher in the Postgraduate Studies Division of the Faculty of Electrical Engineering of the Michoacana University of San Nicolas de Hidalgo, in Morelia Michoacan Mexico.

Chang-Tsun Li received the BEng degree in electrical engineering from National Defence University (NDU), Taiwan, in 1987, the MSc degree in computer science from U.S. Naval Postgraduate School, USA, in 1992, and the PhD degree in computer science from the University of Warwick, UK, in 1998. He was an associate professor of the Department of Electrical Engineering at NDU during 1998-2002 and a visiting professor of the Department of Computer Science at U.S. Naval Postgraduate School in the second half of 2001. He was a professor of the Department of Computer Science at the University of Warwick, UK, until Dec 2016. He is currently a professor of the School of Computing and Mathematics, Charles Sturt University, Australia, leading the Data Science Research Unit. His research interests include multimedia forensics and security, biometrics, data mining, machine learning, data analytics, computer vision, image processing, pattern recognition, bioinformatics, and content-based image retrieval. The outcomes of his multimedia forensics and machine learning research have been translated into award-winning commercial products protected by a series of international patents and have been used by a number of police forces and courts of law around the world. He is currently Associate Editor of the EURASIP Journal of Image and Video Processing (JIVP) and Associate of Editor of IET Biometrics. He involved in the organisation of many international conferences and workshops and also served as member of the international program committees for several international conferences. He is also actively contributing keynote speeches and talks at various international events.

Professor Asoke K. Nandi Professor Asoke K. Nandi received the degree of Ph.D. from the University of Cambridge (Trinity College), Cambridge (UK). He held academic positions in several universities, including Oxford (UK), Imperial College London (UK), Strathclyde (UK), and Liverpool (UK). In 2013 he moved to Brunel University (UK), to become the Chair and Head of Electronic and Computer Engineering. Professor Nandi is a Distinguished Visiting Professor at Tongji University (China), and an Adjunct Professor at University of Calgary (Canada). His current research interests lie in the areas of signal processing and machine learning, with applications to communications, gene expression data, functional magnetic resonance data, and biomedical data. He has made many fundamental theoretical and algorithmic contributions to many aspects of signal processing and machine learning. He has much expertise in Big Data, dealing with heterogeneous data, and extracting information from multiple datasets obtained in different laboratories and different times. He has authored over 550 technical publications, including 220 journal papers as well as four books. His Google Scholar h-index is 67 and ERDOS number is 2.

Professor Nandi is a Fellow of the Royal Academy of Engineering and also a Fellow of seven other institutions including the IEEE and the IET. Among the many awards he received are the Institute of Electrical and Electronics Engineers (USA) Heinrich Hertz Award, in 2012, the Glory of Bengal Award for his outstanding achievements in scientific research in 2010, the Water Arbitration Prize of the Institution of Mechanical Engineers (UK), in 1999, and the Mountbatten Premium, Division Award of the Electronics and Communications Division, of the Institution of Electrical Engineers (UK), in 1998.