WATERMARKING WITH LOW EMBEDDING DISTORTION AND SELF-PROPAGATING RESTORATION CAPABILITIES

S. Bravo-Solorio, C-T Li[*]

Computer Science Department
University of Warwick
Coventry, CV4 7AL, UK

A. K. Nandi

Department of Electrical Engineering & Electronics
The University of Liverpool
Brownlow Hill, Liverpool, L69 3GJ, UK

ABSTRACT

This paper presents a new fragile watermarking method, whereby two mechanisms are hierarchically structured to provide self-recovery capabilities. The first one is a secure block-wise mechanism, resilient to cropping, aimed at localising altered pixel-blocks. The second one is an iterative mechanism capable of reconstructing the original contents, by means of exhaustive attempts. The key features of the proposed method, which compare favourably to those of existing schemes, are low embedding distortion and resilience to cropping. Results demonstrate that the proposed scheme is capable of restoring the altered contents, even when the tampered region covers up to 32% of the total pixels in the image.

Index Terms— Fragile watermarking, authentication, self-recovery.

1. INTRODUCTION

The conspicuous increase in the use of digital images in very diverse application fields has motivated the development of security measures capable of providing effective authentication and integrity verification of the content of images. Fragile watermarking describe techniques to embed information imperceptibly – i.e. a *watermark* – in digital images, so that manipulations in *host* image can be indirectly revealed by changes identified in their watermarks [1–3].

Over the recent years, some fragile watermarking schemes have been proposed not only to identify and expose tampered regions, but also to restore the altered contents. This functionality is typically referred to as *self-recovery*. In Lin *et al.*'s method [4], information derived from the six most significant bit-planes (MSBPs) of small pixel-blocks, and subsequently embedded in another pseudo-randomly selected block, is used to localise and restore altered blocks. Nonetheless, tampered blocks cannot be restored when the blocks that allocate their information have been changed. To avoid this situation, usually referred to as the *tampering coincidence problem*, Zhang and Wang [5] proposed an scheme, whereby some reference bits, derived from the five most significant bits (MSBs) of every pixel, are embedded in the image. After localising the altered pixels, an exhaustive search mechanism is implemented to retrieve the exact pixels. However, the image can be restored as long as the proportion of tampered pixels is less than 6.6%. In [6], the reference bits, which are reversibly embedded, are used to reconstruct the original non-watermarked pixels. Nevertheless, the restoration mechanism works only when the tampered region covers less than 3.2% of the image. A similar approach was adopted in [7], where the reference bits, derived from blocks of quantised DCT coefficients, are retrieved to restore the image, given that the proportion of tampered pixels represents less than 59%. An improved version of the method, which manages to restore images containing tampered regions that represent up to 66%, has been presented in [8].

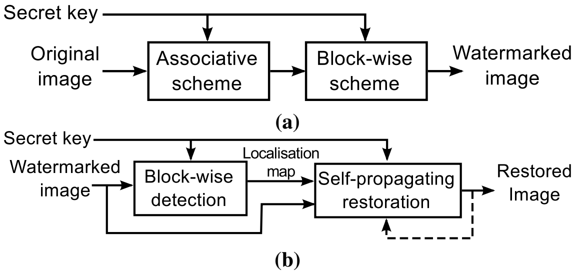


Fig. 1: General view of the method. (a) Embedding process. (b) Detection/Restoration process.

In this paper, a watermarking method that employs a novel iterative restoration mechanism is proposed. First, a block-wise mechanism is used to localise tampered pixel-blocks. Then, some reference bits allocated in unaltered blocks are used by the restoration mechanism, whereby the universe of possible combinations of values are exhaustively searched to identify potential candidates of the original blocks. The potential candidates are subsequently refined towards to a single representation of the original content of tampered blocks. The rest of the paper is structured as follows. Section 2 describes the proposed method and some results are reported in Section 3. Finally, some conclusions are presented in Section 4.

2. PROPOSED METHOD

2.1. Embedding Process

Consider a grey-scale image X, of size $n_1 \times n_2$, and denote its total number of pixels as $n = n_1 n_2$. We assume that both n_1 and n_2 are multiples of 8. The embedding process, illustrated in Fig. 1(a), is comprised of the two stages detailed below.

2.1.1. *Associative scheme*

Split the image into non-overlapping blocks of 2×2 pixels. The aim here is to associate every block to four different subsets of blocks. Some generated reference bits will serve to bind together the blocks

[*] This project has been jointly supported by the Milamber Project – funded by the Technology Strategy Board, October 2011 - September 2012, Project number: 4347-35177 –, and the National Council of Science and Technology (CONACyT) Mexico.

in every subset. This information is exploited by the self-propagating restoration mechanism described in Section 2.2.2.

The following steps are repeated four rounds; let r denote the round number, which is initialised to $r = 1$. In every round, some reference bits are allocated in the second LSB of a different pixel in every block.

1. Pseudo-randomly divide the blocks into disjoint subsets of m blocks each, using a key derived from a secret predefined function $k(r) \in \mathbb{N}$. Let X_i be the i-th subset, X_{ij} be the j-th block in X_i, and $X_{ij}(t)$ return the t-th pixel in X_{ij}. Let $\bar{X}_{ij}$ be an approximation of the block X_{ij}, defined as the mean of the 4 MSBPs of X_{ij} for $r = 1$, and as the mean of the 6 MSBPs of X_{ij} for $r \neq 1$. That is,
$$\bar{X}_{ij} = \begin{cases} (\sum_{t=1}^{4} \lfloor X_{ij}(t)/16 \rfloor)/4 & , \quad \text{if } r = 1 \\ (\sum_{t=1}^{4} \lfloor X_{ij}(t)/4 \rfloor)/4 & , \quad \text{otherwise} \end{cases} \quad . \tag{1}$$
where $\lfloor \cdot \rfloor$ is the floor function. The importance of this definition is explained in Section 2.2.2.
2. Compute m reference bits with the mean derived from every block in the subset, as $h = \mathcal{H}(\bar{X}_{i1}, \ldots, \bar{X}_{im})$, where $\mathcal{H}(\cdot)$ is a cryptographic hash function (e.g. SHA [9]).
3. Replace the second LSB of the r-th pixel in each block of the subset for one reference bit by,
$$\hat{X}_{ij}(r) = (2 \times \lfloor X_{ij}(r)/4 \rfloor) + (\lfloor h/2^{j-1} \rfloor \mod 2) \ , \tag{2}$$
4. Make $r = r + 1$ and repeat the steps while $r \leq 4$. Finally, gather together the watermarked blocks to form the watermarked image.

2.1.2. *Block-wise scheme*

Most of the existing watermarking schemes with self-recovery capabilities lose synchronisation when the dimensions of the cover images have changed as a result of cropping. To solve this problem, we adopted the block-wise method in [10].

Divide $\hat{X}$ into 8×8 non-overlapping blocks of pixels and denote the p-th pixel-block as $\hat{X}_p$. For each block $\hat{X}_p$, encode 64 authentication bits as, $w_p = \mathcal{I} \,||\, n_1 \,||\, n_2 \,||\, p$, where $\mathcal{I}$ is an image index exclusively associated to the image, and $||$ denotes concatenation of bits. Note that all the authentication bits share a *common prefix* (i.e. $\mathcal{I}_X \,||\, n_1 \,||\, n_2$). Let μ be the length of the common prefix.

To enable the identification of corrupted blocks, a hash is calculated as $g_p = \mathcal{H}(\hat{X}_p, k(1))$. Then, the watermarked block X_p^w is encoded as follows,

$$X_{p,q}^w = (\hat{X}_{p,q} \times 2) + (g_{p,q} \oplus w_{p,q}) \ , \tag{3}$$

where $\hat{X}_{p,q}$ and $X_{p,q}^w$ denote the q-th pixel in $\hat{X}_p$ and X_p^w, respectively, $g_{p,q} = [(g_p/2^{q-1}) \mod 2]$, and, $w_{p,q} = [(w_p/2^{q-1}) \mod 2]$. Finally, gather all the watermarked blocks to form the watermarked image X^w.

Assume that the distribution of the embedded watermarks is uniform. This is a reasonable assumption, because of the characteristics of cryptographic hashes. Since only the 2 LSBPs of the image are being replaced by the watermarks, the average energy of the distortion induced on each pixel is,

$$E_D = \frac{1}{16} \sum_{i=0}^{3} \sum_{j=0}^{3} (i-j)^2 = \frac{5}{2} \ , \tag{4}$$

so, the approximate average peak signal-to-noise ratio (PSNR) is,

$$\text{PSNR} \approx 10 \log_{10} \left(\frac{2 \times 255^2}{5} \right) = 44.2 \text{ dB} \tag{5}$$

2.2. Detection and Restoration Process

The detection and restoration process, illustrated in Fig. 1(b), is comprised of the following two stages.

2.2.1. *Block-wise detection*

Consider an $n_1' \times n_2'$ image, divided into non-overlapping blocks of 8×8 pixels. Encode a bit string h_p' with the LSBP of each block Y_p, and extract its authentication bits by, $w_p' = \mathcal{H}(\hat{Y}_p, k(1)) \oplus h_p'$, where $\hat{Y}_p = \lfloor Y_p/2 \rfloor$ and $k(1)$ is the key derived from the secret function. Let $\mathbf{A} = \{w_{a_1}', \ldots, w_{a_u}'\}$ be the set of authentication bits, whose μ MSBs are identical to each other. If the cardinality of $\mathbf{A}$ is greater than a predefined threshold τ_1, the image Y is deemed watermarked.

If no watermark could be identified, 64 shifted versions of Y are generated and analysed as described above. In every shifted version, all the pixels in Y are displaced λ_1 rows and λ_2 columns, where $-8 < \lambda_1, \lambda_1 \leq 0$. If none of the shifted versions were regarded as watermarked, the detection algorithm is terminated altogether. The probability that a non-watermarked image will be misjudged as watermarked can be modelled by,

$$\mathcal{P}_{D1} = \left[1 - \sum_{i=0}^{\tau_1} \binom{n_{\text{bw}}}{i} 2^{-i\mu} (1 - 2^{-\mu})^{n_{\text{bw}} - i} \right] \times 64 \ , \tag{6}$$

where n_{bw} is the total number of blocks.

If Y was deemed watermarked, retrieve its original dimensions, n_1 and n_2, from the common prefix. In case of cropping ($n_1' \neq n_1$ or $n_2' \neq n_2$), the dimensions of Y are restored by adding rows/columns of zeros to the right/bottom edges of the image. At this point, the shape of Y has been restored, but the content may have been displaced from its original location. Consider a function $\mathcal{L}(w_p')$ that returns the block index from the $(64 - \mu)$ LSBs in w_p'. Find a set of authentication bits $\mathbf{B} = \{w_{b_1}', \ldots, w_{b_v}'\}$, $\mathbf{B} \subseteq \mathbf{A}$, such that $\mathcal{L}(w_{b_1}') - b_1 = \ldots = \mathcal{L}(w_{b_v}') - b_v = \lambda$. The value λ is the *common displacement*; that is, the number of block slots the content has to be shifted to correct a possible displacement caused by cropping. Finally, an $n_1 \times n_2$ binary map M is encoded to localise altered pixel-blocks (filled with ones); note that only the pixel-blocks associated to the authentication bits in $\mathbf{B}$ are deemed genuine.

2.2.2. *Self-propagating restoration*

Divide Y into blocks of 2×2 pixels and identify all the unaltered blocks. Those blocks and the reference bits allocated in them will be called *reserved blocks* and *reserved reference bits*, respectively. The rest will be referred to as *tampered blocks* and *tampered reference bits*.

The following restoration mechanism uses the reserved blocks and the reserved reference bits to estimate the original mean of the 6 MSBPs of the tampered blocks, by means of exhaustive attempts. Follow the steps below, starting with $r = 1$.

1. Using the key $k(r)$, pseudo-randomly divide the blocks into disjoint subsets of m blocks each. Let Y_i be the i-th subset, Y_{ij} be the j-th block in Y_i, $Y_{ij}(t)$ returns the t-th pixel in Y_{ij},

and $\hat{h}_i^{(r)}$ be the m reference bits retrieved from the second LSB of $Y_{i1}(r), \ldots, Y_{im}(r)$.

2. Denote the number of tampered blocks as c in the subset Y_i. Skip the analysis of Y_i if the following condition is not met: $0 < c \leq \tau_2$, where τ_2 is a predefined threshold (usually set to 4). The importance of τ_2 is explained later on.

   Define $\bar{Y}_{ij} = (\sum_{t=1}^{4} \lfloor Y_{ij}(t)/16 \rfloor)/4$ for every reserved or restored Y_{ij}. For every tampered block, all the possible values that can be represented with 4 bits will be exhaustively tested, that is: $\bar{Y}_{ij} = 0, \ldots, 15$. Thus, for each combination (of a total of $n_c = 16^c$), a bit string $h' = \mathcal{H}(\bar{Y}_{i1}, \ldots, \bar{Y}_{im})$ is compared with the corresponding $(m - c)$ reserved reference bits in $\hat{h}_i^{(r)}$. Every match will serve to extend the set of potential 6-bit values associated to every tampered block. For example, consider that a match was obtained with a combination, say $\bar{Y}_{iu} = 5$ and $\bar{Y}_{iv} = 11$. So, their binary representation will become the 4 MSBs of the 6-bit numbers in the sets associated to the tampered blocks Y_{iu} and Y_{iv}, that is, $\mathbf{V_{iu}} = \{20, 21, 22, 23\}$ and $\mathbf{V_{iv}} = \{44, 45, 46, 47\}$.

   Once every subset has been analysed, make $r = 2$ and proceed with the next steps

3. Pseudo-randomly form the subsets using the key $k(r)$. This step is executed only for subsets that contain one or more tampered blocks (i.e. $c > 0$).

   The aim of this step is to refine the set of potential candidates associated to tampered blocks. Nonetheless, there may exist tampered blocks without associated sets (i.e. Step 2 was skipped). In these cases, a set with all the possible 6-bit values $0, \ldots, 63$ is associated to them.

   Define $\bar{Y}_{ij} = (\sum_{t=1}^{4} \lfloor Y_{ij}(t)/4 \rfloor)/4$ for every reserved or restored block Y_{ij}. The refinement of the potential values is possible only if the number of combinations to test is not too large. Hence, the analysis of the subset Y_i is skipped in this round if the number of combinations is greater than a predefined threshold τ_3. Then, exhaustively test all the combinations of potential candidates associated to every tampered block. For each combination, a bit string $h' = \mathcal{H}(\bar{Y}_{i1}, \ldots, \bar{Y}_{im})$ is compared with the corresponding $(m - c)$ reserved reference bits in $\hat{h}_i^{(r)}$. Then, discard, from the sets associated to tampered blocks, potential values that produced no matches.

   Once every subset has been analysed, make $r = r + 1$ and repeat this step while $r \leq 4$.

4. Restore the blocks Y_{ij} associated to a single value. For example, consider a tampered block, say Y_{iw}, associated to the set $\mathbf{V_{iw}} = \{52\}$. The 6 MSBs of the four pixels in Y_{iw} are set to $110100_2 (= 52_{10})$, and the block is flagged as *restored*.

5. Steps 1 to 4 are iteratively repeated, starting with $r = 1$, until no further blocks can be restored in Step 4.

A comprehensive probabilistic analysis of the proposed restoration mechanism is on their way. However, there are some interesting features that can provide valuable hints about the effectiveness of the scheme. Recall that the original mean of the 6 MSBPs of a tampered block must be in the range $[0, 63]$, and let β $(= 2^{c-m})$ denote the probability that a single trial will produce a match (recall Step 2). The probability that, in Step 2, a reduced number of potential candidates (< 64) will be found is given by,

$$\mathcal{P}_{R1} = \sum_{i=0}^{63} \binom{n_c}{i} \beta^i (1 - \beta)^{n_c - i} \quad , \tag{7}$$

From (7), it is evident that the exponential growth of n_c affects not only the computation effort, but also reduces the chances of filtering the number of potential candidates. The threshold τ_2, in Step 2, was introduced to avoid computationally expensive comparisons that are very likely to result in an unfruitful filtering – i.e. $\mathcal{P}_{R1} \approx 0$. The probability that the condition $0 < c \leq \tau_2$, in Step 2, will be met for a tampered subset can be modelled as,

$$\mathcal{P}_{R2} = \sum_{i=1}^{\tau_2} \binom{m}{i} \alpha^i (1 - \alpha)^{m-i} \quad . \tag{8}$$

where α denote the ratio between the number tampered blocks and the total number of blocks. We have empirically found that setting $\tau_2 = 3$ allows excessive computations without affecting the restoration performance.

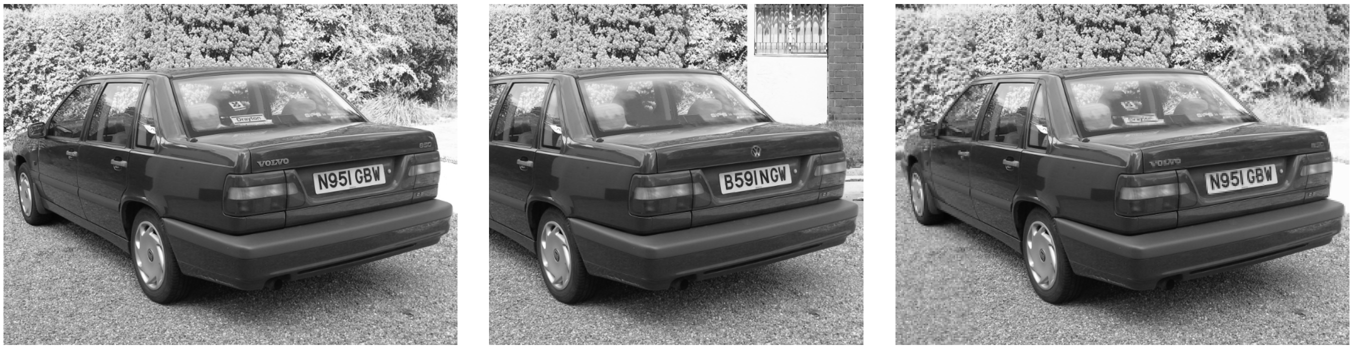
Finally, it is important to observe that α decreases as the number of restored blocks grows, thereby increasing the probability $\mathcal{P}_{R2}$, in (8), for the subsequent iteration. This produces an apparent *self-propagating effect* in the restoration mechanism.

3. RESULTS

The 480×640 image in Fig. 2(a) was watermarked with the proposed method, using the empirically defined/adjusted settings: $m = 12$, $\tau_1 = 0.01$, $\tau_2 = 3$, and $\tau_3 = 4096$. The PSNR between the original and the watermarked image was assessed to be 44.1 dB, which confirms the theoretically predicted distortion in (5). The watermarked version of the test image was tampered to generate the counterfeit in Fig. 2(b). The plate number was altered, the stickers on the rear window and the emblems on the boot were removed, and a new emblem and a portion of a house were superposed to create a more convincing counterfeit. Additionally, 18% of the left-most columns of the image were removed by cropping, thereby changing the shape of the image to 480×528. In total, an approximate of 31% of the pixels were either altered or removed. The block-wise method managed to retrieve the original shape of the image, and the restoration mechanism managed to reconstruct the image in Fig. 2(c). The PSNR between the restored region and the originally watermarked region was assessed to be 29.8 dB.

In an experiment conducted using 400 natural images, in the Caltech-256 data set [11], the proposed scheme managed to restore the content whenever the tampered area covered up to 32% of the image. The average embedding distortion was assessed to be 44.2 dB (PSNR), while the average PNSR between the restored area and the equivalent region in the watermarked image (restoration quality) was assessed to be 29.9 dB.

Table 1 compares the proposed method with 5 existing schemes. Note that, although cropping is commonly used to hide undesired information in images, it is not a manipulation supported by most of the existing methods. Both the embedding distortion and the restoration quality achieved with the proposed scheme is comparable to that in Lin *et al.*'s method. Nonetheless, due to the tampering coincidence problem, Lin *et al.*'s scheme is incapable of restoring a fraction of the tampered blocks, even when the altered region covers less than 20% of the image. Most of the remaining methods manage to restore images with a higher proportion of tampered pixels, compared to the proposed scheme, at the expense of inducing a significantly higher embedding distortion.



(a) (b) (c)

Fig. 2: Restoration test. (a) Original image. (b) Tampered image ($\approx 31\%$). (c) Restored image.

Table 1: Performance comparison.
* Measures empirically obtained from a data set of 400 natural images.

Method	**Average embedding distortion (PSNR)**	**Average restoration quality (PNSR)**	**Cropping**	**Condition for restoration**
Method in [4]	44.2 dB	29.9 dB	No	Limited by the tampering coincidence problem
Method in [7]	37.9 dB	[26,29] dB	No	Tampered area $< 59\%$
Method in [12]	37.9 dB	35.0 dB	No	Tampered area $< 35\%$
Method 1 in [8]	37.9 dB	$+\infty$	No	Tampered area $< 24\%$
Method 2 in [8]	37.9 dB	[22,40] dB	No	Tampered area $< 66\%$
Proposed method	44.2 dB *	29.9 dB *	Yes	Tampered area $\leq 32\%$ *

4. CONCLUSIONS

A new watermarking scheme with low embedding distortion and self-recovery capabilities. A block-wise mechanism is used to localise tampered blocks and correct possible displacements resulting from cropping. Subsequently, some reference bits are retrieved from the authentic blocks to reconstruct the altered content by means of exhaustive and iterative attempts. Results show that the proposed mechanism is capable of restoring images with a fairly good quality, even when the tampered region covers up to 32% of the image. Further work is planned to extend the proposed self-propagating restoration mechanism for larger portions of tampered pixels.

5. REFERENCES

[1] C. T Li, "Reversible watermarking scheme with image-independent embedding capacity," *IEE Proceedings Vision, Image and Signal Processing*, vol. 152, no. 6, pp. 779–786, 2005.

[2] C.-T. Li and Y. Yuan, "Digital watermarking scheme exploiting non-deterministic dependence for image authentication," *Optical Engineering*, vol. 45, no. 12, pp. 127001–1–6, 2006.

[3] Y. Yang, X. Sun, H. Yang, C.-T. Li, and R. Xiao, "A contrast-sensitive reversible visible image watermarking technique," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 5, pp. 656–667, 2009.

[4] P.-L. Lin, C.-K. Hsie, and P.-W. Huang, "A hierarchical digital watermarking method for image tamper detection and recovery," *Pattern Recognition*, vol. 38, no. 12, pp. 2519 – 2529, 2005.

[5] X. Zhang and S. Wang, "Fragile watermarking scheme using a hierarchical mechanism," *Signal Processing*, vol. 89, no. 4, pp. 675–679, 2009.

[6] X. Zhang and S. Wang, "Fragile watermarking with error-free restoration capability," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1490–1499, 2008.

[7] X. Zhang, S. Wang, and G. Feng, "Fragile watermarking scheme with extensive content restoration capability," in *Proc. of IWDW – International Workshop on Digital Watermarking*, 2009, pp. 268–278.

[8] X. Zhang, S. Wang, Z. Qian, and G. Feng, "Reference sharing mechanism for watermark self-embedding," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 485–495, 2011.

[9] *Secure Hash Standard*, National Institute of Standards and Technology, Washington, 2002, Federal Information Processing Standard 180-182.

[10] S. Bravo-Solorio and A. K. Nandi, "Secure fragile watermarking method for image authentication with improved tampering localisation and self-recovery capabilities," *Signal Processing*, vol. 91, no. 4, pp. 728–739, 2011.

[11] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," Tech. Rep. 7694, California Institute of Technology, 2007.

[12] Z. Qian, G. Feng, X. Zhang, and S. Wang, "Image self-embedding with high-quality restoration capability," *Digital Signal Processing*, vol. 21, no. 2, pp. 278–286, 2011.