Digital Watermarking of Satellite Images

Yogesh Chauhan
Department of Computer Sc. & Engg.
Indian Institute of Technology Kanpur,
Kanpur 208 016, India
e-mail:ychauhan@iitk.ac.in

P. Gupta
Department of Computer Sc. & Engg.
Indian Institute of Technology Kanpur,
Kanpur 208 016, India
e-mail:pg@iitk.ac.in

K. L. Majumder
Image Proc. & Data Prod. Group
Space Applications Center,
Ahmedabad 380 053, India
e-mail:klm@ipdpg.ipdpg.gov.in

Abstract

In this paper an efficient watermarking algorithm is proposed for copyrighting of satellite images. A look-up table method in pixel domain that does not distort certain specific regions in the original image has been used. A watermark is embedded invisibly and irreversibly in the host image without disturbing the vital areas of ones interest. This watermark is embedded in such a way that it can be easily extracted on the production of the watermarking key.

1. INTRODUCTION

Digital image watermarking is a method of embedding information in an image in such a manner that it cannot be removed. This watermark can be used for ownership protection, copy control and authentication. Any sort of copyright infringement forms a legal basis for prosecution. An effective watermarking technique for satellite images should have the following features:

1) *Imperceptible* : The watermark should be imperceptible to the naked eye.
2) *Undeletable :* The watermark must be undeletable, at least without visibly degrading the original image.
3) *Statistically Undetectable* : A pirate should not be able to detect the watermark by comparing several watermarked signals belonging to the same author.
4) *Unambiguous* : Retrieval of the watermark should unambiguously identify the owner.
5) *Easy Decodable :* The watermark should be readily detectable by the proper authorities.
6) *Selective :* The watermarking technique should not distort certain specific areas in the image.
7) *Blind :* The watermark extraction should not require the original image

The image watermarking algorithms can be classified into two categories: spatial domain techniques (spatial watermarks) and frequency domain techniques (spectral watermarks). The spatial domain techniques directly modify the intensities or color values of some selected pixels while the frequency domain techniques modify the values of some transformed coefficients. The simplest spatial-domain image watermarking technique is to embed a watermark in the least significant bits (LSB) of some randomly selected pixels. The watermark is invisible to human eyes but the watermark can be easily destroyed if the watermarked image is low-pass filtered or JPEG compressed. Schyndel *et. al.* [11] have proposed such an LSB-manipulation algorithm but the technique is non-blind. To increase the security of the watermark, Matsui and Tanaka [5] have proposed a method that uses a secret key to select the locations where a watermark is embedded, e.g. the use of a pseudo-random number generator to determine the sequence of locations on the image plane. Voyatzis and Pitas [2] have used a total automorphism approach to scramble the digital watermark before a watermark is inserted into an image. Bruyndonckx [9] has proposed a scheme based on pixel region classification. Pixels are classified as pertaining to regions of hard, progressive or noise contrast. Then, the pixels have their gray levels changed following a rule that takes into account the region where the pixel is inserted and the value of the bit to be embedded. Kutter *et. al.* [6] have introduced a new blind watermarking method based on 2-D amplitude modulation. In this method, single watermark bits are multiply embedded by modifying pixel values in the blue channel. These modifications are proportional to the luminance and either additive or subtractive, depending on the value of the bit. This new method is resistant to both classical attacks such as filtering and geometrical attacks. There exist a good number of algorithms which increase the robustness of the watermark. But all modify some properties of selected pixels or blocks. Darven and Scott [10] have proposed a fractal-based steganographic method to embed the watermark. In this method, there is a visual key that specifies the position of the range and domain regions containing the message. Lee and Lee [1] have suggested an adaptive pixel-domain image watermarking technique that is robust to common image processing operations such as low-pass filtering and JPEG compression. The proposed approach utilizes the sensitivity of the human visual system to adaptively modify the intensities of some pixels in a block. The modification of pixel intensities depends on the content of a block. If the contrast of the block is high (e.g. an edge block), the intensities can be changed greatly without introducing any distortion to human eyes. On the other hand, if the contrast is low (e.g. a smooth block), the intensities can only be tuned slightly.

The frequency-domain techniques first transform an image into a set of frequency domain coefficients. The transformation may be discrete cosine transform (DCT), discrete Fourier transform (DFT) or discrete wavelet transform (DWT) etc. The watermark is then embedded in the transformed coefficients of the image such that the watermark is less invisible and more robust to some image processing operations. Finally, the coefficients are inverse-transformed to form the watermarked image. In frequency domain techniques, the embedding of even a single bit of information modifies all the pixels of the host image. Hence, these techniques may not satisfy the requirement of selectivity for watermarking of satellite images. Wu and Liu [7] have proposed a look-up table [LUT] based technique and is based on block-DCT transform in the frequency domain.

In this pixel-mapping algorithm, the host image is watermarked in the pixel-domain rather than the frequency domain. The frequency domain is generally more popular in watermarking techniques as it makes the watermark robust against attacks such as lossy compression. Most of the available algorithms do not cater to the domain of satellite images. Their main aim is to generate a robust watermark. However, satellite images may not require the watermark to be robust as any attempt to tamper the image would result in the deterioration of the commercial value of the image altogether, rendering it unfit for reuse or further distribution. In fact, the frequency domain cannot be used because the embedding of even a single bit of information in the frequency domain may affect all the pixels of the host image, even the regions which are of interest to the image user. Also, these transforms require the computation of exponential, sine, or cosine functions whose values can only be approximated but not exactly determined. So, taking a transform of an image and then the inverse transform does not yield the same original image. Such alterations may not be tolerated in case of satellite images. The use of pixel domain offers some other advantages as well such as computationally cheaper and easier to implement.

In this paper, a pixel-domain look-up table based watermarking algorithm is proposed which does not distort certain specific regions in the original image. Unlike the fixed LUT used by Wu and Liu [7], in our proposed algorithm, we have used different LUT for every image transformation. Section 2 provides the basis of the algorithm for embedding the watermark in the host image and its extraction. The best choice for the watermarking parameters is discussed in section 3. The experimental results are presented in the next section. Improvements and extensions of the algorithm are given in Section 5. Conclusion is in Section 6.

2. PROPOSED PIXEL DOMAIN WATERMARKING SCHEME

2.1 Region of Interest

A particular organization may specifically be interested in a particular region only. For instance, a fishery is interested in the waterbodies, whilst an oil company has the plains and the sea-shores as its interest. This region, termed as the Region of Interest (RoI), is specific to the image user's needs and requirements. It specifies the range of pixel values which are of interest to the user, as these pixel values should not be modified during embedding. The pixel ranges for different areas are stored in a database by the owner organization. The different areas may be water-bodies, landmass, mountainous region, plains, clouds etc. A particular region may be selected depending on the customer's interest.

2.2 Embedding Binary Bits via Table Look-up

A look up table (LUT) is a random sequence of 0's and 1's with runs of 0's and 1's being limited in length. It also constitutes a part of the key for the watermark extraction algorithm. The process of mapping a large (possibly infinite) set of values to smaller set is called *quantization*. Every possible value of the host image pixel is quantized using a *quantization function* (Q()) to a small set of values, equal in number to the size of the LUT. The table then maps the quantized value to 1 or 0. To embed 1, the coefficient is unchanged if the entry of the table corresponding to that coefficient is also a 1. However, if the entry of the table is 0, then the coefficient is changed to the value of its nearest neighbor for which the entry is 1. We follow similar approach to embed 0. The look up function (Lookup()) function simply returns a 0 or 1 depending upon the input index,

Lookup(x) = value in LUT at index x

The LUT() function takes the pixel value of the original image (also referred to as the coefficient) as the input and maps it to a 0 or 1 depending upon the Look-up Table. Thus, the LUT() function is actually a simple composition of the lookup and the quantization functions :

LUT(x) = Lookup(Q(x))

Assume, v_i is the original coefficient, v_i' is the marked one, b_i is the bit to be embedded and LUT() is the mapping by the Look-up Table. Then the process altering a coefficient in the original image can be written as the following formula

$$v_i' = \begin{cases} v_i \text{ if } LUT(v_i) = b_i \\ v_i + \delta \text{ if } LUT(v_i) \neq b_i \end{cases}$$

where $\delta = \min_{|d|} \{d \in Z | LUT(v_i + d) = b_i\}$ and Z is set of integers.

The embedding is done by scanning iteratively each pixel b_i of the watermark and then altering a corresponding pixel v_n in the original image using a mapping in the LUT, as explained above. The procedure is depicted in the Block Diagram in Figure 1. For finding the corresponding pixel position in the original image, a prime constant N is chosen. The corresponding n^{th} pixel in the original image for the i^{th} pixel of the watermark is given by (i * N)%P. However, if the pixel belongs to the range of the RoI range or if the modified value of the pixel obtained from LUT belongs to the RoI range, then it is not altered. Therefore, the corresponding pixel for b_i is actually given by [(i+j)*N]%P, where j is the number of pixels which has been to be left unaltered as they belong to the RoI. Thus the original image has been watermarked when the corresponding original image pixel for each of the pixels in the watermark image has been modified. Based on this idea watermarking algorithm can be given as follows:

Step 1: Initialize i to 0, n to 0
Step 2: For pixel b_i in the watermark,
do the following
If $v_n \in$ RoI then n = (n+N)%P; Goto step 2;
Compute v_n' as follows :
$\delta = \min_{|d|} \{ d \in Z \mid LUT(v_n + d) = b_i \}$
if $LUT(v_n) = b_i$ then $v_n' = v_n$; else $v_n + \delta$;
If $v_n' \in$ RoI then n = (n+N)%P; Goto step 2;
Modify v_n to v_n'; set n = (n+N)%P; set i = i +1;
Step 3: Output the watermarked image

When an image has been watermarked using the above algorithm, a watermarking key is required for watermark extraction to be possible. The watermarking key K can be any invertible function (Key()) of N and LUT. Therefore the watermarking key K which has to be produced by the image owner to authenticate the ownership of the satellite image, can be defined as follows.

K = Key(LUT, N)

2.3 Watermark Extraction

The watermark can be extracted easily on the production of the watermarking key K. The size of the watermark can either be fixed by the organization or it can be obtained from the watermarking key by modifying the function Key(). The watermark W is assumed to be fixed here. The value of W and the size of the watermarked image together determine the value of N uniquely. LUT can be obtained from the key K and the value of N. Once N and LUT are known, the pixel values of the watermark can be extracted from the LUT taking into consideration the pixel values left unaltered, as either the pixel values belong to the RoI or the modified values of the pixels as obtained from table-lookup belong to the RoI. The table can be looked up as $b_i' = LUT(v_i')$ where b_i' is the extracted bit representing the i^{th} pixel of the watermark and v_n' is the corresponding pixel in the original image. As in the case of embedding, the value of n is given by [(i+j)*N]%P, where j = the number of pixels which has been left unaltered as they belonged to the RoI. The method is outlined in the Block Diagram shown in Figure 2.

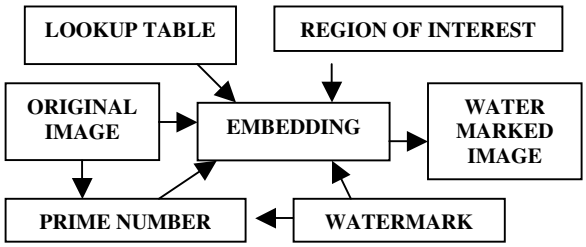

Figure 1 : Block Diagram of Embedding Process

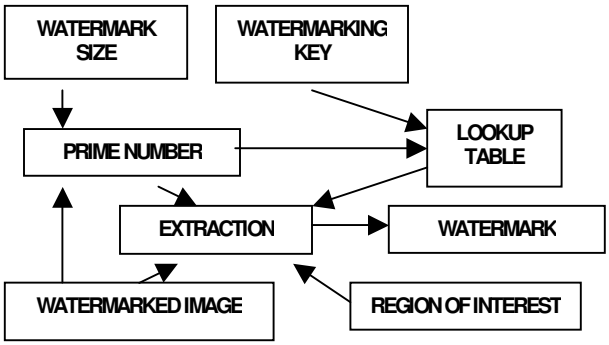

Figure 2 : Block Diagram of Extraction Process

The watermark cannot be extracted without the key. Hence, the key should be kept a secret by the image owner. In case of any dispute, the ownership can be proven by this pixel-extraction method. The extracted watermark, when it is compared with the original watermark, can be used to check if the image has been tampered with. The difference between the two images can also be used to localize the region of the original image that has been modified.The algorithm for watermarking extraction is similar to the one used for embedding:

Step 1: Initialize i to 0, n to 0
Step 2: For the pixel v_n' in the watermarked image,
do the following
If $v_n' \in$ RoI then n = (n+N)%P; Goto Step 2;
Extract the watermark b_i' as follows :
$b_i' = LUT(v_n')$; n = (n+N)%P; i = i+1;
Step 3: Output the watermark

3. CHOICE OF PARAMETERS

The various parameters for the algorithm are the watermark W, the prime number N, the Look-up Table

LUT, the quantization function Q() and the watermarking key generation function Key(). The authentication data embedded in the image should be a visually meaningful binary pattern. The watermark must be a two-pixel valued image as the binary Look-up Table can be used to embed only two different values. The size of the watermark should be sufficiently less than the original image, and it may be fixed by the owner organization. It should unambiguously identify the owner and the intended recipient. Therefore, no two customers can buy a single image and share the cost. The watermarks in the images may not be same and therefore, the customer using the other's image can be sued.

The choice of W may be fixed for a particular organization. The ideal choice of N is a prime number, which is not a factor of the size of the original image P, to avoid getting mapped to the same pixel in the original image again, in case of a wraparound. The value should be large enough to avoid the embedded data to be concentrated in a particular region. If a pixel belongs to the RoI, it is very likely that the next pixel would also be similar. So, a sufficiently large value of N also avoids such unnecessary checks. At the same time, the value should not be too large to avoid excessive wraparounds which would unnecessarily increase the computational cost of the algorithm. Hence, the ideal choice of N is determined by P and the size of the watermark W as follows :

$N=\min_{|x|}\{ x \in \mathbf{N} \mid x \text{ is a prime } > P/W \text{ and } P \neq m*x \text{ for } m \in \mathbf{N} \}$

The Look-up Table LUT should have limited continuous runs of 0's and 1's to avoid modifying the original image too much. If the runs are large, the original pixel may be modified by a noticeable amount, thereby making the watermark visible. The choice of quantization function Q() can be fixed by the owner organization such that it is not trivial to guess what the function is. There are several choices for quantizers available. The watermarking key generation function Key() is a crucial one. If an outsider is able to somehow gain the knowledge of the function being used, he can watermark the same image with his watermark in such a way that the distorted regions by the existing watermarked are left untouched. This way, he can claim the image ownership. Also, the function should be an invertible one, as the look-up table needs to be obtained from the key in watermark extraction process. A trivial choice for Key() is the multiplication function :

$$Key(N, LUT) = LUT * N$$

LUT is the decimal value of the binary sequence of the table. Given the Key K, the LUT can be easily obtained by:

$$LUT = bin(K/N)$$

where bin(x) = binary representation of the decimal number x.

4. EXPERIMENTAL RESULTS

A JPEG compressed satellite image of the western region of the Indian subcontinent of size 578x494(P= 285532) is shown in Figure 3(a). The watermark of 100x100 size is shown in Figure 3(b).

Figure 3(a) : Original Image (JPEG)

Figure 3(b) : Binary Watermark Image

Suppose the image is to be sold to an organization interested in studying the salt desert of Kutch, therefore, the Region of Interest is the white salty areas visible in the image. In JAVA, the RGB color model represents each pixel as an integer (4 bytes) with the Alpha, Red, Green and Blue(0xAARRGGBB). The first byte or the alpha value denotes the degree of transparency, 0 for a fully transparent and 255 for a fully opaque image. Thus, the value of alpha is 255 for all the pixels in the images used in this algorithm. The integer representing a pixel is being referred to as the "value" of that pixel. It is found that the pixel value range in which the salty areas lie is : -2302756 to -1. The values -2302756 and -1 are the minimum and maximum values of a pixel expected in the salty region. The Look-up Table used is a 512 bit binary pseudorandom sequence as shown in Figure 4.

The value of N taken is 29. The quantization function (Q()) used is a simple modulo function with respect to the size of the table, i.e. Q(x) = abs(x%S) where x is the input coefficient, S is the size of the LUT and abs() is the

absolute value function. The key function Key() is the concatenation function with the Lookup Table along with the prime number N. With these specifications, the image in Figure 3(a) has been watermarked to obtain the image shown in Figure 5. The difference image is shown in Figure 6.

0,0,1,1,1,1,0,1,0,0,1,1,0,0,1,1,1,1,1,0,0,0,1,1,1,0,1,1,0,0,0,
1,0,0,1,1,0,1,1,0,0,0,1,1,1,0,1,1,0,0,1,0,0,1,0,0,0,1,1,1,0,1,
1,0,0,1,0,1,1,0,0,0,0,1,0,0,0,1,1,0,1,0,1,0,1,0,0,1,1,1,0,1,0,1,1,
0,1,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,1,
1,0,1,1,0,1,1,0,0,0,0,1,0,0,1,0,1,1,1,1,1,0,1,0,1,1,1,1,1,0,1,1,1,
0,1,1,1,1,0,1,1,0,0,1,0,0,0,1,0,0,1,1,1,1,1,0,0,1,1,0,0,0,1,1,1,
1,0,1,0,1,1,0,1,1,1,1,0,1,0,1,0,0,1,1,0,1,0,1,1,1,0,0,1,0,0,0,0,
1,1,0,0,1,0,0,0,1,0,1,1,1,0,0,1,0,0,0,1,0,1,1,1,1,1,0,0,0,1,1,0,
0,0,1,0,1,1,1,1,0,0,0,1,0,1,1,1,0,0,1,0,1,1,1,1,1,0,0,1,0,1,0,0,1,
0,1,1,1,1,0,0,1,1,1,0,0,0,1,1,1,0,1,1,0,1,0,1,1,1,0,0,0,0,0,1,1,0,
1,0,1,1,0,1,1,1,0,1,0,0,1,1,1,1,1,0,0,1,1,1,0,1,1,1,1,0,1,1,1,1,0,1,
0,0,1,0,1,0,1,1,1,1,1,0,0,0,1,0,0,0,1,0,0,1,0,0,0,1,0,1,1,1,1,0,1,
1,1,0,0,0,1,0,0,0,1,1,0,0,1,1,0,0,0,1,0,0,1,0,1,1,1,0,1,0,0,1,1,
0,1,1,1,1,1,0,1,1,0,0,0,1,0,0,1,1,1,1,1,0,0,1,1,1,1,0,1,1,1,1,1,1,
0,1,1,1,0,1,1,1,0,1,1,1,0,1,0,0,0,0,1,0,0,1,1,0,0,1,1,0,0,1,1,
0,1,0,0,1,1,0,1,1,1,0,0,1,0,0, 0,0,1,1,1,0,0,1,0,0,0,0,1,1,0,0,
0,1,1,1,1,1,1,0,1,0,1,0,1,0,0,1,1

Figure 4 : Lookup Table

Figure 5 : Watermarked Image with Kutch as RoI

It is not difficult to see that the areas where the salt desert is present have not been distorted still the watermark has been embedded uniformly in the other parts of the image.The algorithm encountered 2522 pixel values that belong to the Region of Interest and therefore, has been skipped. The value of the index used to locate the pixel in the host image in which the watermark bit has to be embedded (i.e. the value of n) wrapped around just once over the pixels of the original image. The watermarking key obtained is :

3d33e3b1-363b2476-5846a75b-5552a11b-612f5eef-644f31eb-7a9ac864-5c8bc62e-2e5e52f3-8ed61add-3ceee95e-2245dc46-625a6fb1-3cf7dddd-0999a6e4-390c7d53-0000001d

The watermark of Figure 3(b) can be easily extracted from the watermarked image shown in Figure 5 with the help of this key.

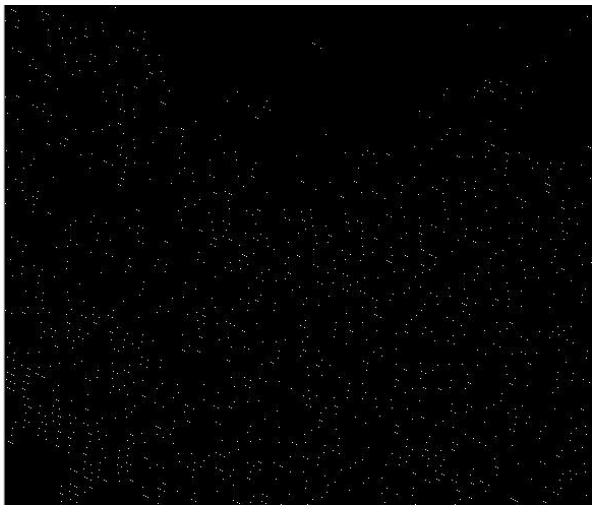

Figure 6 : Difference Image with Kutch as RoI

Now consider the case where the Region of Interest is the sea region, like when the intended recipient is a fishery. The range of RoI in this case is : -13413007 to -9068625. The other parameters have the same value as in the case of the Kutch being the RoI. The watermarked image and the difference images are shown in Figure 7 and Figure 8, respectively.

Figure 7 : Watermarked Image with sea as RoI

It is quite noticeable from the naked eye that the sea region has been left undistorted, still the watermark has been embedded uniformly in the other parts of the image, comprising of the pixels for the land region.

The algorithm encountered 4346 pixel values that belong to the Region of Interest and therefore, has been skipped. The value of n wrapped around once over the pixels of the original image. The key value obtained is same as in the previous experiment but it can be easily changed by having a different choice for the Lookup Table. The watermark of Figure 3(b) has been extracted from the watermarked image shown in Figure 7.

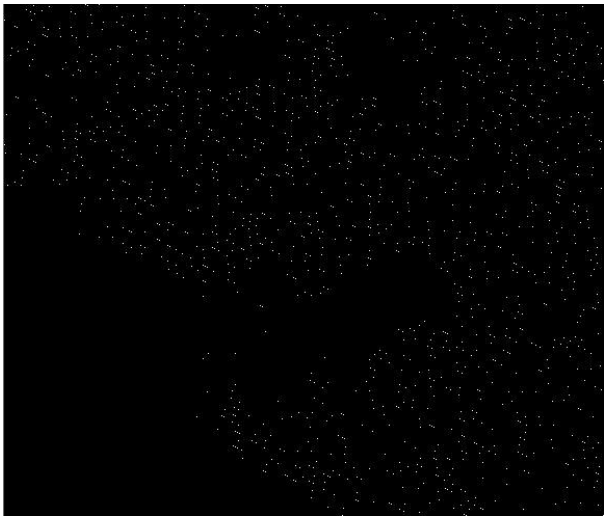

Figure 8 : Difference Image with sea as RoI

6. CONCLUSION

In this paper, a novel watermarking scheme has been presented for copyright control of satellite images. The proposed scheme can be used to watermark satellite images without distorting the vital regions that are of interest to the customer. Hence, the value of the image is preserved. At the same time, the ownership of the satellite image can be proven whenever required on the production of the key by the legal owner, thereby, keeping a check on illegal copying of the copyrighted image. It s yet to be seen whether algorithm can be improved by relaxing constraints, double watermarking and extension of LUT as the satellite image is generally much larger than the watermark image.

6.1 Relaxation of Constraints

The algorithm assumes that the image to be watermarked is much larger than the watermark itself. Also, for the algorithm to succeed, there should be a sufficiently large region in the image, which does not belong to the Region of Interest. These problems can be tackled by having lesser data to be embedded by devising a coding mechanism, compromising with the fact that the watermark may not be visually meaningful.

6.2 Double Watermarking

Double Watermarking means that the same image is watermarked twice, using different watermarks. One watermark may denote the owner while the second watermark will denote the customer.

6.3 Extension of the Look-up Table

The Look-up Table may be extended to cater to tri- or multi- colored watermarks by using a ternary or n-ary sequence instead of a binary sequence of 0's and 1's in the Look-up Table.

REFERENCES

[1] Chang-Hsing Lee and Yeuan-Kuen Lee, "An Adaptive Digital Image Watermarking Technique for Copyright Protection", *IEEE Transactions on Consumer Electronics*, Vol. 45, November 1999.

[2] G. Voyatzis and I. Pitas, "Applications of toral automorphisms in image watermarking", *Proceedings of IEEE International Conference on Image Processing*, Vol. 2, 1996.

[3] I. Pitas, "A method for signature casting on digital images", *Proceedings of IEEE International Conference on Image Processing*, Vol. 3, 1996.

[4] Joshua R. Smith and Barrett O. Comiskey, "Modulation and information hiding in images", in *Proceedings of First International Workshop on Information Hiding*, 1997.

[5] K. Matsui and K. Tanaka, "Video-Steganography: How to Embed a Signature in a Picture", in *Proceedings of IMA Intellectual Property*, Vol. 1, 1994.

[6] M. Kutter, F. Jordan and Frank Bossen, "Digital watermarking of color images using amplitude modulation", *Journal of Electronic Imaging*, Vol. 7, 1998.

[7] M. Wu and B. Liu, "Watermarking for image authentication", *IEEE Inter. Conf. on Image Processing* (*ICIP'98*), Chicago, 1998.

[8] N. Nikolaidis and I. Pitas, "Robust image watermarking in the spatial domain", *Signal Processing* special issue on *Copyright protection and Access Control*, 1998

[9] O. Bruyndonckx, J. J. Quisquater and B. Macq, "Spatial method for copyright labeling of digital images", *Proceedings of IEEE Nonlinear Signal Processing Workshop*, 1995.

[10] P. Davern and M. Scott, "Fractal based image steganography", in *Proceedings of First International Workshop on Information Hiding*, 1997.

[11] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A Digital watermark", *Proceedings of IEEE International Conference on Image Processing*, Vol. 1, 1994.