Pattern Recognition Letters 35 (2014) 120–129

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Watermarking in binary document images using fractal codes

Fatemeh Daraee, Saeed Mozaffari *

Electrical and Computer Engineering Department, Semnan University, Semnan, Iran

ARTICLE INFO

Article history:
Available online 10 May 2013

Keywords:
Fractal image coding
Binary document image processing
Watermarking and information hiding in documents
Document analysis
StiMark tool

ABSTRACT

This paper presents a novel watermarking method based on fractal theory. In the proposed method, information is embedded into binary document images. First, host image is coded by the proposed fractal coding method which is designed particularly for binary images. To insert the watermark uniformly over the entire host image, specific Range segments with predefined conditions are selected. Then, the watermark is added to the number of ones in the fractal code of the selected Range segments. Finally, the watermarked image is obtained by the fractal decoding procedure. Experimental results show that the output image quality of the proposed methods is acceptable to human eyes. Furthermore, empirical results show that the proposed fractal based watermarking is robust to the common attacks.

1. Introduction

As the internet becomes ubiquitous, and digitizing devices such as scanners and digital cameras become more available, individuals easily share their own resources on the web. Besides numerous advantages, such widespread information transmission threatens to ruin ownership and copyright protection. Watermarking has emerged as a technique to insert particular data into a cover media like image, text, audio, and video to detect malicious activities and authenticates ownerships. Most of the previous efforts on watermarking and information hiding embed the watermark into color and grayscale images. Because, such host images contain numerous details, the watermark can be easily inserted without noticeable visual deterioration.

Watermarking methods can be categorized into two main groups, namely spatial domain and transform domain (Chandramoul and Memon, 2001). In spatial domain methods, gray level values of the host image pixels are manipulated and information is inserted directly into them. Although spatial domain methods can be easily implemented and have high capacity, they are vulnerable to noise and attacks. On the other hand, transform domain methods not only tolerate noise and attacks but they have good-looking output. In the transform domain methods, first, the host image is converted by a predefined transformation. Then, the watermark is embedded in the transformed image or in the transformation coefficients. Finally, the inverse transform is performed to obtain the watermarked image. Since the watermark is distributed over the whole range of pixels of the host image, rather than local parts, transform domain methods are more robust to attacks. Fig. 1 shows different watermarking approaches.

Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) (Cox et al., 2008), and Spread transform (Maity and Kundu, 2011a,b) are common frequency domain methods used for watermarking. Singular Value Decomposition (SVD) is also used in conjunction with transform domain techniques for watermarking (Mansouri et al., 2009). Over the past decade, fractal coding has been exploited for image compression, pattern recognition, and watermarking and information hiding (Fisher, 1994; Pi et al., 2004).

Most of the previous efforts on fractal based watermarking used gray level host images. Best to our knowledge, this is the first attempt to utilize fractal coding to embed the watermark in a binary document image. Like other transformation based watermarking techniques, fractal methods also insert the watermark into fractal codes and perform the fractal decoding procedure to obtain the watermarked image. Fractal codes consist of several parameters such as position of best Range–Domain blocks, index of the isometric transformation which maps the Range block into its corresponding Domain block, and contrast scaling and luminance offset.

In Davern and Scott (1996), the Domain block pool is divided into two sets and the watermark is inserted in a set of selected Range blocks according to which half the best-pair Domain block belongs to. Puate and Jordan (1996) inserted a 32 bit signature into local search region for each Range block. The local search region is divided into two regions called A, and B. Every bit of the signature, s_i, was embedded with a redundancy of U Range blocks which are randomly chosen and denoted by $\{Rb\}_i$. If $s_i = 0$, $\{Rb\}_i$ is coded by searching for the best match in region A. Otherwise, region B is explored. To find the best pair Range–Domain blocks, each Range block is converted by 8 affine transformations. In Wu and Chang (2003), these transformations are divided into two subgroups. If

* Corresponding author.
E-mail addresses: daraee@semnan.ac.ir (F. Daraee), mozaffari@semnan.ac.ir (S. Mozaffari).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.04.022

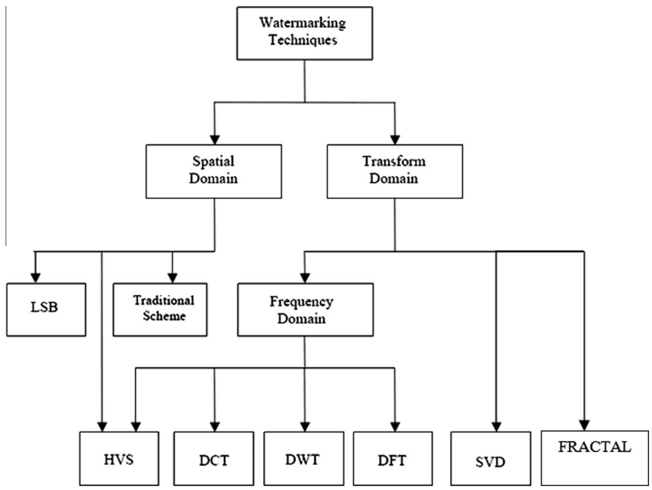

Fig. 1. Review of common watermarking methods.

a bit is zero, the corresponding Range block is encoded using the first subgroup of transformations. Otherwise, it is encoded by the second subgroup. Under malicious activity, contrast scaling and luminance offset of the original and attacked images may be different. So, the above methods try to hide the watermark by classifying search domain or index of transformations rather than utilizing contrast scaling and luminance offset parameters. In Hong Pi et al. (2006), an orthogonal fractal coding was proposed in which the Range block mean substitutes the luminance offset. Since such fractal decoding is a mean-invariant, a watermark is inserted into the quantized Range block means, followed by fractal block decoding.

2. State-of-the-art techniques and related works

In Section 1, watermarking methods in the general form are studied. This section presents a short review on recent advancements in *binary document* image watermarking and data hiding techniques (Kim et al., 2008; Chen et al., 2001). Most of the scanned and computer-generated images are presented in binary format which have only two possible values for each pixel. Binary images can be classified into halftone and non-halftone groups. Halftone images are binary representations of grayscale images mostly used in newspapers. Halftone technique simulates grayscale images through the use of dots, varying either in size, shape or spacing. This technique reduces infinite range of grey levels into a bi-level range that can be printed with only one ink. On the other hand, binary document images composing of characters, drawings, schematics, diagrams, equations are regarded as non-halftone images (Kim et al., 2008).

Proposed methods for watermarking in color and grayscale cover images cannot be used directly for binary host images. Binary images are more fragile and arbitrarily changes can be easily detected by human eyes. Since binary document images have less variation than halftone images, information hiding in these cover images needs more attention than non-halftone images. Although some transformation like DCT and wavelet can be applied on binary images, the reverse transform does not essentially results a binary watermarked image. So, most of the previous efforts focused on spatial methods rather than transformation methods.

Some of the previous methods embedded information by *shifting* lines or words in a formatted paragraph (Low et al., 1995; Chotikakamthorn, 1999). In the *partitioning* methods the document image is partitioned into $m \times n$ blocks. Each block having the predefined conditions can be used for information hiding (Wu et al., 2000; Koch and Zhao, 1995). In each language, some characters are more suitable for information hiding. These *special characters* usually composed of several parts or have complex shapes. Dots (Hassan Shirali-Shahreza and Shirali-Shahreza, 2006), sloping letters (Davarzani and Yaghmaie, 2009), diacritics (Aabed et al., 2007; Lahcen Bensaad and Bachir Yagoubi, 2011) have been used for watermark insertion. In addition to methods described above, some researchers used *syntactic and semantic* approaches for information hiding. Some punctuation signs such as full stop (.) and comma (,) in proper places are used to hide information in a text file (Bennett, 2004). These methods require identifying proper positions for putting punctuation signs and the capacity is trivial.

3. Baseline fractal coding method

Fractal image coding is based on self-similar sets and Iterated Function System (IFS) (Barnsley, 1988). It is inspired from the fact that our natural environment shows self similarity on different scales and has considerable amount of redundancy. By IFS, there would be a contractive transformation for each image that has the fixed-point resemble to the original image. In other words, applying that transform (T) iteratively on an arbitrary initial image, the result converges to the original image. In practice, after less than 10 iterations, the fixed-point image is obtained. The following equations describe IFS system (Fisher, 1994)

$$I_{n+1} = T(I_n) \quad (1)$$

$$I = \lim_{n\to\infty} T(I_n) \quad (2)$$

$$T(I) = \bigcup_{i=1}^{n} T_i(I) \quad (3)$$

In which, T is transformation, I is the processed image and n shows number of iteration. For fractal coding of an image with the size of $M \times N$, the entire image is first partitioned into B blocks, where $B = (M/b) \times (N/b)$. These non-overlapping squares are called *Range blocks* which have size of $b \times b$. A *Domain block* pool is then obtained from the original image by sliding a window of size $2b \times 2b$, starting at the top left corner of the image. Domain blocks may have overlap (step-size of δ) along the horizontal or vertical directions. For each Range block (Rb), we search the Domain block pool to find the most similar domain block (Db) and an affine transformation T which relates these blocks. Finally, the fractal coding can be formulized as:

$$Rb^{(n)} = s \times \sigma(Db^{(n-1)}) + gU \quad (4)$$

where s and g are contrast scaling and luminous offset, respectively. $\sigma(.)$ is a contractive operator to shrink the Domain blocks (Db) with the size of $2b \times 2b$ and match them with $b \times b$ Range blocks (Rb). U is a unit matrix.

4. Proposed method for binary image fractal coding

Fractal coding of gray level images is a time consuming task. This is mainly due to large numbers of sequential search through a list of Domains, needed to find the best match for a given Range block. After finding the best match for each Range block, several parameters should be determined according to Eq. (4).

On the other hand, fractal coding of binary images is faster than gray level images. Because pixel's values in Range and Domain blocks are limited to 0 and 1. Furthermore, some of parameters in the fractal code, such as contrast scaling and luminous offset,