# ROBUST SPATIAL IMAGE WATERMARKING USING PROGRESSIVE DETECTION

*Anastasios Tefas and Ioannis Pitas*

Department of Informatics, Aristotle University of Thessaloniki
Box 451, Thessaloniki 540 06, GREECE, pitas@zeus.csd.auth.gr

## ABSTRACT

A novel method for image watermarking robust to geometric distortions is proposed. A binary watermark is embedded in a grayscale or a color host image. The ability of progressive watermark detection enables fast and robust watermark detection even after several geometric distortions of the watermarked image. Simulation results indicate the ability of the proposed method to deal with the aforementioned attacks. Experiments conducted using the StirMark Benchmarking tests, indicate the superiority of the proposed method.

## 1. INTRODUCTION

In the last decades massive digitization of multimedia data such as photographs, paintings, speech, music, video, documents etc., became very popular. New techniques for the representation, storage and distribution of digital multimedia information have been developed. At the same time, the amount of digital data that is distributed through international communication networks has increased rapidly . In such an environment original digital products can be easily copied, tampered and transmitted back in the network. Consequently, the design of robust techniques for copyright protection of multimedia data became necessary.

The main challenge in multimedia watermarking for copyright protection is the robustness of the watermarking techniques against several types of attacks. The attacks that are usually encountered in image watermarking methods are image filtering, lossy image compression and geometric distortions. Although for the first two types of attacks many robust methods were proposed [1, 2], the robustness of the watermarking algorithms to geometric distortions is an unsolved problem yet.

In this paper a novel technique for image watermarking robust to geometric distortions is proposed. It is based on an established watermarking technique [3, 4]. A watermark signal is embedded in the spatial domain according to appropriate embedding functions. In the watermark detection the watermark signal is generated and its existence is examined according to the corresponding detection functions. The original image is not necessary during the watermark detection. The novelty of the method is based on the design of a robust watermark detector which can be implemented progressively, increasing significantly the watermark detection speed. Watermark detection after several geometric distortions of the watermarked image is performed in real time. The proposed watermarking technique is also robust against image filtering and lossy image compression.

## 2. WATERMARK GENERATION AND EMBEDDING

The watermark generation procedure aims at generating a three-valued watermark $w(\mathbf{x}) \in \{0, 1, 2\}$, from an image $f(\mathbf{x})$, given a digital key $K$. The watermark is a random sequence of three-valued data, thus, it is usually produced by a pseudorandom number generator. An alternative to random number generators is to use chaotic mixing systems or sequences that are produced by chaotic maps having prespecified spectral properties [5, 6].

After the watermark generation we proceed to the watermark embedding by altering the pixels of the original (host) image according to the following formula:

$$f_w(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } w(\mathbf{x}) = 0 \\ g_1(f(\mathbf{x}), \mathcal{N}(\mathbf{x})) & \text{if } w(\mathbf{x}) = 1 \\ g_2(f(\mathbf{x}), \mathcal{N}(\mathbf{x})) & \text{if } w(\mathbf{x}) = 2 \end{cases} \quad (1)$$

where $g_1, g_2$ are suitably designed functions based on $\mathbf{x}$ and $\mathcal{N}(\mathbf{x})$ denotes a function that depends on the neighborhood of $\mathbf{x}$. The functions $g_1, g_2$ are called *embedding functions* and they are selected so as to define an inverse detection function $G(f_w(\mathbf{x}), \mathcal{N}(\mathbf{x}))$. The detection function, when applied to the watermarked image $f_w(\mathbf{x})$, gives the watermark $w(\mathbf{x})$:

$$G(f_w(\mathbf{x}), \mathcal{N}(\mathbf{x})) = w(\mathbf{x}) \quad (2)$$

Obviously several embedding functions and the appropriate detection function can be designed giving different watermarking schemes. The function that is used in our method is based on a superposition of real quantities in the pixels which are going to be signed:

$$g_1(f(\mathbf{x}), \mathcal{N}(\mathbf{x})) = \mathcal{N}(\mathbf{x}) \oplus \alpha_1 f(\mathbf{x}) \quad (3)$$

$$g_2(f(\mathbf{x}), \mathcal{N}(\mathbf{x})) = \mathcal{N}(\mathbf{x}) \oplus \alpha_2 f(\mathbf{x}) \quad (4)$$

where $\alpha_1, \alpha_2$ are suitably chosen constants and $\mathcal{N}(\mathbf{x})$ is local neighborhood operation of the pixels around $\mathbf{x}$. The sign of $\alpha_1, \alpha_2$ is used for the detection function and its value determines the watermark power.

The size of the region around $\mathbf{x}$ used for the calculation of $\mathcal{N}(\mathbf{x})$ is important for the watermarking procedure. Moreover, the number of pixels used for the calculation of $\mathcal{N}(\mathbf{x})$ determines the upper bound of the number of watermarked pixels in an image. If a pixel to be signed is contained in the neighboring region of another signed pixel, the related watermark detection may be affected by the neighboring pixel alterations, thus resulting in a false detection. To avoid such problems we should use small watermark embedding neighborhoods (i.e., of size 3 × 3). The maximum number of pixels that can be signed in a host image of dimensions $N \times N$ by using blocks of $(2r + 1) \times (2r + 1)$ for calculating $\mathcal{N}(\mathbf{x})$ is given by:

$$k = \frac{N^2}{(r+1)^2} \quad (5)$$

This work has been supported by the European Project IST-1999-10987 CERTIMARK

## 3. WATERMARK DETECTION

In the detection procedure we generate first the watermark $w(\mathbf{x})$ according to the watermark key $K$. The detection function resulting from (3,4) is defined by:

$$G(f_w(\mathbf{x}), \mathcal{N}(\mathbf{x})) = \begin{cases} 1 & \text{if } f_w(\mathbf{x}) - \mathcal{N}(\mathbf{x}) > 0 \\ 2 & \text{if } f_w(\mathbf{x}) - \mathcal{N}(\mathbf{x}) < 0 \end{cases} \quad (6)$$

The detection function is valid if $\alpha_1 > 0$ and $\alpha_2 < 0$. This fact should be accounted for the design of the embedding functions. By employing the detection function in the watermarked image a bi-valued detection image $d(\mathbf{x})$ is produced:

$$d(\mathbf{x}) = G(f_w(\mathbf{x}), \mathcal{N}(\mathbf{x})) \quad (7)$$

Based on the watermark $w(\mathbf{x})$ and the detection image $d(\mathbf{x})$, we can decide whether the watermark under investigation is embedded in the imge or not. The detection is based on the pixel to pixel comparison for the nonzero pixels in $w(\mathbf{x})$. By comparing the watermark $w(\mathbf{x})$ and the detection image $d(\mathbf{x})$ we form the false detection image:

$$e_w(\mathbf{x}) = \begin{cases} 1 & \text{if } w(\mathbf{x}) \neq 0 \text{ and } w(\mathbf{x}) \neq d(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

The false detection image has value 1 (white pixels) if a watermarked pixel is falsely detected and 0 otherwise. The detection ratio is given by the ratio of the correctly detected pixels to the sum of the watermarked pixels in the image.

$$D_w = 1 - \frac{\text{card}\{e_w(\mathbf{x})\}}{\text{card}\{w(\mathbf{x})\}} \quad (9)$$

The embedding functions are designed in such way, so as the probability $p$ of a pixel to be detected as signed with $g_1$ or $g_2$, for an unwatermarked image, to be $0.5$. Thus, the detection ratio in an unwatermaked image forms a binomial distribution. The cumulative distribution function (*cdf*) of the watermark detection ratio is given by:

$$P_n = p^k \sum_{i=0}^{n} \frac{k!}{i!(k-i)!} \quad (10)$$

where $k$ is the total number of the watermarked pixels and $n$ is the number of correctly detected watermarked pixels.

The decision about the image ownership is taken by comparing the watermark detection ratio of the image to a predefined threshold $T$. The value of the threshold determines the minimum acceptable level of watermark detection.

## 4. PROGRESSIVE WATERMARK DETECTION AND ROBUSTNESS TO GEOMETRIC DISTORTIONS

The proposed watermarking algorithm is adequate for progressive watermark detection. By the term progressive watermark detection we mean that under certain conditions the watermark detection procedure is not necessary to be performed to the entire image. That is, the watermark detection in a small region of the image is sufficient for deciding whether the image is watermarked or not. The minimum size of the search region depends on the minimum number of watermarked pixels needed for the detection procedure. In order to estimate the minimum number of watermarked pixels needed for the detection procedure we can condition the minimum acceptable false watermark acceptance probability. Let us denote by $\epsilon_1$ the upper bound of false watermark acceptance probability ($\epsilon_1$ is usually set equal to $10^{-4}$):

$$Pr\{D_{w_i} > T\} < \epsilon_1, \quad \forall w_i \neq w \quad (11)$$

where $T$ is the watermark detection threshold and $w$ is the correct watermark.

The watermark detection ratio, for a watermark other than the one embedded in the image or for an unwatermarked image, follows (10). Thus, equation (11) can be rewritten as:

$$\frac{1}{2^m} \sum_{i=Tm}^{m} \frac{m!}{i!(m-i)!} < \epsilon_1 \quad (12)$$

By using DeMoivre-Laplace theorem [7] it follows that:

$$\mathbf{G}\left\{\frac{Tm - 0.5m}{0.5\sqrt{m}}\right\} > 1 - \epsilon_1 \quad (13)$$

and the minimum number $m$ of watermarked pixels needed for the watermark detection is:

$$m > \left(\frac{\text{erfinv}(0.5 - \epsilon_1)}{2T - 1}\right)^2 \quad (14)$$

where erfinv is the inverse function of the error function:

$$\text{erf}\, x = \frac{1}{\sqrt{2\pi}} \int_0^x e^{-y^2/2} dy \quad (15)$$

Having calculate the minimum number of watermarked pixels needed for watermark detection, the minimum search region for the watermark detection can be evaluated. The minimum number of watermarked pixels needed for watermark detection and the size of the corresponding search regions for several watermark detection thresholds $T$ and $\epsilon_1$ equal to $10^{-4}$ are plotted in Figure 1. It is obvious from the plots that a region of approximately $59 \times 59$ is sufficient for watermark detection for a watermark threshold greater than $0.65$. Simulation results for 1000 watermarks at each threshold agree with the theoretical analysis as it can be observed in Figure 1b. The false watermark detection probability is always approximately $\epsilon_1 = 10^{-4}$.

According to the previous analysis we can start the watermark detection from regions smaller than the lower limit provided that the corresponding watermark detection threshold is increased such that inequality (14) is always satisfied. As the search region is increased the watermark detection threshold is decreased. The size of the smaller region that we can start from, is given by the condition that the probability of rejecting the correct watermark, by performing watermark detection only in this region, should be lower than $\epsilon_2$:

$$Pr\{D_{w_i} < T\} < \epsilon_2, \quad w_i = w \quad (16)$$

where $T$ is the watermark detection threshold, $w$ is the correct watermark and $\epsilon_2$ is a very small number (approximately $10^{-4}$). The progressive watermark detection procedure is illustrated in Figure 2. Watermark detection starts from a small region and if the watermark detection ratio in this area is larger than the corresponding threshold the watermark detection proceeds to the next contour. By comparing at each step the watermark detection ratio to the corresponding threshold the algorithm stops when the search region becomes greater than the lower limit given by (14), deciding that the watermark under investigation is indeed embedded in the

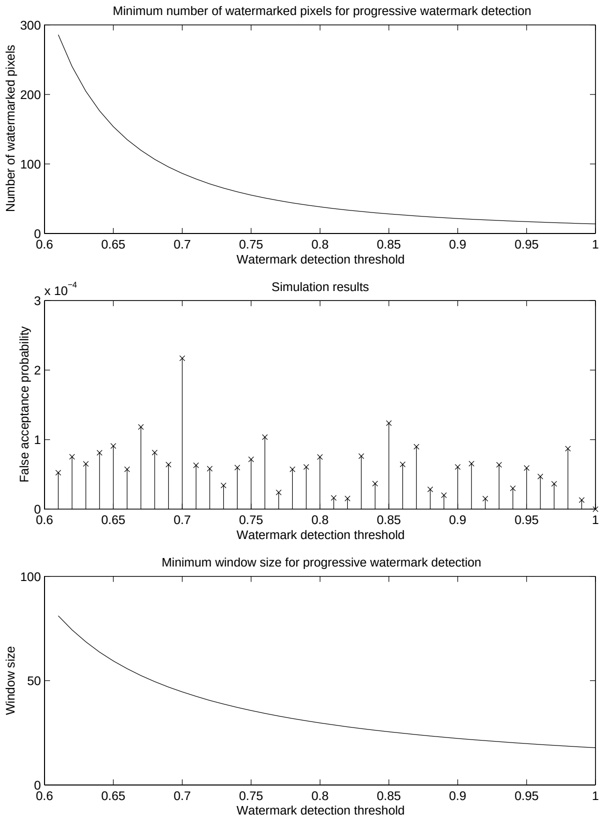


**Fig. 1**. Minimum number of watermarked pixels needed for progressive detection. False watermark acceptance probability achieved by simulation results for several thresholds and the corresponding theoretic number of watermarked pixels. The corresponding search region size for several thresholds.

image or when the watermark detection ratio falls under the corresponding threshold, deciding that the image is not watermarked.

The proposed algorithm allows very fast watermark detection thus enabling correct watermark detection for several geometric distortions of the watermarked image. Specifically, the detection of a watermark shifted to every pixel of an image is performed in real time. Moreover, robustness against image cropping is achieved by exhaustive search of the watermark in the test image which is performed in real time due to progressive watermark detection. Accordingly, the watermark detection for several geometric distortions, like scaling and rotation, of the watermarked image is implemented in a few seconds.

The speed of the proposed algorithm is further improved by constructing watermark signals of specific structures. That is, the watermark is embedded more than one times in the host image. If the host image is larger than the watermark then the watermark is embedded at several non-overlapping regions of the host image until the entire image is covered. The copies of the watermark that are embedded in the image are rotated versions of the original watermark. Thus, searching for rotated versions of the watermark in the detection procedure can be applied in a reduced search space, enabling faster detection.

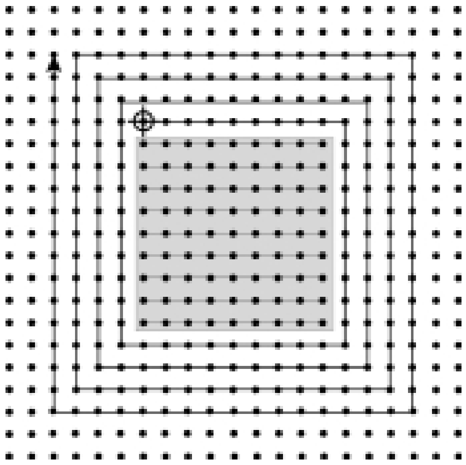


**Fig. 2**. Progressive watermark detection.

## 5. EXPERIMENTAL RESULTS

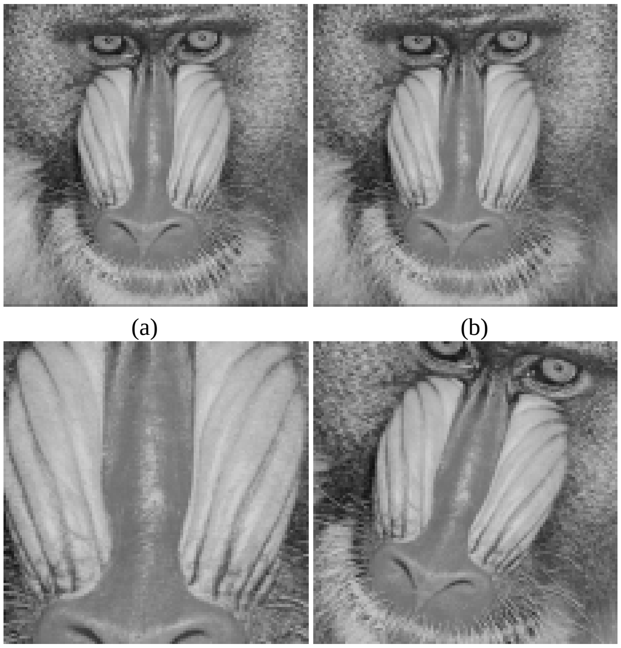
The proposed approach was evaluated using a standard series of tests detailed in the Stirmark program [9, 10]. The tests are divided into the following sections: signal enhancement, compression, scaling, cropping, shearing, rotation, row/column removal and random geometric distortions. The images used for the experiments are: Lena, Baboon, FishingBoat, Watch, Skyline_Arch and Bear. The watermark embedding strength was set so that the PSNR of the watermarked images to be $\approx$ 38dB. Each attack is considered either combined with lossy image compression (JPEG with quality 90%) or alone.

Each image is being watermarked and then a series of attacks are applied to the watermarked images. The algorithm is trying to detect the watermark in each attacked image and if the watermark is correctly detected a score 1 is assigned otherwise a zero value is assigned. The original image Baboon and the watermarked version of it are illustrated in Figures 3a, 3b. Two examples of attacked images resulted using Stirmark are shown in Figures 3c and 3d. The first image is the watermarked image after cropping of 50% and the second image is a rotated, cropped and scaled version of the watermarked image. The watermark embedding algorithm takes less than a second to be applied while the average watermark detection time for all tests was approximately 50 sec. on a Pentium 500 MHz. This is sufficiently fast for commercial applications. We note here that if the algorithm does not search for geometric distortions of the test image then the watermark detection is performed in real time.

The results are summarized in Table 1. The number assigned at each attack is the average correct watermark detection of all experiments conducted for this attack. In the case of image enhancement the watermark is always correctly detected. The attacks in the image enhancement section was Gaussian, median and sharpening filters as well as the Frequency mode Laplacian Removal attack. The watermark detection method is successful against JPEG compression down to a level of 10 quality factor. Against scaling the proposed watermarking method is also quite robust since the detection algorithm fails only when the watermarked image is decimated using a scaling factor of 0.5. In that case the information lost due to image interpolation reduces the watermark detection ratio causing a missed detection. In the case of image cropping

**Table 1**. Evaluation of algorithm performance using Stirmark

| Attack | Proposed Method | Robust Template Matching [8] | Digimarc | Suresign | Unige |
|---|---|---|---|---|---|
| Enhancement | 1 | 1 | 1 | 1 | 0.92 |
| Compression | 1 | 0.74 | 0.65 | 0.87 | 0.63 |
| Scaling | 0.89 | 0.78 | 0.76 | 0.9 | 0.85 |
| Cropping | 0.9 | 0.89 | 0.99 | 0.93 | 0.83 |
| Shearing | 0.69 | 1 | 0.5 | 0.42 | 0.29 |
| Rotation | 0.88 | 1 | 0.96 | 0.44 | 0.98 |
| Row/Column removal | 1 | 1 | 1 | 0.89 | 0.83 |
| Random geometric | 1 | 0 | 0.17 | 0 | 0 |
| Average | 0.92 | 0.8 | 0.75 | 0.68 | 0.66 |



(a) (b)

(c) (d)

**Fig. 3**. (a) Original image. (b) Watermarked image (PSNR≈ 38dB). (c) Cropped image (Cropping 50%). (d) Rotated, cropped and scaled image (Rotation 15 degrees).

and rotation the watermark detection algorithm exhibits very good performance. In the case of image cropping watermark detection fails when 75% of the watermarked image is cropped and thus the remaining image regions contain only a small part of the watermark. The performance of the proposed algorithm can be further improved if the search regions of the watermark will be increased. In that case, watermark detection succeeds almost 100%, but the time needed for the detection procedure is also increased. It is worth noting that the proposed watermarking method can deal with the random geometric distortions applied using Stirmark, since in all cases the watermark is correctly detected.

The performance of other commercial products and watermarking methods proposed in the literature against the Stirmark benchmarking tests is drawn in Table 1. It can be observed that the proposed method attains the best average performance. More specifically, to our knowledge, the robustness of the proposed method against lossy image compression and the random geometric distortions of Stirmark can not be found in any other method.

## 6. CONCLUSIONS

An image watermarking method has been presented. The minimum number of watermarked pixels needed for reliable watermark detection has been theoretically derived and embedding of smaller watermarks has been enabled. A progressive watermark detection technique has also been proposed that enables fast and robust watermark detection after several geometric distortions of the watermarked image. The major advantage of the proposed method is its robustness against geometric distortions.

## 7. REFERENCES

[1] M.D. Swanson, M. Kobayashi, and A.H. Tewfik, "Multimedia data embedding and watermarking technologies," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064–1087, June 1998.

[2] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1079–1107, July 1999.

[3] G. Voyatzis and I. Pitas, "Digital image watermarking using mixing systems," *Computer & Graphics*, vol. 22, no. 3, 1998.

[4] G. Voyatzis and I. Pitas, "The use of watermarks in the protection of digital multimedia products," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1197–1207, July 1999.

[5] J. Fridrich, A.C. Baldoza, and R.J. Simard, "Symmetric ciphers based on 2d maps," in *Proc. IEEE Conf. on Systems, Man, and Cybernetics*, Octomber 1997, pp. 1105–1110.

[6] N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, A. Tefas, V. Solachidis, and I. Pitas, "Applications of chaotic signal processing techniques to multimedia watermarking," in *Proceedings of the IEEE workshop on Nonlinear Dynamics in Electronic Systems*, Catania Italy, May 18-20 2000, pp. 1–7.

[7] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, New York, 1991.

[8] S. Pereira and T. Pun, "Robust template matching for affine resistant image watermarks," *IEEE Trans. on Image Processing*, vol. 9, no. 6, pp. 1123–1129, June 2000.

[9] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Attacks on copyright marking systems," in *Second International Workshop on Information Hiding*, Portland, USA, April 15-17 1998, pp. 219–239.

[10] F.A.P. Petitcolas and R.J. Anderson, "Evaluation of copyright marking systems," in *International Conference on Multimedia Systems*, Florence, Italy, June 7-11 1999.