Computer Modelling & New Technologies, 2004, Volume 8, No.1, 39-47
Transport and Telecommunication Institute, Lomonosov Str.1, Riga, LV-1019, Latvia

A FAST AND ROBUST WATERMARKING METHOD FOR JPEG IMAGES

W. LUO[1], G.L. HEILEMAN[2]

[1]Engineering Department, St. Mary's University, San Antonio, TX 78228, U.S.A.
E-mail: wluo@stmarytx.edu
[2]Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, U.S.A.

In this paper, a JPEG domain image watermarking method that utilizes spatial masking is presented. The watermarking algorithm works in the compressed domain, and can be implemented efficiently in real-time (only 50ms is required for a 512x512 24-bit color image on a 700MHz computer). In many applications, particularly those associated with delivering images over the Internet, the ability to watermark images in real-time is required. In order to achieve a real-time watermarking capability, the proposed technique avoids many of the computation steps associated with JPEG compression. Specifically, the forward and inverse DCT do not need to be calculated, nor do any of the computations associated with quantization. Robustness to JPEG compression, different kinds of noise (additive, salt & pepper, and speckle) and image cropping attacks are achieved with the proposed system, and the relationship between watermark robustness and watermark position is described. A further advantage of the proposed method is that it allows a watermark to be detected in an image without referencing to the original unwatermarked image, or to any other information used in the watermark embedding process.
Keywords: *image watermarking, spatial masking*

1. Introduction

The success of the Internet, cost-effective and popular digital recording and storage devices, and the promise of higher bandwidth and quality of service (QoS) for both wired and wireless networks has made it possible to create, replicate, transmit, and distribute digital content in an effortless way [1,7]. The need for developing watermarking techniques that protect electronic information has become increasingly important due to the widespread availability of methods for disseminating exact copies of this information (e.g., via the Internet), and the ease with which this information can be reproduced [2,6,8]. Digital watermarking is increasingly being used for the purposes of protecting digital content against unauthorized usage or theft, and for documenting or ensuring (i.e., verifying, guaranteeing, or proving) the integrity of multimedia content. Digital image watermarking involves the embedding of additional information into an image in a manner that is imperceptible to the human observer, but which can be discovered by watermark detection algorithms. Digital image watermarking is typically performed in either the frequency or spatial domain. Early digital image watermarking methods used the spatial domain to perform watermark embedding by simply changing the least significant bit of each pixel in order to encode a message. It has been found that transform domain watermarking schemes are typically much more robust to image manipulation as compared to spatial domain schemes. The method proposed here belongs to frequency domain watermarking category, and in particular involves modifications to the discrete cosine transform (DCT) domain coefficients.

In the DCT domain, watermarks should be embedded in those coefficients that meet the following requirements in order for the watermarks to be invisible and also robust against attacks aimed at removing them [3]. First, watermark embedding should target those coefficients having large perceptual capacity, allowing strong watermarks (strong against attacks) to be embedded without perceptual distortion. Second, the embedding should focus on those coefficients that will change little when common image processing and noise corruption attacks are applied. This should include both intentional and unintentional attack possibilities.

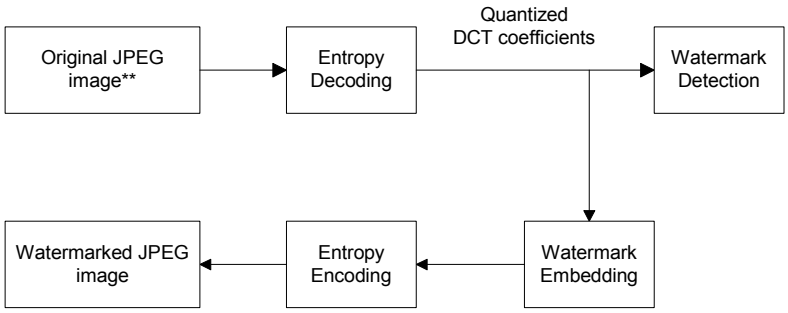


Figure 1. The proposed watermarking system

In many applications related to Internet-based delivery of images, particularly those associated with large image databases; there are requirements for real-time watermark insertion and extraction. The method described in this paper was developed with these requirements in mind. In particular, this method is capable of embedding a watermark in a 512x512 24-bit color JPEG image in approximately 50ms using a 700MHz computer. Watermark detection and extraction are similarly fast, and do not require access to the original (unwatermarked) image, or to any other information used in the watermark embedding process, in order to detect and extract the watermark. Thus, costly searches through an image database can be avoided during the watermark detection/extraction process.

2. Watermarking Method

JPEG compression method starts by dividing an image into disjoint 8x8 blocks of pixels. Next, for each block, the forward DCT is calculated, producing 64 DCT coefficients. Let us denote the (x, y)-th DCT coefficient of the k-th block as $d_k(x,y)$, $0 \le x, y \le 7$, $k = 1, \cdots, B$, where B is the total number of blocks in the image. In each block, all 64 coefficients are further quantized to integers $D_k(x,y)$ using a JPEG quantization matrix Q:

$$D_k(x,y) = R\left(\frac{d_k(x,y)}{Q(x,y)}\right), \tag{1}$$

where R denotes the integer round operation. The quantized coefficients are then arranged in a *zig-zag* manner denoted by $Z_k(i)$, $0 \le i \le 63, k = 1, \cdots, B$, and compressed using a Huffman coder. The resulting compressed stream, together with a header, forms the JPEG compressed image file. For robustness and simplicity reasons, the method described here embeds watermarks in the luminance (Y) component of an image, leaving the chromatic components (C_b and C_r) intact. The general model for the watermarking system is shown in Figure 1. Note that for the watermark detection/extraction process, we assume that the input to the system (i.e., the "original JPEG image") is the watermarked, and possibly attacked, image.

The specific computations performed at each step in the watermarking embedding process are as follows:

- Read the original JPEG image, and for each block k, perform entropy decoding in order to obtain the quantized DCT coefficients $Z_k(i)$, $0 \le i \le 63, k = 1, \cdots, B$
- Let i_0 denote the initial coefficient within the coefficient block (in *zig-zag* order) where watermark insertion begins. Then, in four adjacent DCT coefficient blocks, as shown in Figure 2, a single bit w is embedded as follows:

8x8 DCT block 1	8x8 DCT block 2
8x8 DCT block 3	8x8 DCT block 4

Figure 2. An embedding unit

if $w = 1$ **then**
if $Z_1(i_0) < \overline{M} + \delta$ **then** $Z_1^*(i_0) = \overline{M} + \delta$
else ($w = 0$)
if $Z_1(i_0) < \overline{M} - \delta$ **then** $Z_1^*(i_0) = \overline{M} - \delta$

where $\overline{M} = R((Z_2(i_0) + Z_3(i_0) + Z_4(i_0))/3)$, and δ is determined by the local characteristics of the image. Specifically, let $M_{\max}$ = max $\{ Z_1(i_0),\ Z_2(i_0),\ Z_3(i_0),\ Z_4(i_0) \}$, $M_{\min}$ = min $\{ Z_1(i_0),\ Z_2(i_0),\ Z_3(i_0),\ Z_4(i_0) \}$, and assume T_1 and T_2 are two adjustable threshold values with $T_1 < T_2$. Then, δ is computed as follows:

$\delta = 1$, if $(M_{\max} - M_{\min}) \le T_1$; $\delta = 2$, if $T_1 < (M_{\max} - M_{\min}) \le T_2$; $\delta = 3$, if $T_2 < (M_{\max} - M_{\min})$.

The algorithm uses the fact that the relationship between DCT coefficients at the same position in different 8x8 blocks of an image will hold even if these coefficients are quantized by an arbitrary quantization table in the JPEG compression process [5]. The algorithm also exploits the fact that $Z_1(i_0)$ is usually close to $\overline{M}$.
Since the visibility of the superimposed watermark signal is affected by the background texture [3], the stronger the texture in the background, the lower the visibility of the embedded signal will be (this is texture masking) [4]. The method therefore embeds a stronger watermark signal in stronger texture areas.

- Replace $Z_1(i_0)$ with the new coefficients $Z_1^*(i_0)$. The above procedure is applied to all four DCT coefficient adjacent blocks. The final watermarked image I^* is then obtained by entropy encoding these modified DCT blocks.

Watermark extraction is the inverse of the watermark embedding procedure. Suppose that I^* is the signal-distorted or maliciously-attacked watermarked image. To extract the watermark from I^*, $\overline{M}^*$ is calculated from $Z^*(i_0)$ in the same way as in step 2 above. Then, the watermark w is extracted according to following rule:

if $Z_1^*(i_0) > \overline{M}^*$ **then** $w = 1$ **else** ($Z_1^*(i_0) \le \overline{M}^*$) $w = 0$.

The experiments detailed in the section 4 were used to quantify the performance of this watermarking system.

3. Watermark Mixing

Visually meaningful pattern, such as letters and logos, can serve as a quick check for ownership. Shown in Figure 3 is a 53x53 binary pattern “EECE”.

Figure 3. Original watermark

The decision on whether an image is watermarked or not can be made by (automatically) comparing the extracted pattern with the original one, if available, or by human (e.g., jury in court) based on visualizing the extracted pattern. The latter case uses a reasonable assumption that human can distinguish a “meaningful” pattern from a random one. It is also possible to make such decision automatically, e.g., computing a randomness measure. Obviously, without appropriate adjustment for the spatial relationship of the watermark, a common image cropping operation may destroy the visual pattern of embedded watermark.

To survive image cropping, a two-dimensional “torus automorphism” is used to permute (or mix) the watermark to disperse its spatial relationship. A two-dimensional “torus automorphism” can be considered like a spatial transformation of planar regions which belong in a square two-dimensional area. The set of torus automorphisms is special subset of Anosov diffeomorphisms, which exhibit strongly chaotic motion i.e. local instability, ergodicity, mixing and decay of correlations [9]. A great subset of torus automorphisms is represented by the family of one-parameter systems which is defined as follows:

$$A_N(k): L \to L \quad \begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} (\mathrm{mod}\, N) \tag{2}$$

where $(x_n, y_n) \in L = [0, N-1] \times [0, N-1]$. For the $N-1$ integer values of k in the domain $[1, N)$ we obtain a finite family of systems $A_N(k)$. For any integer lattice L of size N there is an integer $P = P(k, N)$ such that

$$A_N^P(k)\xi = \xi \ (\mathrm{mod}\, N), \quad \forall \xi \in L. \tag{3}$$

We call the integer P "recurrence time". Thus any lattice point is a fixed point under the action of $A_N^P(k)$ and also the periodicity condition $A_N^{i+jP}(k)\xi = A_N^i(k)\xi$ holds, for all positive integers i, j and for all $\xi \in L$.

In Figure 4, the watermark is mixed using torus automorphism $A_{53}^5(1)$ to survive image cropping.

Figure 4. Mixing of binary pattern "EECE" by $A_{53}^5(1)$

4. Experimental Results

The similarity measurement ``Normalized Correlation" (*NC*) between the original watermark w and the extracted watermark $\overline{w}$ is defined as:

$$NC = \frac{\sum_i \sum_j w(i,j)\overline{w}(i,j)}{\sum_i \sum_j [w(i,j)]^2}.$$

This provides objective judgment of the extracting fidelity.

The experiments described here compare the performance of different embedding strategies in terms of robustness against JPEG compression, different kinds of noise, and image cropping attacks. The two standard color images shown in Figure 5, *lenna* and *baboon*, were used in our experiments. All watermarked images derived from these test images were 512x512 pixels and 24-bits in color.

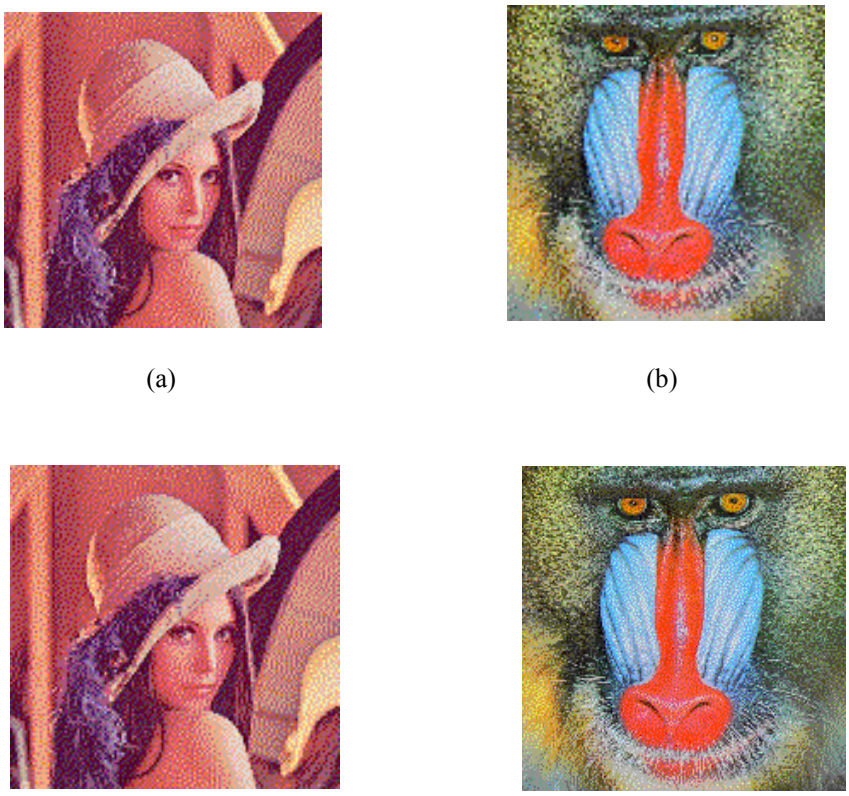
First the two test images were JPEG compressed by quality factor 75. Next, the same watermark was embedded into these two compressed images using parameters $T_1 = 15$, $T_2 = 30$, and at various coefficient locations determined by i_0, $0 \le i_0 \le 63$. The average watermarking time of the proposed system was approximately 50ms using a 700MHz computer. A single bit was embedded in a 2x2 block, and in total 1024 bits were embedding in an image. Figure 5 shows the original and the watermarked images, respectively, when $i_0 = 10$. The method does not cause perceptible changes to be introduced into the watermarked images.

The first experiment studies the effects of watermark embedding at various positions ($i_0 = 0, 2, 4, 6, 8, 10$, and 12) on the peak signal-to-noise ratio (PSNR), which is calculated as the difference between the original test image and the watermarked image.

From Figure 6, we see that images containing stronger texture features, e.g., *baboon* will yield a lower PSNR. This is exactly what we want to achieve with texture-based watermarking. That is, the goal is to always embed stronger watermark signals into rich texture areas.

The next set of experiments details watermark robustness to JPEG compression. The watermarked images, embedded with parameters $i_0 = 0, 2, 6$, and 10, were attacked by JPEG compression at different quality levels. The results are shown in Figures 7 (a), (b), and (c). From Figures 7 (a) and (b), it is obvious that watermarks embedded at lower frequencies are more robust than watermarks embedded at higher frequencies. Also, Figure 7 (c), with $i_0 = 10$, shows that the watermark

in *baboon* is more robust to JPEG compression attacks than the one in *lenna*. This is because the watermark in *baboon*, as described above, is embedded stronger due to the spatial masking used during watermark embedding. Another experiment studies watermark robustness to additive noise attacks. The watermarked images ($i_0 = 10$) are attacked by additive Gaussian noise at different energy levels. The results are presented in Figure 7 (d). It is easy to see from this figure that the watermarked images are very robust to additive noise attacks.



(a) (b)

(c) (d)

Figure 5. The original images (a) *lenna* and (b) *baboon* and the corresponding watermarked versions (c) and (d)

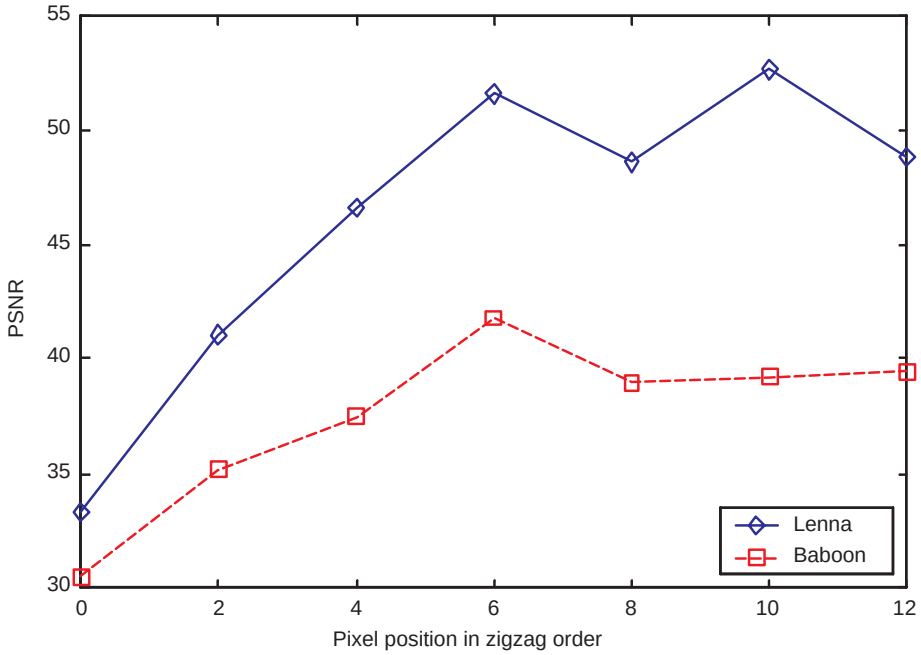


Figure 6. The effects of watermark embedding on PSNR

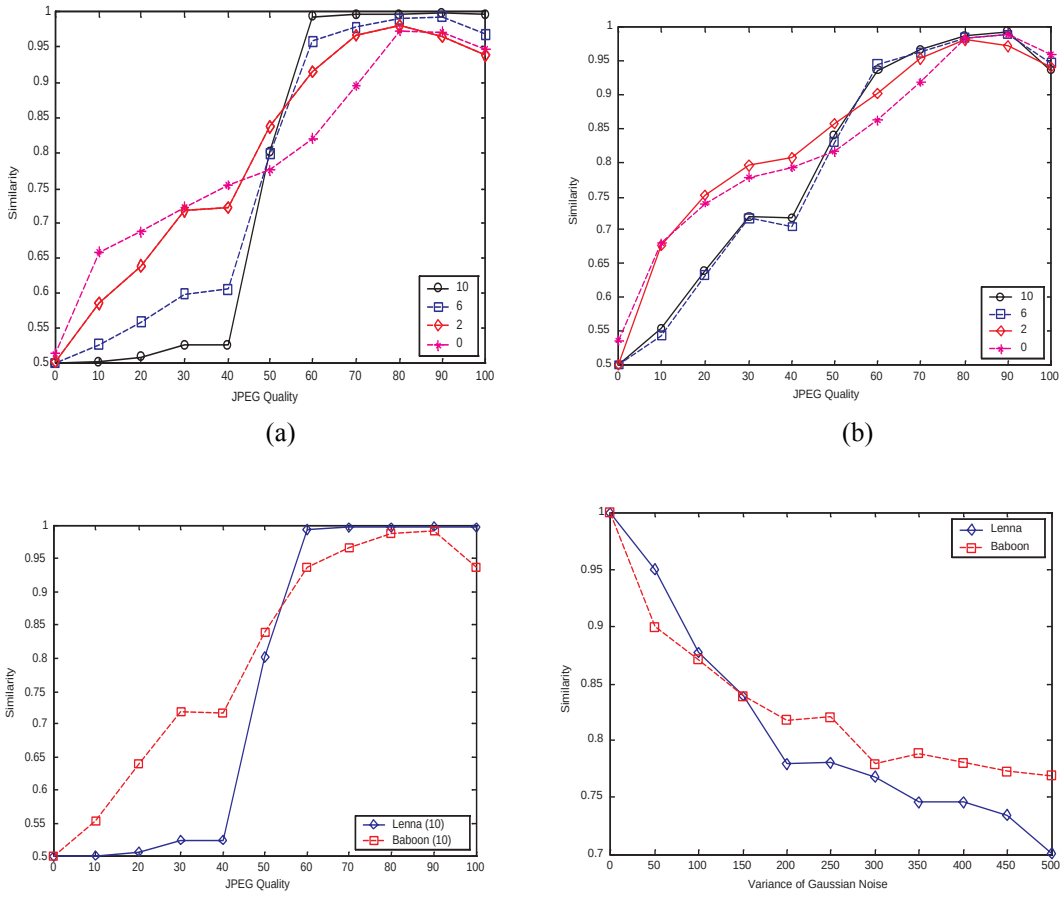


Figure 7. Comparison of watermark robustness to JPEG compression relative to watermark position in (a) *lenna* and (b) *baboon*; (c) comparison of watermark robustness to JPEG compression relative to specific images (with different texture components); (d) comparison of watermark robustness to additive noise attack for the watermarked *lenna* and *baboon* images

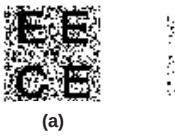


Figure 8. Recovered watermark from *lenna* after JPEG compression attack

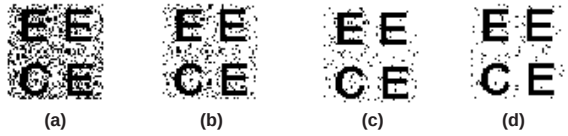


Figure 9. Recovered watermark from *baboon* after JPEG compression attack

TABLE 1. *NC*s of recovered watermarks after JPEG attack

Figure	**(a)**	**(b)**	**(c)**	**(d)**
Q	50	60	70	80
NC (lenna)	0.65	0.91	0.97	0.97
NC (baboon)	0.69	0.85	0.93	0.96

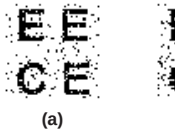


Figure 10. Recovered watermark from *lenna* after Gaussian additive noise attack

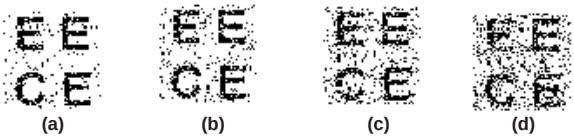


Figure 11. Recovered watermark from *baboon* after Gaussian additive noise attack

TABLE 2. *NC*s of recovered watermarks after Gaussian additive noise attack

Figure	**(a)**	**(b)**	**(c)**	**(d)**
Variance	20	50	100	200
NC (lenna)	0.95	0.88	0.81	0.77
NC (baboon)	0.95	0.91	0.87	0.81

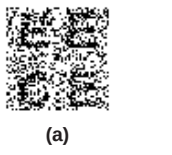


Figure 12. Recovered watermark from *lenna* after salt & pepper noise attack

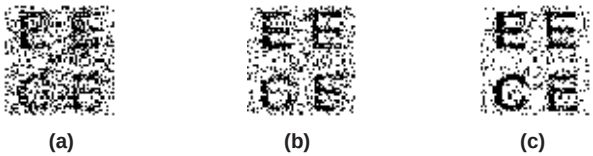


Figure 13. Recovered watermark from *lenna* after speckle noise attack

Figure 14. Recovered watermark from *lenna* after image cropping attack

TABLE 3. *NC*s of recovered watermarks after salt & pepper noise attack

Figure	**(a)**	**(b)**	**(c)**
Density	0.02	0.01	0.005
NC	0.72	0.81	0.86

TABLE 4. *NC*s of recovered watermarks after speckle noise attack

Figure	**(a)**	**(b)**	**(c)**
Variance	0.02	0.01	0.005
NC	0.72	0.79	0.85

TABLE 5. *NC*s of recovered watermarks after image cropping attack

Figure	**(a)**	**(b)**	**(c)**
Cropping ratio	25%	50%	Random cropping
NC	0.72	0.50	0.80

Finally, we conducted a series of experiments by embedding the visual pattern in Figure 4 into the original image to further test the watermarking effects and robustness of our watermarking scheme to JPEG compression, different kinds of noise (Gaussian, salt & pepper, and speckle), and image cropping attacks when it is used in some real applications.

- First, different JPEG compression qualities ($Q = 50, 60, 70, 80$) are applied to the watermarked image. The recovered visual patterns after JPEG attack are shown in Figures 8 and 9. The *NC*s of recovered watermarks and the corresponding JPEG compression qualities Q are listed in Table 1.
- Second, different kinds of noises with different variances and densities are applied to the watermarked image. The recovered visual patterns after different noise attacks are shown in Figures 10-13. The *NC*s of recovered watermarks and the corresponding variances and densities are listed in Tables 2-4.
- The last experiment studies watermark robustness to image cropping attacks. We cropped the watermarked image with different cropping ratios (25%, 50% and random cropping). The recovered visual patterns after image cropping attacks are shown in Figure 14. The *NC*s of recovered watermarks and the corresponding cropping ratios are listed in Table 5.

From the experiments above, we see that the embedded visual watermark was shown to be robust to JPEG compression, different kinds of noise and image cropping attacks.

5. Conclusions

In this paper, we have proposed a fast and robust JPEG domain image watermarking method. The proposed method can be implemented very efficiently, requiring approximately 50ms to embed or extract a watermark using a 700MHz computer. The embedded watermark was experimentally shown to be robust to JPEG compression, different kinds of noise (additive, salt & pepper, and speckle) and image cropping attacks, and can be extracted without reference to the original (unwatermarked) image and embedding parameters. Also, the experiments described here show that embedding watermarks in coefficients, which are "important" for the image, are more likely to retain embedded watermark data, despite attacks that result in visually unimportant distortions. Correct choices for the threshold values T_1 and T_2 are of fundamental importance for watermark invisibility and good detector/extractor performance. It is possible to adjust the threshold values according to different image features, and even add additional threshold level. So, our watermarking scheme is very flexible, and can be tailored to meet the requirements of different real applications.

6. Acknowledgements

The authors are grateful to Elisar Software Corporation for providing financial support for this research.

References

[1] Cox I.J., Bauml R., and Girod B. Digital Watermarking, *Academic Press*, 2002.
[2] Hartung F., and Kutter M. Multimedia Watermarking Techniques, *Proc. IEEE*, Vol. 87, No. 7, pp. 1079-1107, 1999.
[3] Huang J. and Shi Y.Q. and Shi Y. Embedding Image Watermarks in DC Components, *IEEE Trans. on circuits and systems for video technology*, Vol. 10, pp. 974-979, 2000.
[4] Langelaar G.C. and Lagendijk R.L. Optimal Differential Energy Watermarking of DCT Encoded Images and Video, *IEEE Trans. on image processing*, Vol. 10, pp. 148-158, 2001.
[5] Lin C.Y. and Chang S.F. A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation, *IEEE Trans. on circuits and systems for video technology*, Vol. 11, pp. 153-168, 2001.
[6] Petitcolas F.A.P., Anderson R.J., and Kuhn M.G. Information Hiding - A Survey, *Proc. IEEE*, Vol. 87, No. 7, pp. 1062-1078, 1999.
[7] Podilchuk C.I. and Delp E.J. Digital Watermarking Algorithms and Applications, *IEEE Signal Processing Magazine*, Vol. 7, pp. 33-46, 2001.
[8] Swanson M.D., Kobyashi M., and Tewfik A.H. Multimedia Data-Embedding and Watermarking Technologies, *Proc. IEEE*, Vol. 86, No. 6, pp. 1064-1087, 1998.
[9] Voyatzis G. and Pitas I. Applications of Toral Automorphisms in Image Watermarking, *Proc. of ICIP96*, Vol. 2, pp. 237-240, 1996.

Received on the 20th of April 2004