A SPATIAL WATERMARKING ALGORITHM FOR DIGITAL IMAGES

Dumitru Dan BURDESCU, Liana STANESCU

University of Craiova, Faculty of Automation, Computers and Electronics
burdescu@topedge.com ; stanescu@nt.comp-craiova.ro

Abstract: *The rapid growth of digital multimedia technologies brings tremendous attention to the field of digital watermarking. The owner or the distributor of the digital images can insert unique watermark into copies for different customers or receivers, which will be helpful to identify the source of illegal copies. Watermarking embeds a secret message into a cover multimedia data. In media watermarking the secret is usually a copyright notice and the cover a digital image. In digital watermarking, robustness is still a challenging problem if different sets of attacks needed to be tolerated simultaneously. In this paper we present an original spatial watermarking technique for digital images. Our approach modifies blocks of the image by a spatial watermark insertion. Spatial mask of suitable size is used to hide data with less visual impairments. Watermark insertion process exploits average color of the homogeneity regions of the cover image.*

Keywords*: watermarking, robustness, multimedia, JPEG, algorithm*

1. INTRODUCTION

In the recent time, the rapid and extensive growth in Internet technology is creating a pressing need to develop several newer techniques to protect copyright, ownership and content integrity of digital media. This necessity arises because the digital representation of media possesses inherent advantages of portability, efficiency and accuracy of information content. On the other hand, this representation also puts a serious threat of easy, accurate and illegal perfect copies of unlimited number. Unfortunately, the currently available formats for image, (also audio and video) in digital form do not allow any type of copyright protection. A potential solution to this kind of problem is an electronic stamp or digital watermark, which is intended to complement cryptographic process. While the later techniques facilitates access of the encrypted data only for valid key holders but fails to track any reproduction or retransmission of data after decryption. On the other hand, in digital watermarking, an identification code is embedding permanently inside a cover image, which remains within that cover invisibly even after decryption process. Digital watermarking is now considered an efficient technology for copyright protection.

Several image watermark schemes have been developed in the past few years, both spatial and frequency domains are used for watermark embedding. *Spatial watermarks* are constructed in the image spatial domain, and embedded directly into an image's pixel data. Spectral (or transform domain – based) watermarks may be incorporated into an image's transform coefficients (discrete cosine transform DCT, discrete wavelets transform DWT, Fourier transform FT). Spectral theory has emerged as an effect means of representing image signals in terms of a multi-resolution structure. Based on multi-resolution representation, a signal is divided into a number of components, each corresponding to different frequency bands. Since each component has a better frequency and time localization, the multi-resolution decomposed signal can be processed much more easily than its original representation. The multi-resolution successive approximation not only

enhances the resolution of an image, but also enhances the resolution of a watermark simultaneously [1].

The requirement of watermarking technique, in general, needs to posses the following characteristics: (a) imperceptibility for hidden information, (b) redundancy in distribution of the hidden information inside the cover image to satisfy robustness in watermark extraction process even from the cropped watermarked image and (c) possible use of one ore more keys to achieve cryptographic security of hidden content [2].

Besides these general properties, an ideal watermarking system should also be resilient to insertion of additional watermarks to retain the rightful ownership. The perceptually invisible data hiding needs insertion of watermark in higher spatial frequency of the cover image since human eye is less sensitive to this frequency component. But in most of the natural images majority of visual information are concentrated on the lower end of the frequency band. So the information hidden in the higher frequency components might be lost after quantization operation of losing compression [3]. This motivates researchers to realize the importance of perceptual modeling of human visual system and the need to embed a signal in perceptually significant regions of an image, especially if the watermark is to survive losing compression [4]. In spatial domain block based approach, this perceptually significant region is synonymous to low variance blocks of the cover image.

Since the meaning of multimedia data is based on its content, it can modify the multimedia bit-stream to embed some codes, i. e. watermarks, without changing the meaning of the content. The embedded watermark may represent either a specific digital producer identification label, or some content-based codes generated by applying a specific rule. Because the watermarks are embedded in the data content, once the data is manipulated, these watermarks will also be modified such that the authenticator can examine them to verify the integrity of the data. For complete verification of uncompressed raw multimedia data, watermarking may work better than digital signature methods because: (a) the watermarks are always integrated with the data such that the authenticator can examine them conveniently, and (b) there are many spaces in the multimedia data to embed the watermarks without degrading the quality too much [5].

However, there is no advantage to use the watermarking method in a compressed multimedia data for complete verification. Compression standards e.g. JPEG or MPEG have user-defined sections where digital signature can be placed. Because multimedia data are stored or distributed in the file format instead of pixel values, therefore the digital signature can be considered as being "embedded" in data. Once the multimedia data is modified, the user-defined section of the original data is usually discarded by the editing software. Even if the digital signature can be reserved by the software, it can easily detect the modification, since the hash values of the modified data will not be as the original. Moreover, because there is less space for compressed multimedia to hide watermarks, if it does not want to sacrifice too much visual quality on the multimedia data, there may not be enough information bits to protect the data. For content verification, a watermarking method that can reliably distinguish compression from other manipulations still has not been found. The watermarks are either too fragile for compression or too flexible for manipulation.

It is found in the literature that the robust watermarking systems proposed so far can only withstand some of the possible external attacks but not all. The attacks against the watermark try to neutralize the watermark, without damaging the image too much. The watermark is neutralized if: (a) the detector cannot detect the watermark (distortion, attenuation etc.), (b) the detector cannot recognize the watermark in the image from another one, and (c) the watermark is no longer in the image.

The attacks can be very different: (a) in the spatial domain, it can be scaling, cropping, rotation, noise addition, STIRMARK, an open source software available on the Web, (b) in the frequential domain it can be filtering, (c) compression and (d) adding another watermark over the first one. The STIRMARK software generates random rotations and distortions on blocks of the image. The STIRMARK software simulates JPEG coding, filtering operations, rotation, scaling and cropping. The result is very slight alterations on image, but watermarks are usually heavily damaged.

While spatial domain watermarking, in general, is easy to implement on computational point of

view but too fragile to withstand large varieties of external attacks. On the other hand, frequency or transformed domain approach offers robust watermarking but in most cases implementation need higher computational complexity. Moreover the transform domain technique is global in nature and cannot restrict visual degradation of the cover image. But in the spatial domain scheme, degradation in image quality due to watermarking could be controlled locally leaving the region of interest unaffected.

The present paper describes a computationally efficient block based spatial domain watermarking technique for a one level watermark symbol. The selection of the required pixels is based on variance of the block and watermark insertion exploits average color of the blocks. The proposed algorithms were tested on a few of the most usual transformations of images and the obtained results showed that the proposed method is efficient.

2. WATERMARKING ALGORITHMS

All watermarking methods share the same building block – an embedding system and the watermark extraction or recovery system [3]. Any generic embedding system should have as inputs: a cover data/image (I), a watermark symbol (W) and a key (k) to enforce security. The output of the embedding process is always the watermarked data/image (I').

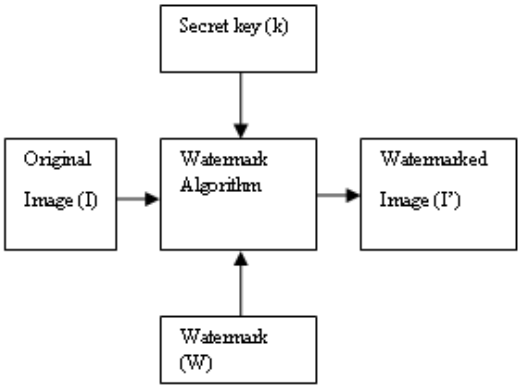

Fig. 1: Block diagram of a watermarking algorithm

The generic watermark recovery process needs the watermarked data, the secret key and depending on the method, the original data and/or the original watermark as input while the output is recovered watermark W with some kind of confidence measure for the given watermark symbol or an indication about the presence of watermark in the cover image under inspection.

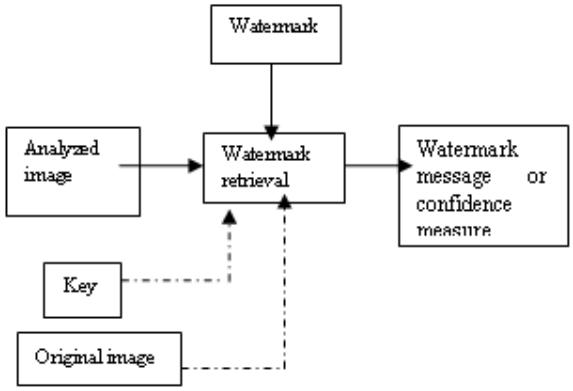

Fig. 2: Generic watermark recovery scheme

The original cover image I is a standard image of size $N \times N$ where $N = 2^p$ with a 24 bit RGB format. In the proposed work a binary image of size 256×256 or 512×512 is considered. It is marked each image with a watermark coefficient. That means, for each pixel, it is changed the value of the pixel given the following formula:

$$D(i,j) = C(i,j) + a*M*W$$

where C(i,j) is the original value of a pixel at position (i,j); D(i,j) is the watermarked value of the same pixel; *a* is a scalar factor (here *a* is chosen constant, but can be a variable of the position to improve the invisibility of the watermark and its detection); M is the mean of the block; and W is the watermark coefficient to be embedded. In our work, W could take the values +1 or –1 (one can easily extends the implementation to M).

The pixels of image are arranged into hexagons. Then the image is viewed as a graph not as a pixel matrix. The vertices represent the pixels and the edges represents neighborhood between pixels.

As it is said, the image is arranged into hexagons having different dimensions (edges). For introducing the key W, there will be considered certain nodes from the graph (i, j), which may be selected to represent a letter, a number or a function. In these nodes there are changed the three color's channels of the considered pixel depending on the three color channels of the bottom pixel minus one or another constant (const). We consider a started node having the coordinates (m0, n0) where it is started the marking process and an ended node where the process of marking is ended.

The algorithm for this operation is as following:

```
procedure construct_graph (Image I, edge ) is :
for * i->0,width/edge – 3*edge
    for * j->0;height/3
        if (i modulo 3==0)
         if (j modulo2 ==0)
          bmap[i][j]=bmp.bmap[edge*i][edge*j+edge-1];
         if (jmodulo2 ==1)
          bmap[i][j]=bmp.bmap[edge*i][edge*j+edge+2];
                #
        if (i modulo 3==1)
         if (j modulo2 ==0)
          bmap[i][j]=bmp.bmap[edge*i-1][edge*j-edge];
         if (jmodulo2 ==1)
          bmap[i][j]=bmp.bmap[edge*i-1][edge*j+edge*2];
              #
        if (i modulo3==2)
         if (j modulo2 ==0)
          bmap[i][j]=bmp.bmap[edge*i-2][edge*j+edge-1];
         if (j modulo2 ==1)
          bmap[i][j]=bmp.bmap[edge*i-2][edge*j+edge+2];
                #
        #
    *output the graph.
```

The algorithm for marking the image graph is shown bellow:

```
procedure mark_graph (Graph bmap) is :
*choose 2 nodes in the graph (m0,n0) and
(m1,n1)
  for * i->m0,n0
    for * j ->m1,n1
          change_color(i, j, const)
              #
          #
```

For reconstructing the marked image, there will verify only the graph's nodes corresponding to the selected key. If the marked pixel has the color in the interval [min, max] with respect to the color of the bottom pixel, then it will consider that this pixel was marked in conformity with the given algorithm. The values min, max resulted from a lot of experiments.

3. EXPERIMENTAL RESULTS

There are a lot of transformation that can be done on images [6], [7]: rotation, redimension, compression (transforming the image to JPEG), cropping and of course the case in which the image is not changed. Because it is not known what transformation the user done, all these transformations are verified one –by – one and the percent of similitude between the original image and the verified one is returned.

If the image is not transformed, it is applied the algorithm presented below:

```
procedure detect_untransformed (Image I,int edge,int
w,int h, int  percent) is :
 *construct_graph (I,edge );
      count=0;
 for * i->m0,n0
    for * j->m1,n1
     if (color(bmap[i][j]) =  = color(bmap[i+1][j])  -1)
        count = count +1;
              #
          #
```

*output percent of similitude between the original marked image and the image verified.

Also this algorithm may be applied if the image is cropped. In this case it may be possible to lose some marked pixels depends on the position where the image was cropped.

In Fig. 3, the first image is the image marked and the second one is the image marked and cropped. The detection algorithm detects this cropped image in percent of 88.88%.

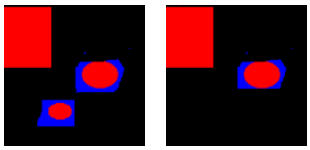

Fig. 3: Image marked and image marked and cropped

In the case of image rotation (angle of 90, 180 and arbitrary), there are verified all the image nodes because the nodes' position is changed. We search the nodes for which all of the three color' channels have the values like the color's channels of the bottom pixel minus one (or a certain constant). Before verifying these nodes, the image is dimensioned again to the initial dimensions at which the image is marked. In Fig. 4, the first image is the marked image and the second is the image marked and rotated by 30 degree.

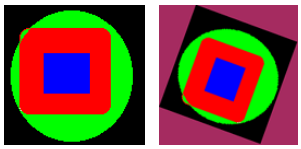

Fig. 4: Image marked and image marked and rotated

procedure detect_rotated (Image I,int edge,int w,int h, int percent,int constant) is:

***redimension(I,w,h);**

***construct_graph** (I,edge);

count=0;

```
for * i->0,w
   for * j->0,h
          if (color(bmap[i+1][j]) - constant <=
          color(bmap[i][j]) <= color(bmap[i+1][j])
          +constant)
          count=count+1;
          #
   #
```

*output percent of similitude between the original marked image and the image verified;

By experiments, there resulted that in the case of rotation by 90 and 180 degree, the constant is zero, but in the case of rotation by arbitrary angle the constant is very great (100).

The marked image was rotated with different angles. From experiments resulted the following percents of similitude between the marked image and the marked rotated image, as in the table.

Rotation Angle	Similitude Percent
30	33.33%
60	33.33%
90	100%
180	100%

The implemented algorithm entirely detects an image that was transformed to JPEG, only if the image is compressed with a quality of 100% and 80%. For image processes (compression, decompression), it was used ADOBE PHOTOSHOP. The variable m0, n0, m1, n1, h, w are known, being the same variables that are used in the process of image marking. The color of pixels arranged into nodes selected by us for marking the image is changed because of transformations supported by the image. Then the color of these pixels is searched into a certain interval.

procedure detect_jpg (Image I, int percent, int constant):

***decompressed(I);**

***construct_graph (I,edge);**

count=0;

```
for * i->m0,n0
   for * j->m1,n1
          if (color(bmap[i+1][j]) -constant
          <=color(bmap[i][j]) <= color(bmap[i+1][j])
          +constant)
          count = count +1;
          #
   #
```

*output percent of similitude between the original marked image and the image verified;

From experiments results that a good value for this constant is 30. Using different degree of image compression for JPEG, there are resulted the following percents of similitude between the marked image and the JPEG marked image.

Quality	Similitude Percent
100	100%
80	100%
60	33.33%
50	33.33%
30	0%
10	0%

In the case when we want to detect an image that was enlarged the results are weaker:

procedure detect_larged (Image I, int edge,int w,int h, int percent,int constant) is :

***redimension(I,w,h);**

***construct_graph (I,edge);**

count=0;

```
for * i->0,m
   for * j->0,n
          if (color(bmap[i+1][j]) -
          constant<=color(bmap[i][j]) <=
          color(bmap[i+1][j]) +constant)
          count = count +1;
          #
   #
```

*output percent of similitude between the original marked image and the image verified;

From experiments results that a good value for this constant is 70.

4. CONCLUSION

Watermarking, as opposed to stenography, has the additional notion of robustness against attacks. Even if the existence of the hidden information is known it is difficult for an attacker to destroy the embedded watermark, even if the algorithmic principle of watermarking method is public. In cryptography, this is known as Kerkhoffs law: - a cryptosystem should be secure, even if an attacker knows the cryptographic principles and methods used but does not have the appropriate key [8].

Meanwhile, the number of image watermarking publications is too large to give a complete survey over all proposed techniques.

Software watermarking is the process of embedding a large number into a program so that: (a) the number can be reliably retrieved after the program has been subjected to program transformations, (b) the embedding is imperceptible to an adversary and (c) the embedding does not degrade the performance of the program.

The method developed above satisfies the necessary requests for the watermarking technique and the series of presented transformations accounts for the fact that it resists possible attacks. The method is easy to implement and the experimentally determined robustness shows that it can be used without fear of being detected or changed

REFERENCES

[1] Hsu C.T., Wu Ja-L., Image Watermarking by Wavelet Decomposition, Academy of Information and Management Sciences Journal, Vol. 3, No. 1, pp. 70-86, 2000.

[2] Katzenbesser S., Petitcolas F.A.P., Information Hidden Techniques for Steganography and Digital Watermarking, Artech House, Boston, MA, 2000.

[3] Hsu C.T. , Wu Ja-L., Hidden Digital Watermarks in Images, IEEE Transaction on Image Processing, No. 8, pp 58-68, 1999.

[4] Cox I. J., Kilian J., Leighton T., Shammon T., Secure Spread Spectrum Watermarking for Multimedia, IEEE Transaction on Image Processing, No. 6, pp 1673-1687, 1997.

[5] Yeung M., Mintzer F., An Invisible Watermarking Technique for Image Verification, IEEE International Conf. on Image Processing, Santa Barbara, oct. 1997.

[6] Lin C., Wu M., Lui Y. M., Bloom J. A., Miller M. L., Cox I. J., Rotation, Scale and Translation Resilient Public Watermarking for Images, IEEE Transaction on Image Processing, Vol. 10, No. 5, pp 767- 782, 2001.

[7] Fei C., Kundur D., Kwong R., Analysis and Design of Watermarking Algorithms for Improved Resistance to Compression, IEEE Transaction on Image Processing, Vol. 13, No. 2, pp 126 – 144, 2004.

[8] Hartung F., Kutter M., Multimedia Watermarking Techniques, Proceedings of the IEEE, vol. 87. N0. 7, 1999.