Digital Signal Processing 22 (2012) 324–336

Contents lists available at SciVerse ScienceDirect

Digital Signal Processing

www.elsevier.com/locate/dsp

Watermarking using similarities based on fractal codification

Pedro Aaron Hernandez-Avalos, Claudia Feregrino-Uribe *, Rene Cumplido

Department of Computer Science, National Institute for Astrophysics, Optics and Electronics, Luis Enrique Erro No. 1, C.P. 72840, Sta. Maria Tonantzintla, Puebla, Mexico

ARTICLE INFO

Article history:
Available online 17 November 2011

Keywords:
Watermarking
Similarities
Fractal code

ABSTRACT

This paper describes a digital watermarking scheme based on fractal codification for 8-bit gray scale images; it replaces range blocks by modified blocks according to the watermark bit being embedded. The main contribution of this work is a decrease in the distortion generated by the watermark embedding in the carrier image compared to the reference scheme based on fractal codification; in addition, the scheme achieves a better robustness against JPEG attacks, a decrease at 13.2 dB in distortion and up to 50% improvement in Bit Correct Ratio (BCR). The scheme relies on the selection of interest points, local searching regions and embedding regions to be successful. Finally, this document presents a comparison of the results obtained with the proposed scheme and other schemes inspired by the fractal codification.

1. Introduction

The explosive growth of the Internet in the last years along with the exponential increase in computer performance, has facilitated exchange of multimedia information like images. As the multimedia content sharing through the Internet has increased dramatically, the necessity to protect digital content authoring has arisen with great strength. This has contributed to the creation of many schemes that hide information in media, called watermarking schemes. Both, the selection of a watermarking scheme and the type of information to be hidden or embedded (such as medical data, serial numbers, authoring and copy control information) depend on the application.

The former name of watermarking is steganography, which comes from the Greek words *stegano*, that means '*hide*', and *graphos*, that means '*writing*', and literally means *covered writing*. Today, the term steganography is not very common anymore; instead, most people use the terms (*digital*) *watermarking*, *data embedding* and *information hiding* indistinctly being watermarking the most recognized. However, there are people who treat steganography and watermarking as two different concepts [1,2]. Cox et al. [3] define watermarking as the practice of imperceptibly altering a medium to embed a message about that medium, whereas steganography is the practice of undetectably altering a medium to embed a secret message. Even though the objectives of watermarking and steganography are quite different, both applications basically alter a medium to embed a message.

In the watermarking jargon, the information to be hidden is called *watermark* or *message*, the medium where the watermark is hidden is called *original medium* or *carrier medium*, and the medium with the watermark inside is called *watermarked medium* or *stegomedium*. The process that hides a watermark in the carrier medium is called *embedding process*, and the process that tries to determine whether a watermark is present, and if so retrieves the encoded message, is the *decoding process*.

There is a varied gamut of watermarking techniques with different characteristics. For example, spread spectrum based schemes are known by its robustness, since it is difficult to intercept and to remove the watermark hidden in the image. The schemes, based on spread spectrum, embed information by linearly combining the carrier medium with a small pseudo-noise signal that is modulated by the message [4]. A different kind of watermarking technique is called *patchwork*, it is a statistical approach that pseudorandomly chooses two patches, A and B, adds a small constant value to the sample values of patch A, and subtracts the same value from the sample values of patch B; thus, the original sample values are slightly modified. The detection process starts with the subtraction of the sample values of the two patches; then it obtains the difference of the sample means, and the expected value helps to decide whether the samples contain a watermark or not. This scheme has two main drawbacks: the samples of each patch should be large enough (uniformly distributed samples), and the sample mean values must be equal [5]. Another kind of watermarking schemes are based on quantization index modulation (QIM). This kind of schemes embeds information by modulating an index, or sequence of indexes, with the message, and by quantizing the carrier medium with the associated quantizer or sequence of quantizers [6]. This kind of schemes exhibit a significant gain in terms of watermark capacity over known-host statistics schemes such as spread spectrum, they were shown in turn to be easily defeated by even the simplest attacks, as it is mentioned in [7] that

* Corresponding author.
E-mail addresses: phernandez@inaoep.mx (P.A. Hernandez-Avalos), cferegrino@inaoep.mx (C. Feregrino-Uribe), rcumplido@inaoep.mx (R. Cumplido).

1051-2004/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.dsp.2011.10.012

besides proposes an improved QIM technique, called Angle QIM, which is insensitive to amplitude scaling attacks.

A different kind of watermarking techniques hides information using the internal tasks of the compression process. For example, information can be hidden by utilizing the transform coefficients of JPEG and JPEG 2000 compression, by altering the color palette of GIF compression, or by handling the image blocks of the fractal compression.

Particularly, for digital images there can be many ways of hiding a message; for example, by hiding information directly on the pixels by varying its color intensity, or by modifying the magnitude of the transform coefficients (as used in internal procedures of image compression). Therefore, there are watermarking techniques working in the space domain and in the transform domain, respectively.

Fractal image compression has been the inspiration for several techniques whose main characteristics are the use of image blocks and their similarities to alter a medium to embed a message, this techniques are further discussed in Section 3.

Although fractal based watermarking techniques are not as popular these days as spread spectrum or QIM, recent works based on fractal codification for watermarking have achieved improvements in terms of either robustness, imperceptibility or capacity. Continuing with these efforts, this work describes a watermarking technique that results in lower distortion when compared with recent works based on fractal codification.

The organization of this paper is as follows: Section 2 presents a basic description of fractal codification, including terminology used throughout the paper. Section 3, exhibits a review of watermarking schemes that are based on, or inspired by, fractal coding. Section 4 describes the proposed watermarking scheme. Section 5 presents the experiments and results obtained and finally conclusions are drawn in Section 6.

2. Fractal codification

The term 'fractal' was created by Benoit Mandelbrot in 1983, it is derived from the Latin *fractus* that means '*interrupted*', '*irregular*' or '*fractional*'. This term is a good description of one of the geometric properties that fractals present: they have a fractional dimension [8].

Basically, fractals have two main properties: self-reference and self-similarity. Self-reference indicates that an object appears in its own definition, which implies the need for a recursive algorithm to generate a fractal. Self-similarity implies that the fractal object presents the same appearance regardless of the object's enlargement degree. In other words, no matter how infinitely the zone of a fractal object expands, it will always contain the initial fractal object.

Fractals have varied forms, including forms found in nature like clouds, mountains, trees, leaves, etc. Fig. 1 shows some fractals having form of plants, they have the two features described previously [9].

The iterated function systems or IFS are commonly used to generate fractals, their functionality is similar to that of a photocopying machine [10]. Suppose that a special kind of photocopying machine reduces the image to be copied by half and reproduces it three times on the copy, as seen in Fig. 2. If the new image is copied in the same way the resulting image will contain nine reduced versions of the original image. Fig. 3 illustrates this process iterated three times; notice that all of the images seem to converge to the same final image, regardless of the original image. Moreover, it can be observed that this resulting image is a copy of itself and is detailed in all scales. This image is a *fractal* and is known as *system attractor*. In this example, the photocopying machine represents a three function system and the feedback action represents the iterative part of the system. Each of these functions contracts and transforms the input image and the three functions together form an output image with three different figures, which represents an Iterated Function System. Each function of the IFS performs an affine transformation consisting of rotation, translation and escalation operations over each point of the figure or fractal curve into study. In other words, a point with coordinates (x, y) is translated to the coordinates (u, v). Eq. (1) controls this transformation.

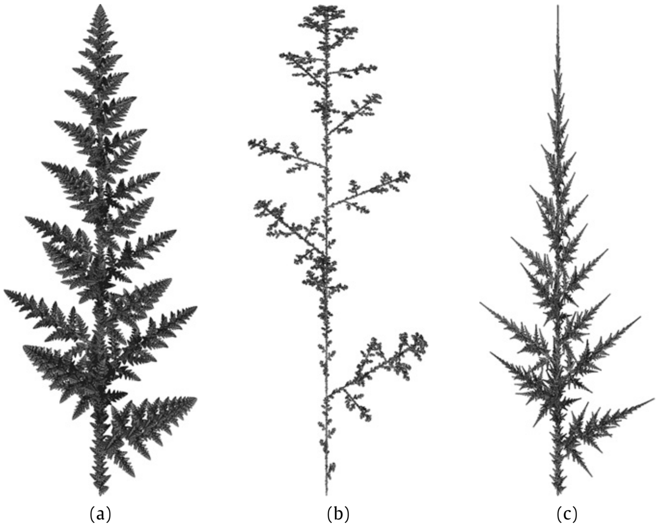


Fig. 1. Fractals with form of plants.

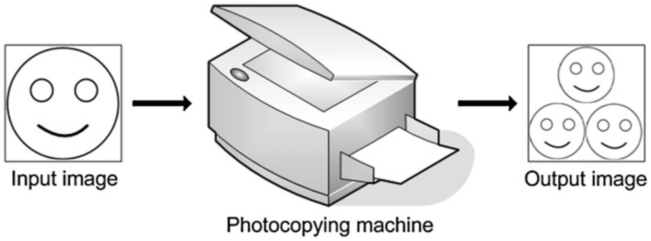


Fig. 2. Photocopying machine that reproduces three times the input image.

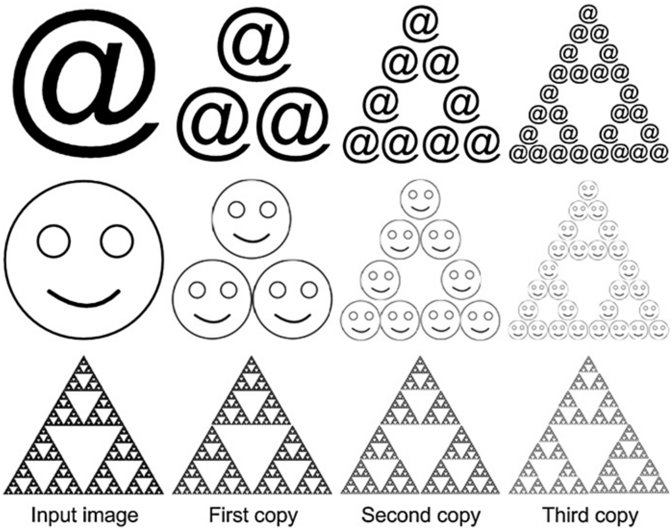


Fig. 3. First three copies generated by the special machine of Fig. 2.

$$\begin{bmatrix} u \\ v \end{bmatrix} = w \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{1}$$

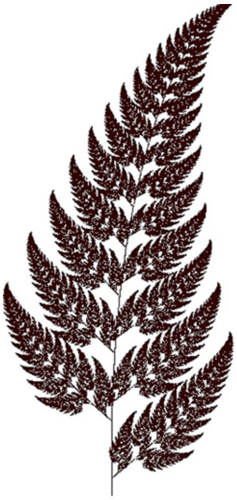


Fig. 4. IFS and its fractal image generated.

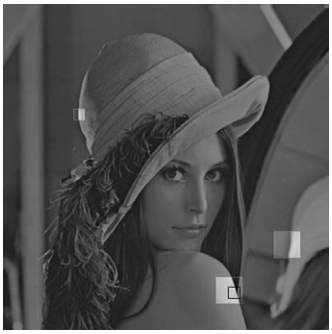


Fig. 5. Self-similar parts in Lenna.

The parameters a, b, c and d perform the rotation of each point, with their magnitudes corresponding to the scaling factor. Parameters e and f perform the linear translation in x and y of the same point. Therefore, the general form of an Iterated Function System is:

$$T_k(x) = \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \cdot x + \begin{bmatrix} e_k \\ f_k \end{bmatrix} \tag{2}$$

for $1 < k < n$, where x is a point in the $\Re^2$ plane and n is the total number of affine transformations. If T_k is a contractive mapping, then the attractor can be obtained through a set of iterated functions. Fig. 4 shows a fractal image generated by a system of four iterated functions.

The fractal image compression comes from the idea of finding an IFS that represents and generates a given image. Since most images do not present the self-reference feature as the fractals in Fig. 1 do, the fractal compression schemes use regions of the image that can be similar at different scales, thus generating a Partitioned Iterated Function System of PIFS. For example, Fig. 5 shows how a small part of the shoulder of Lenna is put on a smaller part of the same region, which is almost identical. In the same way, a region of the mirror reflection is similar to a small region of her hat after a transformation.

The image compression using PIFS is considered fractal due to the compressed image is the system attractor (just like IFS, which generates a self-similar and self-referenced attractor). The main tasks performed by the fractal codification are: partition of the image, search of similarities, modification of luminance and substitution of blocks.

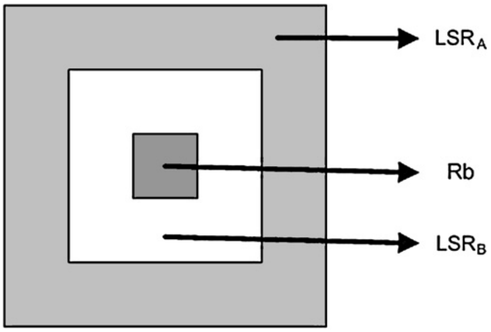


Fig. 6. Range block *Rb* and its local search region LSR_A, LSR_B. The LSR_C is the union of LSR_A and LSR_B.

3. Watermarking inspired by fractal codification

As mentioned earlier, fractal image codification was the inspiration for the creation of several steganographic techniques whose main characteristic is the use of image blocks and their similarities. The main steganographic techniques that use these features are described in this section.

Two kinds of image blocks used in schemes inspired by fractal codification are: *range blocks* and *domain blocks*. Range blocks define the blocks to be replaced by modified blocks, which take information from domain blocks. To select the domain blocks, which resemble more the range blocks, a similarities search is performed.

One of the first works inspired by fractal image codification was developed by Joan Puate and Fred Jordan [11]. In this work, the fractal image compression process is used to hide a watermark. This work is a modification of the fractal compression algorithm proposed by Jacquin [12], limiting the block domain search into a local search region (*LSR*), which led to Local Iterated Function Systems or LIFS. In this scheme, the number of range blocks determines the maximum embedding capacity. Domain block set D for a determined range block *Rb* is limited by subsets LSR_A, LSR_B and LSR_C (LSR_C is the union of LSR_A and LSR_B). These subsets represent regions where a block, with the minimum error with respect to *Rb*, is searched (Fig. 6). In general, in the fractal compression process the search of similarities between the *Rb* and these subsets determines the way the watermark is embedded. The embedding rule is simple: If a bit 1 is to be hidden, a domain block in LSR_A region is searched; if a bit 0 is to be hidden, a domain block in LSR_B region is searched. When the watermark has been embedded and there are range blocks left, the remaining range blocks are codified by searching in LSR_C.

Li Guanhua, Zhao Yao and Yuan Baozong proposed in [13] a similar scheme to the one by Puate. This scheme embeds the watermark using the fractal image compression process by manipulating the luminance shift O_u from each range block. The luminance shift values are divided in two groups: μ_A and μ_B, as shown in Fig. 7, where μ_C is the union of μ_A and μ_B. Each bit of the watermark is embedded during fractal codification as follows: a range block is selected, if a bit 1 is to be embedded, this block is codified by searching the best luminance shift value in μ_A. Otherwise, if a bit 0 is to be embedded, the range block is codified by searching the best luminance shift value in μ_B. As in Puate's scheme, when the watermark has been embedded and there are range blocks left, the remaining range blocks are codified by searching in μ_C.

A property of fractal image codification is called *fix point in fractal codification* and refers to the fact of applying iteratively fractal compression on an image until the initial image (previously codified) is the same as the fractal codified image, in other words, until the distance between both images is zero and fractal parameters are the same. This property was used by Zhen Yao in [14] as a way to check data integrity by means of a watermark. Thus, if the watermarked image is modified, the fractal parameters will not match the initial fractal parameters before the attack. In the same paper, Zhen proposed another scheme to hide information during fractal image codification process. This scheme utilizes a local search region of domain blocks for each range block, as shown in Fig. 8.

For the embedding process, the local search region is divided in two parts: LSR-0 and LSR-1; they are used to search a domain block for each range block *Rb* (with minimum distance) and, depending on the value of the bit to be embedded, a specific zone is used. Thus, if a bit 1 or 0 is to be embedded, the search will be on LSR-1 or LSR-0, respectively. If the domain block gets a better approximation in the region of the opposite bit, a block exchange is performed.

Hong Pi, Hung Li and Hua Li proposed a watermarking scheme that hides a binary watermark into image files compressed by fractal codification [15]. To perform the embedding process the fractal codes are extracted from the carrier image by means of the fractal codification described in [16]. The main difference between this fractal codification and the one described in Section 2 is the use of the range block mean $\bar{R}_u$ instead of the contrast scaling S_u. Then, a watermark w (*m-sequence*) of size N and period P is generated, where N and P are equal to the number of range blocks detected in the fractal codification. A condition for the watermark is that it must end with a 0. The next step is to add the watermark (it can be permuted to increase security) with the quantized range block means $\bar{r}_u$ (quantization of 7 bits) to produce the watermarked fractal codes. To hide the watermark in the carrier image, the fractal decoding process is performed. The fractal decoding process diffuses the watermark throughout the decoded image, this is the watermarked image. Given the range block means $\bar{r}$, the watermark extraction process begins with computing the average of all range blocks of the attacked images $\bar{r}_u^*$, then the attacked watermark $w_u^* = \bar{r}_u^* - \bar{r}_u$ is computed and, if necessary, w_u^* is permuted back. Then, the correlation coefficients between the permuted w_u^* and P shifted m-sequences are computed. Finally, the unique peak is found and the existence of watermark is determined by checking whether or not the peak exceeds a predefined threshold T (in this article, $T = 0.2$). This watermarking scheme only detects the existence of the watermark.

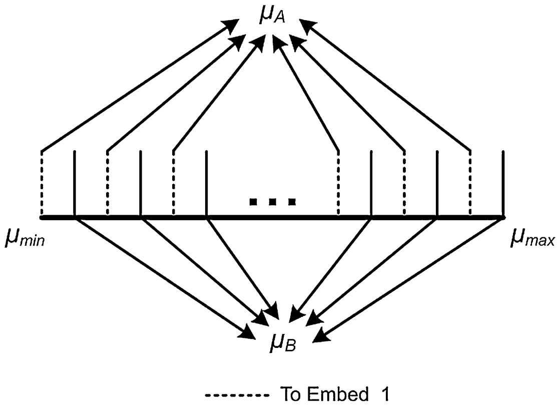


Fig. 7. Luminance shift value range and its groups.

Other watermarking scheme that hides the watermark in the mean of the range blocks is proposed by El-Khamy et al. [17]. The embedding process begins with a classification of range and domain blocks regions (8 regions for this work). For range and domain blocks equal number of regions represents a 0 or a 1. Some range blocks are then chosen to embed the watermark. When the bit to be encoded is 0, the best pair of domain blocks is fetched in regions that represent 0 only. An alien range block is the one chosen for hiding, for instance, 0 while its affiliation is to a region representing 1. This case is called range block/region mismatch. In case of a range block/region mismatch, bit padding technique is used where only the indexes of the padded bits are stored. At the watermark extraction process, only the punching pattern is required to extract the original watermark, discarding the padded bits. To extract the watermark, the image is decoded, then the range blocks previously used to hide the bits are reclassified in the decoded image. According to their means, their affiliations are identified and thus the hidden bits. The extracted watermark is then punched using the punching pattern to extract the final watermark.

In [18], the range blocks are selected randomly to hide the watermark. Then, fractal coding is performed for data embedding. If a watermark bit is 1, only the search space G1 will be considered. Otherwise, if a watermark bit is 0, only the search space G0 will be chosen for fractal watermarking. G1 and G0 represent the domain regions.

The selection of block sets is a common task in several watermarking schemes inspired by fractal codification. Generally, there are two block sets to hide 0 and 1 of the message, respectively. In [19] a novel selection of block sets is proposed. This method utilizes the fuzzy C-mean clustering to classify all domain blocks to four classes (A, B, C, D). Class A and B are used to hide the bits 0 and 1 respectively. Class C and D are not considered for hiding the message.

Some other watermarking schemes combine the use of fractal codification with the discrete wavelet transform (DWT). In [20] both the range and domain regions are coefficients of the first level DWT. The embedding is done by substituting a range block with a new matching block. This idea is used in [21], in this work more DWT levels are used to hide the watermark. Moreover, the

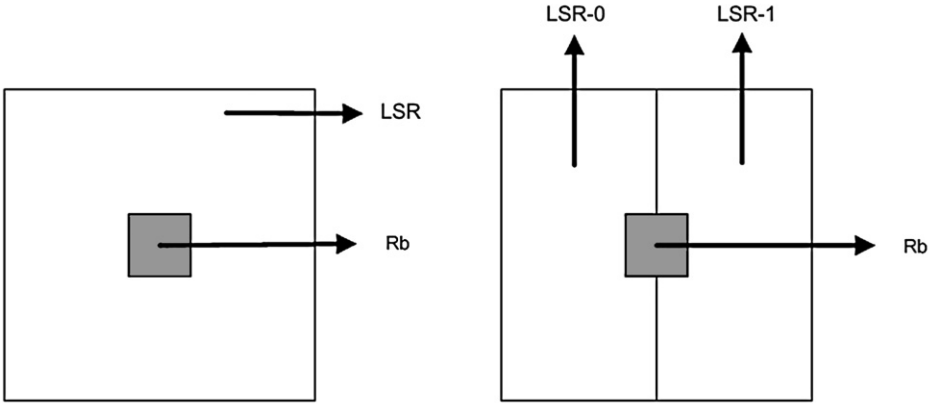


Fig. 8. Local search regions proposed by Zhen.

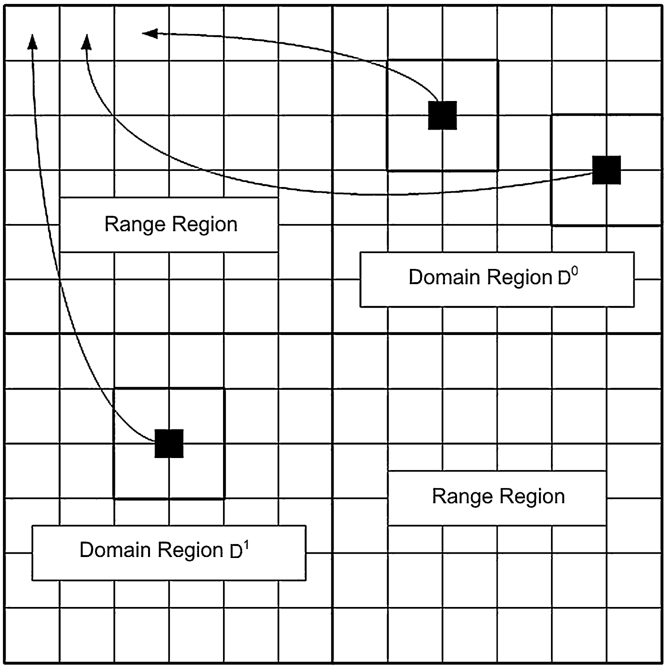


Fig. 9. Image regions proposed by Gulati.

value of the JND (just-noticeable difference) is calculated in order to diminish the distortion generated by blocks substitution. Another scheme which combines these two techniques is proposed in [22]. In this scheme the watermark is embedded in the middle frequency band LH3 of DWT.

In the same way, there are watermarking schemes which combine the use of fractal codification with the discrete cosine transform (DCT). In [23] a fractal watermarking scheme robust to geometric attacks is proposed. To achieve this robustness, the geometrically invariant space is constructed by using image normalization. Then, a significant region is obtained from the normalized image by utilizing the invariant centroid theory. Finally, the watermark is embedded into the modified self-similar DCT blocks of the significant region. Another scheme which combines the use of fractal codification with the DCT transform is proposed in [24]. In this scheme the DCT coefficients of the range blocks are modified to hide a single bit.

Unlike the previous schemes, there are a number of watermarking schemes that do not perform the embedding process using fractal compression, instead they embed the watermark using similarities between range and domain blocks. The schemes proposed by Kamal Gulati in [25], modify range blocks, domain blocks and luminance to generate a new watermarked block.

The first of Gulati's schemes divides the carrier image in range and domain regions as shown in Fig. 9. The range blocks are located on quadrants I and III. The embedding process is performed in a very similar way to Zhen's scheme: if a bit 1 is to be hidden, the modified domain block, which resembles more the range block, must be part of domain region D^1; if a bit 0 is to be hidden, the domain block must be part of domain region D^0. Fig. 9 shows an example of the '100' watermark embedding.

The second scheme also uses the regions of blocks as shown in Fig. 9. The domain blocks are modified pixel by pixel except for the first pixel that remains unchanged. This modification is required only if all of the pixels are different to the first one. At the end of this process, the modified domain block will resemble more the range block. Then, the range block is substituted by the modified domain block, that comes from a domain block that is part of a domain region D^1 or D^0. The bit value depends on the region the domain block was taken from.

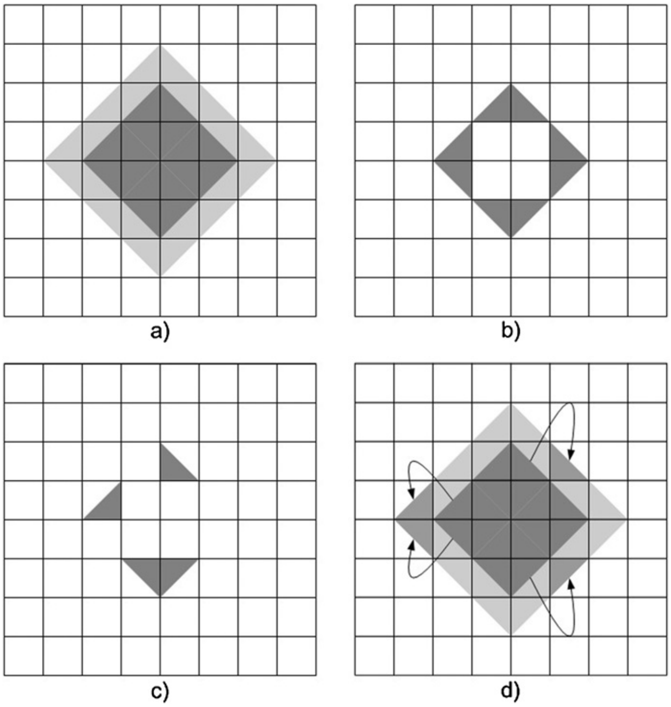


Fig. 10. (a) Partitioned image. (b) Detected domain blocks. (c) Domain blocks after quantization. (d) Range blocks detected by similarities search.

The third scheme only considers the least significant bit (LSB) of pixels. The LSBs of the pixels in a range block are compared with the LSBs of the pixels in each domain block of the selected quadrant (D^1 region or D^0 region). The domain block with the minimum distance is chosen and its LSB plane is copied over the LSB plane of the range block. In this manner, the LSB plane of the range block will become the same as the LSB plane of the domain block.

Another scheme inspired by fractal codification is proposed by Patrick Bas, Jean Marq Chassery and Frank Davoine in [26,27]. This scheme replaces range blocks by modified blocks according to the bit to be embedded. To select the range block to be replaced, as much domain blocks as bits to be embedded are firstly selected. These domain blocks are selected from the carrier image, detecting only those with high standard deviation, then they are quantized to limit even more the selection into dissimilar blocks. Once a set of domain blocks is selected, a similarities search in the entire image, excepting the domain blocks, is performed in order to find a set of blocks that resembles more each of these domain blocks. The blocks found are the range blocks.

The modified blocks that replace range blocks are obtained by generating an approximated version of them. For every range block R, its approximated block, called $\hat{R}$, is calculated by:

$$\hat{R} = \delta \cdot S \cdot \left(\frac{D}{\max(D)}\right) + \bar{R} \tag{3}$$

where $\bar{R}$ is the mean of R, S is the factor of the watermark magnitude, $\max(D)$ represents the maximum value of D and

$$\delta = \begin{cases} +1 & \text{if a bit '1' is embedded} \\ -1 & \text{if a bit '0' is embedded} \end{cases}$$

The replacement of range blocks by their approximations is the embedding process and is repeated until the entire watermark is embedded. This process is shown in Fig. 10.

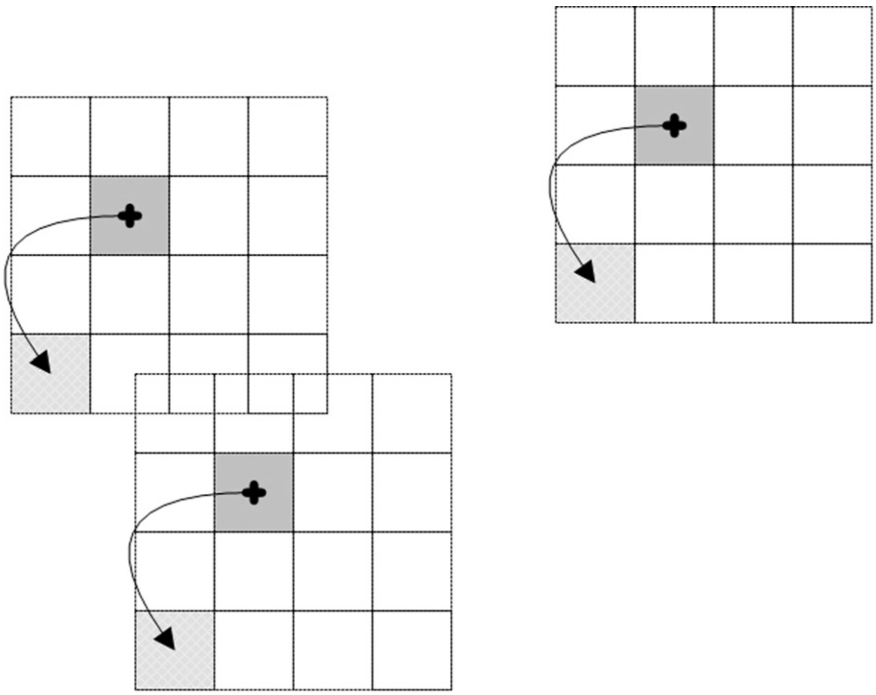


Fig. 11. Similarities embedding in a local window.

The *detection* process in this scheme verifies the correctness of the extracted watermark; it requires a reference to the range blocks where the bits were embedded. The *extraction* process is similar to the watermark embedding process. Both processes, detection and extraction, are performed as follows:

1. Obtain a domain block D from the image.
2. Create an approximated block $\hat{R}$ with Eq. (3) and search the range block R that minimizes *RMS* error:
   (a) If the index of R is located on the range blocks references, the bit is detected and its value depends on the sign of δ.
   (b) If the index of R is not located on the range blocks references, the bit is not detected.
3. Obtain another domain block D from the image and return to step 2 until the extraction is finished.

Other watermarking schemes created also by Patrick Bas, Jean Marq Chassery and Frank Davoine [28], unlike the previous scheme, they utilize range blocks and domain blocks in the watermark embedding and extraction processes. The embedding process starts with the domain blocks selection through a characteristic point detector (Harris detector), such that a characteristic point is the center of the domain block. For each domain block, marked with a cross, a 4 × 4 local window is created and a range block is selected, as seen in Fig. 11. This selected range block is modified and replaced by a reduced version of the domain block (the watermark), in this case the range block is located in the inferior left corner of the local window. The watermark extraction process is performed in a similar way to the embedding process. Domain blocks are obtained, the local window is created, and a temporary watermarked block is built and compared with each range block of the local window.

Other schemes were inspired by image fractal compression theory. In [29] a scheme that hides fractal parameters (the watermark) in an image is proposed. A more ambitious scheme, proposed by Khadivi in [30], hides a watermark in the parameters of an IFS, thus the watermark is transformed into a fractal image. The extraction process is performed using the Collage theorem to obtain the IFS that generates such fractal image [8]. With the IFS, the parameters and consequently the watermark can be obtained. A different scheme that uses fractals although not using the idea of similarities between range and domain blocks to hide the mark is [31]. It presents an extension of the QIM scheme in the context of valumetric distortions. It uses a fractal quantization structure but also a content-dependent quantization grid to achieve both global constant robustness and the ability to recover the watermark after non-linear valumetric is content-dependent.

Fragile watermarking schemes were inspired by image fractal compression theory. In [32] an algorithm about restorable and fragile watermarking based on fractal compression and differentials record theory is proposed. In general, the compressed image (by fractal codification) is stored in the B component to restore the colorful image. The R and G components are processed to detect any change in the image. In [33], the fractal compression codes of some interest regions are embedded into specific locations of the image. This scheme uses a tampering detection method. If any change is detected, the recovery is achieved by using the fractal codes hidden in the same image.

One of the main characteristics of some of these fractal codification inspired works is that they distort an image block (domain or range block) to hide watermark bits. However, this distortion does not represent the information; the information is represented by the position of the distorted block. Also, just one bit can be embedded in a single block. This characteristic may be seen as a disadvantage since the watermarked image is greatly distorted and the embedding capacity is considerably reduced when the watermark is embedded. However, this intrinsic characteristic is important to keep robustness on these algorithms. The scheme proposed in this article hides information in an image block instead of the block position in order to reduce the visual artifacts caused by block replacement.

4. Proposed watermarking scheme

The proposed scheme is based on Patrick Bas et al. scheme [26, 27]. The general diagram of this new scheme is shown in Fig. 12, where the embedding, extraction and detection processes can be observed.

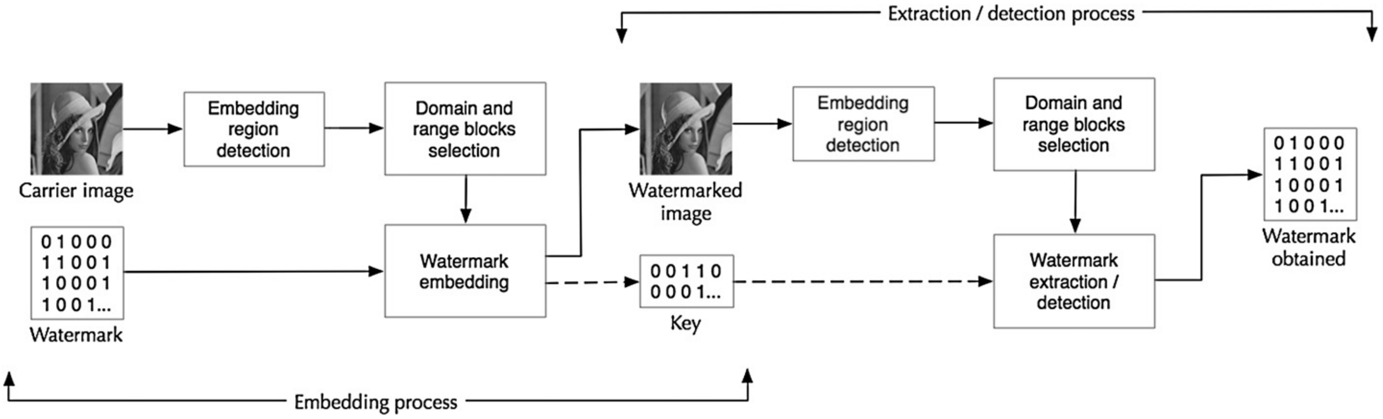

Extraction / detection process
Carrier image
Embedding region detection
Domain and range blocks selection
Watermark embedding
01000
11001
10001
1001...
Watermark
Watermarked image
00110
0001...
Key
Embedding region detection
Domain and range blocks selection
Watermark extraction / detection
01000
11001
10001
1001...
Watermark obtained
Embedding process

Fig. 12. Diagram of the proposed scheme.

Image block processing increases the computational complexity of the algorithm to such a degree that it may prevent its practical use. The previously mentioned schemes process a big amount of blocks. The proposed scheme reduces block processing in several ways. One of them is by using the Harris detector (considered a good point detector, resistant to compression, filtering and geometric attacks [34]) to select the domain blocks; another way to reduce block processing is by limiting the blocks search due to the use of LSR. Moreover, to reduce watermarked image distortion, the Bas embedding algorithm was modified, and to improve the robustness the watermark is embedded redundantly.

The embedding and extraction/detection processes of the proposed scheme are presented next.

4.1. Watermark embedding process

The embedding process of the watermark is carried out in two stages:

1. Selection of k embedding regions RI_k, in which the watermark M will be embedded.
2. Insertion of the watermark M in an embedding region RI_k, this is:

   $\{RI_1, RI_2, \dots, RI_k\} \in I$

A watermark of M_n length will be hidden in each embedding region RI_i

$$M = m_1 m_2 m_3 \dots m_{M_n} \quad \text{with } m_i \in \{0, 1\}$$

The redundancy of the watermark is defined by k.

The selection of k embedding regions RI_k is described as follows:

1. Selection of characteristic points. *The image I_p is created by detecting the characteristic points of image I.*
2. Selection of an embedding region. *Select the point $P_{\max}$ of greater magnitude of I_p. The position of point $P_{\max}$ indicates the center of the embedding region RI_i of $N_B \times N_B$ size, being N_B the number of non-overlapping blocks B_i of $n \times n$ pixels each.*
3. Search for other embedding regions. *Search in I_p for another possible embedding region RI_{i+1}.*

The embedding process of the watermark M in an embedding region RI_i is performed as follows:

1. Partition of the embedding region. *The embedding region RI_i is divided into $N_B \times N_B$ non-overlapping blocks B of $n \times n$ size.*
2. Selection of domain block set. *The domain block set D is built in two steps: detection and quantization. The cardinality of the domain block set is $|D| = Mn$.*
   (a) Detection of domain blocks. *The image RI_p is created by detecting the characteristic points in RI_i. Then the partition of image RI_p in $N_B \times N_B$ blocks B_i is performed with $i = \{1, 2, \dots, (N_B)^2\}$ of $n \times n$ size. The set of domain blocks $D' \in B$ is built with the selection of blocks B_i with characteristic points.*
   (b) Decimation of domain blocks. *The set of domain blocks D' is decimated by eliminating blocks that are alike, forming the set of domain blocks $D \subseteq D'$.*
3. Selection of range block set. *For every domain block $D_i \in D$, an LSR of size $N_{LSR} \times N_{LSR}$ that does not overlap blocks B_i is built. In this LSR a range block R_i will be detected and bit i of the watermark will be embedded. The block B_i that resembles more the domain block D_i is called range block R_i. Then, each range block $R \in LSR$ will be modified using Eq.* (4) *to generate a modified block $\hat{R}$. The embedding equation is:*

   $$\hat{R} = \delta \cdot S \cdot \left(\frac{D_i - \bar{D}_i}{\max(D_i - \bar{D}_i)} \right) + \bar{R} \tag{4}$$

   where $\bar{R}$ is the mean of $R \in LSR$, $\bar{D}_i$ is the mean of D_i, $\max(x)$ is the maximum value of x, the magnitude of the watermark is $S = 2 * DevStd(R)$, where DevStd is the operation of standard deviation and

   $$\delta = \begin{cases} +1 & \text{if a bit '1' is embedded} \\ -1 & \text{if a bit '0' is embedded} \end{cases}$$

4. Watermark embedding. *Given the range block set R, the next step is the watermark embedding. Each range block $R_i \in R$ is replaced by $\hat{R}_i$. The position of this block is stored in an indexes vector that will be used later on during the detection process.*

The watermark embedding process is performed for every embedding region RI_i. The number of regions depends on the image characteristics, for example, non-homogeneous images will have a high number of characteristic points and thus the number of RI will be high. And this is directly related to the embedding capacity and robustness of the scheme.

4.2. Watermark extraction and detection processes

The extraction process is similar to the watermark embedding process, it is performed in two stages: (1) selection of embedding regions, and (2) extraction and detection of the watermark from each region. With the detected and extracted watermarks from each region, a procedure to determine the magnitude of the final extracted watermark of the system is performed next.

The selection of the embedding regions *RI* is performed as in the embedding process, considering same conditions as N_B, n values and restrictions for the Harris detector.

The extraction process from an embedding region *RI* requires the magnitude of n and N_{LSR}, and restrictions for the Harris detector. The detection process requires the indexes of the selected range blocks from the embedding process to determine the existence of every bit.

After *RI* selection, the processes of extracting and detecting the watermark are performed as follows:

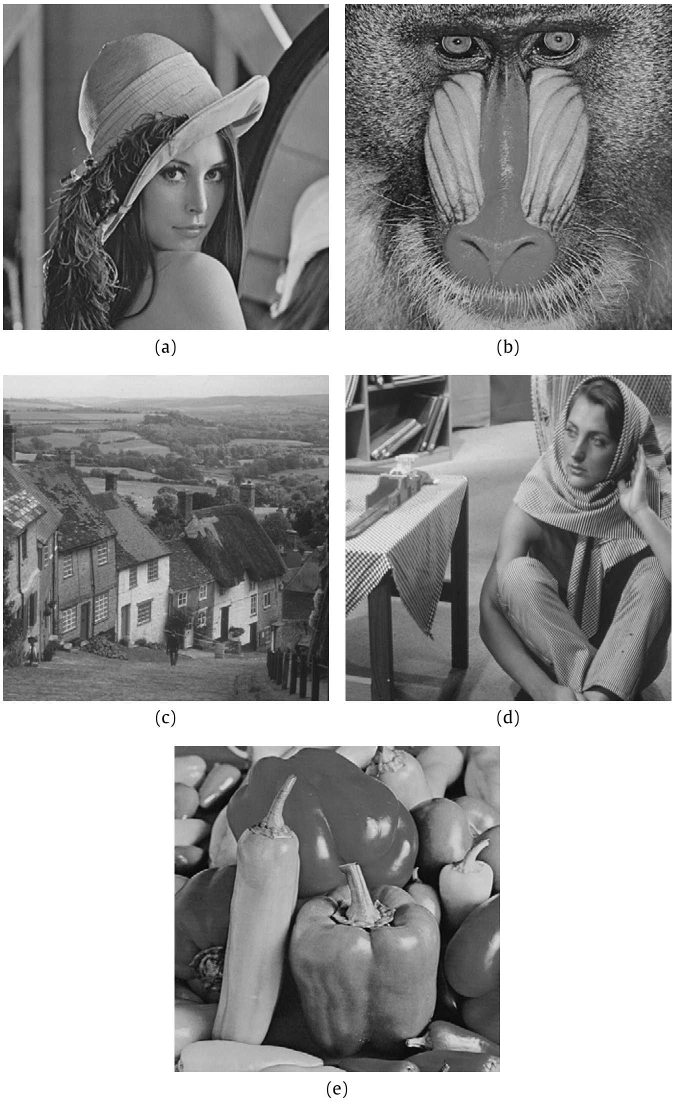
1. Partition of the embedding region. *Embedding region RI is divided into $N_B \times N_B$ non-overlapping blocks of size $n \times n$.*
2. Selection of the domain block set. *Domain block set D is formed in two steps: detection and quantization.*
3. Selection of the range block set. *For each domain block D_i of D, a LSR of size $N_{LSR} \times N_{LSR}$ non-overlapping blocks B_i is built, where a range block R_i will be searched. Each one of the range blocks $R \in LSR$ will be modified through Eq.* (4) *with $\delta = +1$ and $\delta = -1$. Then a modified block $\hat{R}$, which resembles more to the generated range block R, will be selected. The position of the selected block $\hat{R}$ indicates the position of the range block R_i where a bit was embedded.*
4. Watermark extraction. *Given the range block set R, the next step is watermark extraction. This process is performed by taking into account the sign of δ for each selected range block. If $\delta = +1$, a bit* 1 *is extracted; if $\delta = -1$, a bit* 0 *is extracted. The watermark is formed with $M = m_1 m_2 m_3 \ldots m_{M_n}$.*
5. Watermark detection. *Each bit of the watermark will be verified. It is necessary to know the position of the range block where the bit was extracted from. This position is searched in the index vector for the range block generated in the embedding process. A position found indicates a bit correctly detected.*

The extracted watermark of every RI_i is built with the detected bits, non-detected bits and possibly false positive bits (as in all watermarking schemes). Considering all of these possibilities, it is necessary a bit selection to form a final extracted watermark of all *RI*. For the selection of a bit i the next steps are followed:

1. The magnitude of the i-th watermark bit is taken from the magnitude of the bit with higher percentage of appearances in the extracted watermarks of the *RI*.
2. If the percentage of appearances for both bits is the same, in other words, if the number of bits 1 and the number of bits 0 is equal, an average of E_{RMSE} errors (RMSE error of each range block R with modified block $\hat{R}$ where the bit was extracted) of each bits set is computed. The magnitude of the i-th bit is taken from the set with lower average error. For example, if the error of bits 0 is lower, then the i-th bit will have magnitude 0.

5. Obtained results

Fig. 13 shows the images that were used to carry out the experiments; such images are commonly used as test benches for image processing, specially Lenna and Baboon images. The size of the images is 512 × 512 pixels in gray scale with 8 bits depth.



(a) (b) (c) (d) (e)

Fig. 13. Test images for experiments carried out.

For each experiment, distortion and robustness are measured in terms of PSNR (Peak Signal to Noise Ratio) and BCR, respectively. The higher the PSNR the lower the distortions introduced by the scheme. PSNR measures above 38 dB are acceptable [35].

5.1. The behavior of the scheme

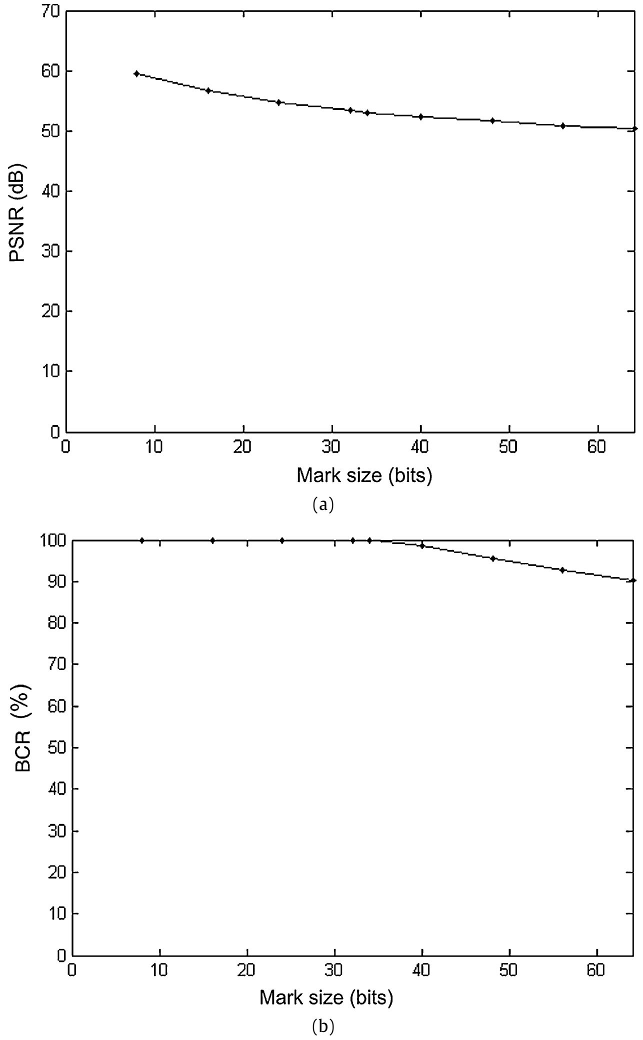
The behavior of the scheme depends on the modification of three main parameters:

1. The size of the blocks. *The size of the blocks equals $n \times n$ pixels.*
2. The size of the local search region. *The size of the LSR is $N_{LSR} \times N_{LSR}$ blocks.*
3. The size of embedding region. *The size of RI is $N_B \times N_B$ blocks.*

Each one affects the other two. After an analysis of the behavior of the scheme by modifying each one of the three parameters mentioned, it can be concluded that the scheme achieves the best results with: blocks of size 4 × 4 pixels, *LSR* of size 19 × 19 blocks and *RI* of size 29 × 29 blocks. Fig. 14 shows a graphic of the scheme behavior with respect to the size of the watermark. The watermark used was of the form $(01)^*$. The scheme detects completely small watermarks of 34 bits.

As mentioned before, the proposed scheme utilizes a key to detect the watermark. The next test shows the behavior of the scheme in the absence of the key. The size of the key increases with the N_B size and with the watermark size. Fig. 15 shows that the detection stage is a process that slightly increases the BCR of the scheme (can be unused).

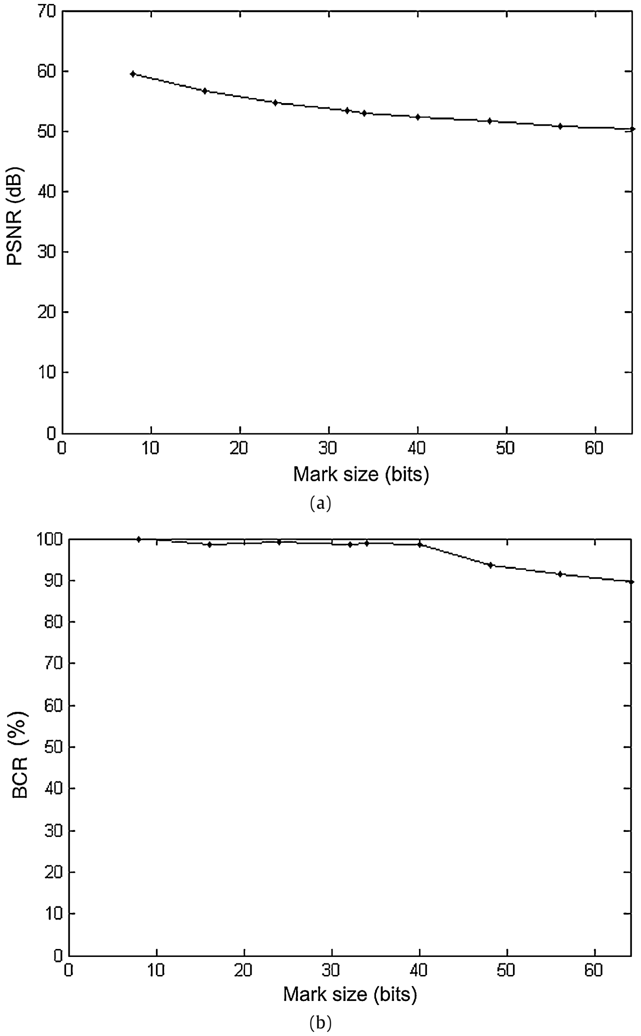
It is worth to mention that the capacity of the scheme depends on the number of detected domain blocks, in other words, the capacity depends on the characteristics of the image. In homogeneous images, capacity is lower than in non-homogeneous images because in homogeneous images there are a lot of similar regions.



(a)

(b)

Fig. 14. (a) Mark size versus PSNR and (b) Mark size versus BCR.



(a)

(b)

Fig. 15. Behaviour of the scheme in the absence of the key. (a) Mark size versus PSNR and (b) Mark size versus BCR.

Table 1
Performance of the proposed scheme against Stirmark tests.

Test	BCR (%)	Test	BCR (%)
PSNR 0	98.176	ROTSCALE −2	45.294
PSNR 10	85.294	ROTSCALE −1	52.353
PSNR 20	80	ROTSCALE −0.75	56.471
PSNR 30	80	ROTSCALE −0.5	74.118
PSNR 40	74.706	ROTSCALE −0.25	81.176
PSNR 50	65.176	ROTSCALE 0.25	84.118
PSNR 60	65.176	ROTSCALE 0.5	78.824
PSNR 70	64.059	ROTSCALE 0.75	55.882
PSNR 80	63.529	ROTSCALE 1	51.765
PSNR 90	63.529	ROTSCALE 2	50
PSNR 100	57.647	MEDIAN 3	67.059
NOISE 0	81.176	MEDIAN 5	56.471
NOISE 20	45.882	MEDIAN 7	47.647
NOISE 40	40.588	MEDIAN 9	50.588
NOISE 60	18.824	LATESTRNDDIST 0.95	50
NOISE 80	12.824	LATESTRNDDIST 1	49.412
CONV 1	50	LATESTRNDDIST 1.05	53.529
CONV 2	49.412	LATESTRNDDIST 1.1	47.059

This scheme uses a domain block detection stage that eliminates the similar blocks.

The more homogeneous the images the lower the number of embedding regions detected and thus the lower the redundancy. So, the possibility of losing a bit from the mark is higher. The opposed is also true. There is always a possibility, no matter how small it is, of losing a watermark bit. In the experiments carried out the watermark was recovered completely, not so after the attacks.

The robustness of the proposed scheme was evaluated using the Stirmark benchmark [36,37], a tool for robustness testing of image watermarking algorithms that introduces random bilinear geometric distortions to de-synchronise watermarking algorithms. The distortions have little effect on the perceptual quality of images, but are known to render most watermarks undetectable [3]. The results of the proposed scheme against Stirmark are shown in Table 1. It can be seen that the scheme is robust against some of the Stirmark attacks. As the severity of the attacks increases, the scheme losses robustness, but also, the image becomes completely distorted.

5.2. Comparison with other schemes

The improvements achieved by the proposed scheme, compared with Bas's scheme, reduce the perceptual impact on the watermarked image and increase robustness against JPEG attacks. These improvements are measured experimentally. Fig. 16 shows the performance against JPEG attack that the scheme achieves for a 34 bits watermark, range and domain blocks of 8 × 8 size and image size of 512 × 512 pixels. It can be observed that also the robustness is improved compared with Bas's scheme. Moreover, the distortion generated in the watermarked medium is lower due to the new embedding equation (4). Experimental results show the improve-

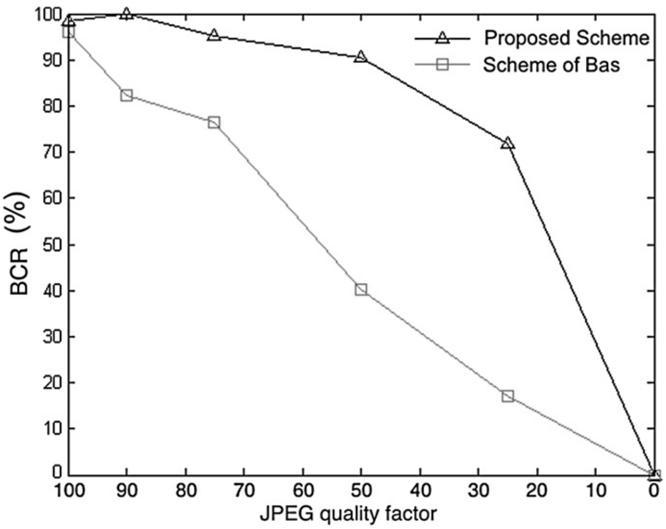


Fig. 16. JPEG comparison between the proposed scheme and the scheme of Patrick Bas.

Table 2
Comparison between the distortion generated in the watermarked blocks with the range blocks.

Block size	Block distortion (PSNR)	
	Proposed scheme	Bas scheme
4 × 4	4.92 dB	7.81 dB
8 × 8	6.47 dB	14.30 dB

ment after using the new embedding equation, this is shown in terms of PSNR in Table 2. For this experiment 1156 range blocks R were compared with the corresponding watermarked block $\hat{R}$ calculated using the proposed embedding equation (4) and the Bas embedding equation (3). It can be observed that the proposed scheme causes less distortion than Bas's scheme in all cases. This reduction is due to the use of the new normalized version of D_i. In the proposed scheme the normalized version of D_i is calculated as $\frac{D_i - \bar{D}_i}{\max(D_i - \bar{D}_i)}$ while in Bas's scheme it is calculated as $\frac{D_i}{\max(D_i)}$. The success of this improvement is due to the fact that the mean subtracted from D does not causes distortion in the normalized version (proposed scheme), since the mean represents a DC component that is not included when generating a watermarked block $\hat{R}$.

Figs. 17, 18, 19, 20, 22 and 23 show the comparisons of the proposed scheme against the schemes of Puate [11], Guanhua [13], Gulati [25], Pi [15], Yang [21] and Kiani [19], respectively, for JPEG attacks under the same conditions (same images, watermark sizes, block sizes). In Zhen Yao work [14] only results after embedding 8 bits are reported due to its high computational cost, so, to make a fair comparison the proposed scheme was also tested hiding only 8 bits, Fig. 15 shows that the proposed scheme can detect 8 bits without BCR loss. The comparison of the proposed scheme against the scheme of El-Khamy [17] for different noise types is shown in Table 3. The BCR of the proposed scheme is similar to El-Khamy's scheme despite the proposed scheme only embeds the watermark in a single *RI*. The watermark size is 176 bits.

In comparison to the scheme of Guanhua [13], the proposed scheme works better for higher JPEG quality factor. However, these results do not show the total capacity of the proposed scheme. The main contribution of this scheme is a decrease in the distortion generated by watermark embedding in the carrier image, as shown in Table 4. Here, the distortion generated by the proposed scheme is significantly lower than the schemes of Guanhua, Puate,

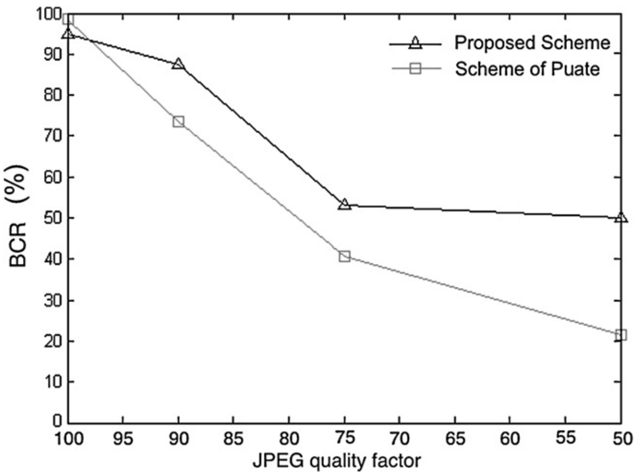


Fig. 17. JPEG comparison between the proposed scheme and the scheme of Puate. Blocks size of 4 × 4 pixels.

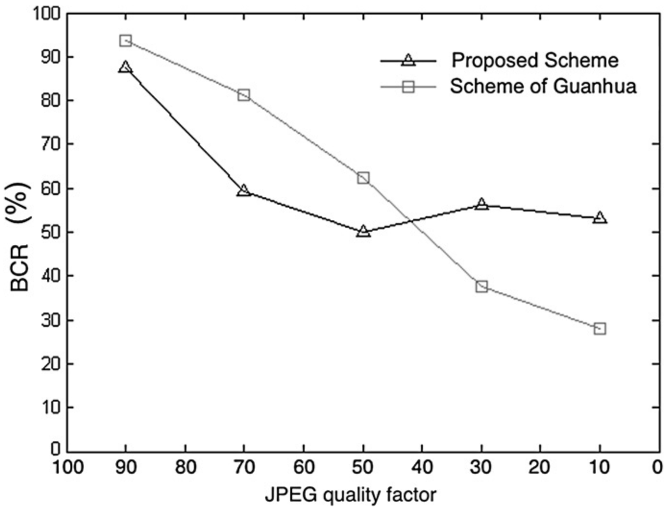


Fig. 18. JPEG comparison between the proposed scheme and the scheme of Guanhua. Blocks size of 4 × 4 pixels.

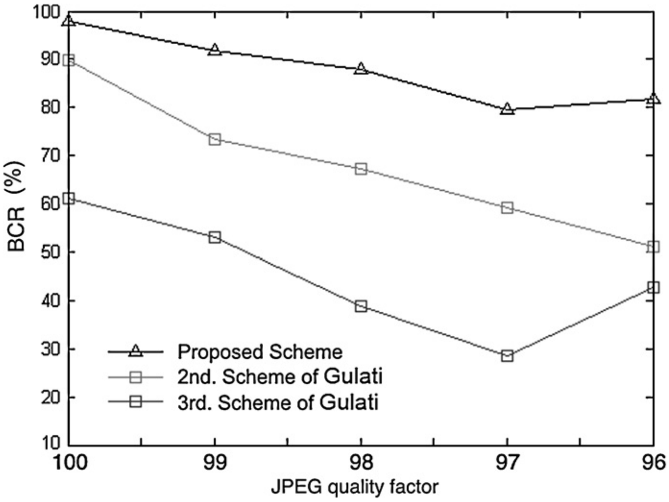


Fig. 19. JPEG comparison between the proposed scheme and the scheme of Gulati. Blocks size of 4 × 4 pixels.

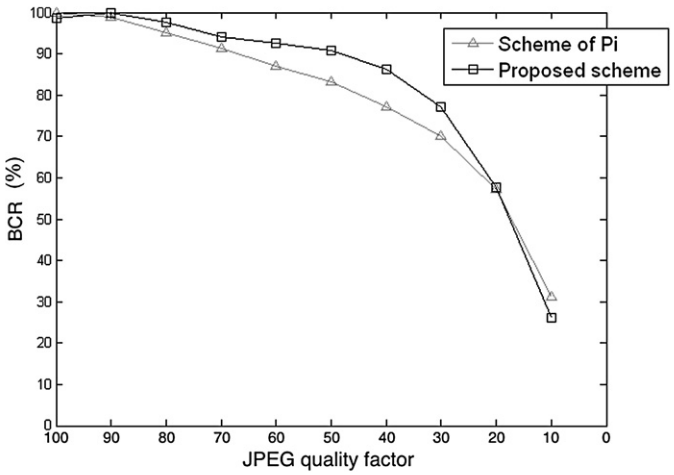
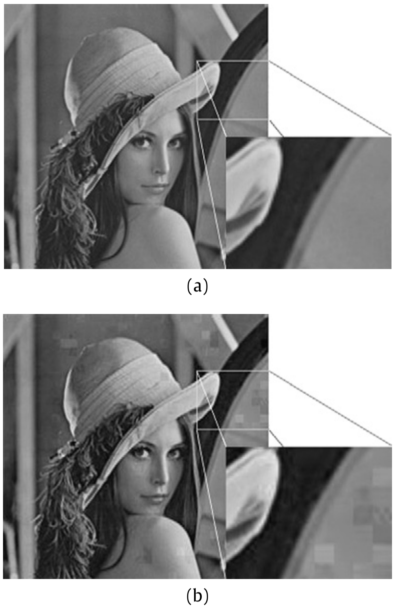


Fig. 20. JPEG comparison between the proposed scheme and the scheme of Pi. Blocks size of 4 × 4 pixels.


(a)

(b)

Fig. 21. Block artifacts generated by: (a) The proposed scheme. (b) The scheme of Patrick Bas.

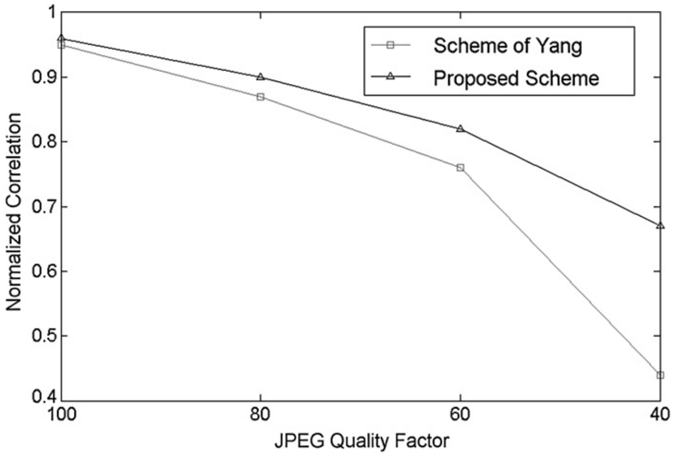


Fig. 22. JPEG comparison between the proposed scheme and the scheme of Yang. Blocks size of 8 × 8 pixels.

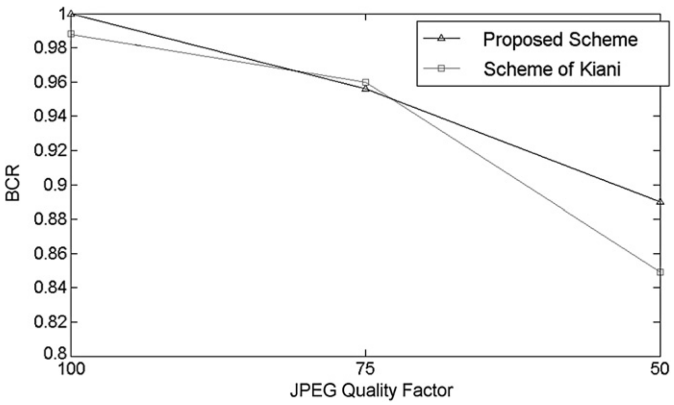


Fig. 23. JPEG comparison between the proposed scheme and the scheme of Kiani. Blocks size of 8 × 8 pixels.

Table 3
Comparison of BCR between the proposed scheme and the scheme of El-Khamy for different noise types.

Noise type	Variance	BCR	
		Proposed scheme	El-Khamy scheme
Gaussian	0.01	76.76	71.6
	0.05	64.23	68.75
	1	57.05	49.44
Salt and pepper	0.1	68.29	72.73
	0.5	57.64	56.82
	1	51.76	45.46

Table 4
Comparison between the distortion generated by the proposed scheme and the distortion generated by other schemes.

Other schemes		Proposed scheme distortion	Watermark size
Scheme	Distortion		
Puate et al.	31.5 dB (PSNR)	55.76 dB (PSNR)	32 bits
Guanhua et al.	30.05 dB (PSNR)	55.76 dB (PSNR)	32 bits
Zhen	32.87 dB (PSNR)	62.01 dB (PSNR)	8 bits
Gulati-II	52.81 dB (SNR)	53.66 dB (SNR)	49 bits
Gulati-III	52.81 dB (SNR)	53.66 dB (SNR)	49 bits
Bas et al.	39.75 dB (PSNR)	52.98 dB (PSNR)	34 bits
Pi et al.	45.53 dB (PSNR)	52.98 dB (PSNR)	16 384 (m-sequence size)

Zhen and Bas. In comparison with Gulati's scheme, the distortion generated by the proposed scheme is lower by almost 1.5 dB. In comparison with the scheme of Pi the distortion generated is nearly 8 dB, this is due to this scheme modifies the mean of all range blocks, used in the fractal codification, with a m-sequence, this watermark does not represent bits because the scheme is oriented to detection, i.e. it verifies the watermark existence. Moreover, it embeds the watermark using the fractal codification that is a very slow codification and depends on a high watermark size to guarantee the detection, furthermore, to detect the watermark it requires the mean of the range blocks codified in the embedding process. In comparison with the scheme of Bas the block artifacts are eliminated, thanks to the improvements in the embedding process, as it can be seen in Fig. 21. For the comparisons against JPEG and noise, images of 256 × 256 pixels were used, except for the schemes of Bas and Pi, where images of 512 × 512 pixels were used.

Finally, Figs. 22 and 23 show comparisons of the proposed scheme against the schemes of Yang [21] and Kiani [19]. As it can be observed, the proposed scheme performs better than these schemes against JPEG attack regardless the JPEG quality factor, except for a quality factor of 75% when the improvement of the scheme of Kiani is 0.06 in BCR. In Fig. 22 the JPEG quality factors considered are 100, 80, 60 and 40% and in Fig. 23 are 100, 75 and 50%.

6. Conclusions

This article proposed a watermarking scheme that owes its success to the adjustment of the watermark magnitude factor S to the standard deviation of the range block R. This improves the imperceptibility of the watermark avoiding block artifacts. Moreover, this scheme hides the watermark in different regions of the image, which increases robustness and does not affect imperceptibility of the embedded watermark. Also, we can conclude that local search regions assure a better correspondence between blocks, which diminishes perceptibility of the embedded watermark and increases BCR of the scheme. In addition, limiting the search of blocks in a determined area, increases embedding and extraction processes speed.

A disadvantage of the utilization of embedding regions is that the scheme may lose synchronization of the regions in a geometric attack, an adaptive disposition of blocks can be used to reach robustness against them. In [38] the utilization of triangular patterns in watermarking is proposed. A further improvement against JPEG attacks may be gained by the use of similarities in the DCT coefficients.

Acknowledgments

The authors acknowledge to CONACYT the support provided through the grant number 201778. The authors thank to MSc. Jennifer V. Marquina and Tomás Balderas for their valuable comments.

References

[1] S. Katzenbeisser, F.A. Petitcolas (Eds.), Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Inc., Norwood, MA, USA, 2000.

[2] N.F. Johnson, Z. Duric, S. Jajodia, Information Hiding: Steganography and Watermarking – Attacks and Countermeasures, Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[3] I.J. Cox, M.L. Miller, J.A. Bloom, J. Fridrich, T. Kalker, Digital Watermarking and Steganography, 2nd ed., Morgan Kaufmann, San Francisco, CA, 2008.

[4] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, Secure spread spectrum watermarking for multimedia, IEEE Trans. Image Process. 6 (12) (1997) 1673–1687.

[5] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Syst. J. 35 (3–4) (1996) 313–336.

[6] B. Chen, G.W. Wornell, Quantization index modulation: A class of provably good methods for digital watermarking and information embedding, IEEE Trans. Inform. Theory 47 (4) (2001) 1423–1443.

[7] F. Ourique, V. Licks, R. Jordan, F. Perez-Gonzalez, Angle qim: A novel watermark embedding scheme robust against amplitude scaling distortions, in: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005, Proceedings (ICASSP '05), vol. 2, 2005, pp. ii/797–ii/800.

[8] B.B. Mandelbrot, The Fractal Geometry of Nature, W.H. Freeman, 1982.

[9] M. Barnsley, Fractals Everywhere, Academic Press Professional, Inc., San Diego, CA, USA, 1988.

[10] Y. Fisher (Ed.), Fractal Image Compression: Theory and Application, Springer-Verlag, 1995.

[11] J. Puate, F. Jordan, Using fractal compression scheme to embed a digital signature into an image, in: A.G. Tescher, T. Chiueh (Eds.), Proceedings of SPIE Photonics East'96 Symposium, Boston, Massachusetts, 1996, pp. 108–118.

[12] A.E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. Image Process. 1 (1) (1992) 18–30.

[13] L. Guanhua, Z. Yao, Y. Baozong, Using the fractal code to watermark images, in: 6th International Conference on Signal Processing, 2002, vol. 1, 2002, pp. 829–832.

[14] Z. Yao, Fixed point in fractal image compression as watermarking, in: IEEE Int. Conference on Image Processing 03, Proceedings, vol. 3, Chicago, Illinois, USA, 2003, pp. 475–478.

[15] H. Pi, H. Li, H. Li, A novel fractal image watermarking, IEEE Trans. Multimed. 8 (3) (2006) 488–499.

[16] M. Pi, A. Basu, M. Mandal, A new decoding algorithm based on range block mean and contrast scaling, in: International Conference on Image Processing, ICIP 2003, vol. 2, 2003, pp. II-271-4.

[17] S.E. El-Khamy, M. Khedr, A. Al-Kabbany, Punched image watermarking: A novel fast fractal coding based technique, in: Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2007, vol. 2, 2007, pp. 629–632.

[18] P.-S. Liao, C.-C. Chen, J.-S. Pan, Robust fractal watermarking on continue-tone images, in: 18th IPPR Conference on Computer Vision, Graphics and Image Processing, 2005, pp. 1042–1407.

[19] S. Kiani, M. Moghaddam, A fractal based image watermarking for authentication and verification, in: 2nd International Congress on Image and Signal Processing, 2009, CISP '09, 2009, pp. 1–5.

[20] L. Yang, D. Sidan, Research on wavelet domain fractal coding in digital watermarking, in: IEEE International Conference on Multimedia and Expo, 2005, ICME 2005, 2005, pp. 61–64.

[21] S. Yang, Chun Xia Li, Sheng He Sun, R. Xie, A fractal watermarking scheme for image in dwt domain, in: Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007, SNPD 2007, vol. 1, 2007, pp. 364–368.

[22] S. Shahraeini, M. Yaghoobi, A robust digital image watermarking approach against jpeg compression attack based on hybrid fractal-wavelet, in: International Conference on Computer Communication and Management, ICCCM 2011, 2011, pp. 616–622.

[23] M. Xi-kui, S. Jin-guang, Z. Yu-han, A new fractal watermarking scheme based on image normalization, in: International Conference on Multimedia Information Networking and Security, 2009, MINES '09, vol. 2, 2009, pp. 149–153.

[24] Masaya Ohta, T. Yamashita, A. Sato, K. Yamashita, Digital watermarking using dct fractal coding without original image information, Electr. Eng. Jpn. 157 (4) (2006) 48–55.

[25] K. Gulati, Information hiding using fractal encoding, PhD thesis, Indian Institute of Technology Bombay Mumbai, Bombai, January 2003.

[26] P. Bas, J.M. Chassery, F. Davoine, Using the fractal code to watermark images, in: IEEE Int. Conference on Image Processing 98, Proceedings, vol. 1, Chicago, Illinois, USA, 1998, pp. 469–473.

[27] P. Bas, J.M. Chassery, F. Davoine, Self-similarity based image watermarking, EUSIPCO' 98 (4) (1998) 2277–2280.

[28] P. Bas, J.M. Chassery, F. Davoine, A geometrical and frequential watermarking scheme using similarities, in: W. Wong, E.J. Delp (Eds.), Conference on Security and Watermarking of Multimedia Contents, no. 3657, IST/SPIE, SPIE, San Jose, California, USA, 1999, pp. 264–272.

[29] C.-H. Hsieh, S.-S. Chen, Application of fractal model to visible watermarking, http://dspace.lib.fcu.edu.tw/handle/2377/2282, 2000.

[30] M.R. Khadivi, Ifs and its use in cryptography and steganography, Tech. rep., Jackson State University, Mississippi, USA, 2002.

[31] P. Bas, A quantization watermarking technique robust to linear and non-linear valumetric distortions using a fractal set of floating quantizers, in: M. Barni, J. Herrera-Joancomartí, S. Katzenbeisser, F. Pérez-González (Eds.), Information Hiding, in: Lecture Notes in Comput. Sci., vol. 3727, Springer, 2005, pp. 106–117.

[32] A. Hu, H. Xiaolong, Y. Zhigang, Z. Wuji, P. Jun, Z. Bing, A fragile watermarking algorithm based on fractal compression and differentials record theory, in: 2nd International Conference on Future Computer and Communication (ICFCC), 2010, vol. 3, 2010, pp. V3-702–V3-705.

[33] S.-S. Wang, S.-L. Tsai, Automatic image authentication and recovery using fractal code embedding and image inpainting, Pattern Recogn. 41 (2008) 701–712.

[34] P. Bas, J.M. Chassery, F. Davoine, Geometrically invariant watermarking using feature points, IEEE Trans. Image Process. 11 (9) (2002) 1014–1028.

[35] M. Barni, F. Bartolini, Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications, Signal Processing and Communication Series, Marcel Dekker, New York, NY, USA, 2004.

[36] F.A. Petitcolas, R.J. Anderson, M.G. Kuhn, Attacks on copyright marking systems, in: D. Aucsmith (Ed.), Information Hiding, in: Lecture Notes in Comput. Sci., vol. 1525, Springer-Verlag, Portland, Oregon, USA, 1998, pp. 219–239.

[37] F.A. Petitcolas, Watermarking schemes evaluation, IEEE Signal Process. Mag. 17 (5) (2000) 58–64.

[38] P. Bas, J.M. Chassery, F. Davoine, Robust watermarking based on the warping of pre-defined triangular patterns, in: EI'2000: Security and Watermarking of Multimedia Content II, San Jose, CA, 2000, pp. 99–109.

Pedro Aaron Hernandez-Avalos is graduated from Instituto Tecnologico de Minatitlan, Mexico. In 2007 he got his MSc from the Department of Computer Science at the National Institute for Astrophysics, Optics and Electronics (INAOE). He is currently pursuing his PhD degree in Computer Science at the same department. His research interests include video watermarking and steganography, as well as video and image processing.

Claudia Feregrino-Uribe is a researcher at the Computer Science Department at the National Institute for Astrophysics, Optics and Electronics (INAOE) in Puebla, Mexico. She received the MSc degree from Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV), Guadalajara and the PhD degree from Loughborough University in the United Kingdom. Her research areas are Data Compression, Cryptography and Steganography (Watermarking), Digital Systems Design, applications of FPGAs. She has served as a PC member for several conferences/workshops and as associate editor of the International Journal of Reconfigurable Computing. She is member of the Researchers National System in Mexico and is founder member for the IEEE Puebla Computer Society Chapter.

Rene Cumplido received the BEng from the Instituto Tecnologico de Queretaro, Mexico, in 1995. He received the MSc degree from CINVESTAV Guadalajara, Mexico, in 1997 and the PhD degree from Loughborough University, UK in 2001. Since 2002 he is a Professor at the Computer Science Department at INAOE in Puebla, Mexico. His research interests include the use of FPGA technologies, custom architectures and digital reconfigurable computing applications. He is co-founder and Chair of the ReConFig international conference and founder editor-in-chief of the International Journal of Reconfigurable Computing.