A Fast, Blind, Transparent, and Robust Image Watermarking Algorithm with Extended Torus Automorphism Permutation

Hanan Elazhary
Electronics Research Institute
Giza, Egypt

ABSTRACT

In this paper, we present a novel algorithm for watermarking a grayscale digital image with a binary watermark in the DCT domain. The algorithm is transparent since the watermark is not really embedded in the host image. This results in zero distortion of the watermarked host image. The embedded watermark is robust to most common unintentional attacks by intelligently utilizing the signs of the DC components of the DCT-transformed host image blocks. The algorithm is blind since only the secret keys are required for watermark extraction. These secret keys are in the form of two shares. One of the shares is registered to the Certified Authority (CA) for additional security and protection against intentional attacks. The size of each share is much smaller than that of other techniques in the literature and the shares are generated faster. The algorithm is practical due to its very fast speed of both watermark embedding and extraction. The paper also proposes an extended version of Torus Automorphism (TA) permutation for scrambling the watermark before embedding and to reassemble it after extraction for additional security against intentional attacks.

General Terms

Imaging, Multimedia Security, Copyright Protection, Intellectual Property, Ownership Verification.

Keywords

digital image watermarking, Torus Automorphism permutation.

1. INTRODUCTION

The Internet has become the most popular channel for transmitting various forms of multimedia digital data such as digital images and video. Multimedia data in digital format can be modified and illegally used with ease. Thus, the ownership or copyright protection of digital images transmitted over the Internet has become an important research topic in recent years. One possible technique is digital image watermarking. In this technique, the watermark image is embedded into the host image such that the embedded watermark can be later extracted to make an assertion about the host image ownership.

There are some essential requirements for this purpose. The first requirement is the invisibility or transparency of the embedded watermark. In other words, the embedded watermark should not be perceived by human eyes and should not degrade the quality of the watermarked host image. A second requirement is the robustness of the embedded watermark. In other words, the embedded watermark should be able to resist both intentional and unintentional attacks. In intentional attacks, the attackers try to extract the embedded watermark for subsequent destruction. In unintentional attacks, on the other hand, the watermarked host image is treated using image processing techniques including compression and filtering. In the literature, digital image watermarking is done in the spatial domain [1-3] or in a transform domain including Discrete Cosine Transform (DCT) [4-6], Discrete Wavelet Transform (DWT) [7, 8] or a combination of such domains [9, 10]. Regardless of the operation domain, all the proposed algorithms aim at improving the invisibility or transparency of the embedded watermark while increasing its robustness against different types of attacks. Another essential requirement is the speed of watermark embedding and extraction [11-13]. This is of ultimate importance to consider the watermarking algorithm practical. Yet another important requirement is the blindness of the algorithm. Digital image watermarking algorithms can be classified according to the watermark extraction process as follows [2]:

- Non-blind algorithms that require both the secret key(s) for watermark embedding and the original host image.
- Semi-blind algorithms that require both the secret key(s) for watermark embedding and the watermark.
- Blind algorithms that require only the secret key(s) for watermark embedding. Neither the original host image nor the watermark is needed.

From the above discussion, it is clear that an ideal digital image watermarking algorithm should have the following characteristics:

- should not cause any changes to the original image; in other words, the watermark should not be really embedded.
- should be robust to most common unintentional attacks and to intentional attacks.
- should be fast.
- should be blind with respect to the extraction process of the embedded watermark.

This paper proposes a novel algorithm for embedding a monochrome watermark into a grayscale host image in the DCT domain with the following characteristics:

- The algorithm is transparent since the watermark is not physically embedded into the host image. Instead, verification information is extracted for future extraction of the watermark.

- The algorithm is robust to common unintentional attacks such as filtering, noise addition, and compression not only by watermarking the low frequency components, but also by intelligently exploiting the most robust signs of the DC components of the DCT-transformed host-image blocks.

- The algorithm is very fast; faster than all the other algorithms we have encountered in the literature.

- The algorithm is blind since it only requires two transparencies or shares for watermark extraction. One of the transparencies or shares is public, while the other is secret and is registered to the Certified Authority (CA) for additional security and protection against unintentional attacks. But, unlike other algorithms in the literature, generating the transparencies is straightforward and fast. Also, the size of each share is equal to half the watermark size.

The paper also utilizes Torus Automorphism (TA) permutation [14-16] to scramble the watermark before embedding and to reassemble it after extraction. This helps increase robustness to intentional attacks while preserving blindness. The paper also proposes an extended version of this technique to increase the robustness against intentional attacks even further. The algorithm is discussed in the rest of the paper.

The paper is organized as follows: Section 2 describes the proposed algorithm. Section 3 provides experimental results that demonstrate the capabilities and effectiveness of the proposed algorithm. Section 4 discusses and compares related research in the literature. Finally, Section 5 provides the discussion and conclusions of the paper.

2. THE PROPOSED ALGORITHM

The proposed algorithm can be described in terms of the watermark embedding process and the watermark extraction process.

2.1 The Watermark Embedding Process

In the Discrete Cosine Transform (DCT) domain, the most significant components (DC or low frequencies) contain more energy than the insignificant components (high frequencies). Also, the human perception system is more sensitive to the low frequency components. Thus, if the watermark is embedded in the low frequencies, the embedded watermark would be robust to unintentional attacks but unfortunately the watermark may be difficult to hide and this may degrade the watermarked image quality. On the other hand, if the watermark is embedded in the high frequencies, it would be easier to hide the watermark, but the embedded watermark may be less resilient to unintentional attacks. The proposed algorithm solves this dilemma by utilizing the low frequencies in the embedding process without physically embedding the watermark bits as explained below.

A DC component in a given DCT-transformed block is the upper left element in the block. This element has the largest absolute value among the other elements and thus its sign (positive or negative) is most unlikely to change under unintentional attacks. The proposed algorithm divides the host image into non-overlapping 4*4 block and DCT-transform these blocks from the spatial domain to the frequency domain. It then intelligently utilizes the signs of the DC components of the DCT-transformed blocks together with the watermark bits to generate verification information that is used as secret keys for later extraction of the watermark. The steps of the proposed watermark embedding process can be explained as follows:

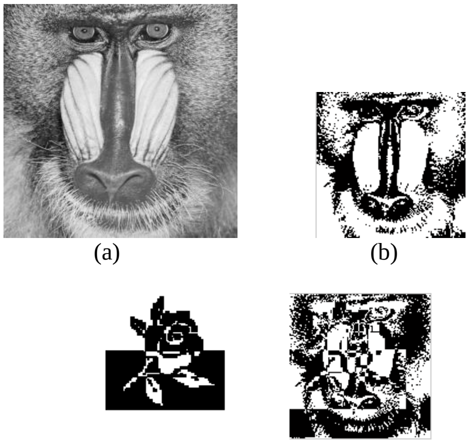

(a) (b)
(c) (d)

Fig 1: (a) The baboon host image, (b) its reduced binary image, (c) the rose watermark image, and (d) the secret key image resulting from XORing the two images (b) and (c).

- Step 1: Divide the host image into non-overlapping 4 * 4 blocks.

- Step 2: Transform each block from the spatial domain to the frequency domain using DCT transform according to the following equation:

$$\mathrm{DCT}(i,j) = C(i) * C(j) * \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \mathrm{pixel}(x,y) * \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \qquad (1)$$

$$\text{where } C(i), C(j) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } i,j=0 \\ \sqrt{\frac{2}{N}} & \text{otherwise} \end{cases}$$

In fact, we do not need to compute all the 16 elements of each DCT-transformed block; only the DC component is needed. This reduces the time of the DCT transform to 1/16 of its value.

- Step 3: If the DC component in a given DCT-transformed block is negative, replace the block by a 0 bit. Otherwise, replace it by a 1 bit. Since each block is of size 4*4 and is replaced by a single bit, the original host image is replaced by a reduced binary image of size equal to 1/16 of the size of the host image. For example, if the host image is of size 512*512, the host image is replaced by a reduced binary image of size 128*128. Figure 1 shows the host image baboon and its corresponding reduced binary image. Similarly, Figure 2 shows the host image boat and its corresponding reduced binary image.

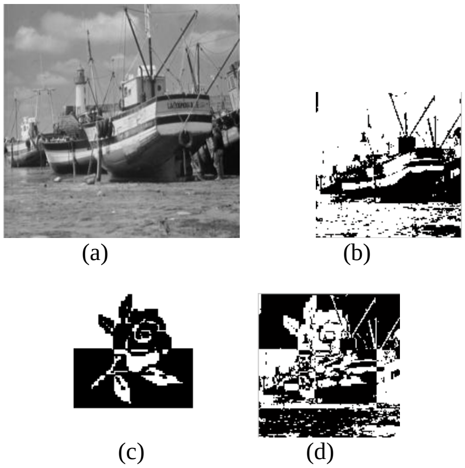


Fig 2: (a) The boat host image, (b) its reduced binary image, (c) the rose watermark image, and (d) the secret key image resulting from XORing the two images (b) and (c).

- Step 4: Resize the binary watermark to be of the same size as the size of the reduced binary image.

- Step 5: XOR each bit of the reduced binary image with the corresponding bit in the watermark image. The result is a secret key image. Figure 1 shows the watermark image rose and the secret key image obtained by XORing it with the reduced binary image of the baboon host image. Similarly, Figure 2 shows the watermark image rose and the secret key image obtained by XORing it with the reduced binary image of the boat host image.

It is clear that the watermark bits are not really embedded in the host image, but are used together with the reduced binary image to extract the secret key image to be used later for watermark extraction. In other words, the embedded watermark is transparent or invisible and thus the quality of the watermarked host image is not degraded by the embedding process.

- Step 6: This secret key image is decomposed into two shares. One share is kept with the user and one is registered to the Certified Authority (CA) for additional security and protection against intentional attacks. The size of each share is thus equal to ½ the size of the watermark image.

2.2 The Watermark Extraction Process

The steps of the proposed watermark extraction process are the reverse of the steps of the watermark embedding process. They can be explained as follows:

- Step 1: Reassemble the two shares to generate the secret key image.

- Step 2: Divide the attacked host image into non-overlapping 4 * 4 blocks.

- Step 3: Transform each block from the spatial domain to the frequency domain using Discrete Cosine Transform (DCT).

- Step 4: If the DC component in a given DCT-transformed block is negative, replace the block by a 0 bit. Otherwise, replace it by a 1 bit to obtain a reduced binary image.

- Step 5: XOR each bit of the reduced binary image with the corresponding bit in the secret key image to obtain the extracted watermark.

It is clear that only the two shares are needed for extracting the embedded watermark. Neither the original host image nor the watermark is needed. The proposed algorithm is thus blind.

2.3 Extended Torus Automorphism Permutation

To increase the robustness of the embedded watermark against intentional attacks, we can use Torus Automorphism (TA) permutation [14-16] to disarrange the watermark bits equally and randomly before embedding and reconstruct it after extraction. This scheme offers cryptographic protection against intentional attacks since the keys utilized in TA permutation (for scrambling the watermark) are also necessary in inverse TA permutation (for reconstructing the watermark after extraction). The watermark is scrambled using the following equation before it is embedded into the host image:

$$\begin{pmatrix} i^{*} \\ j^{*} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ k & k+1 \end{pmatrix} * \begin{pmatrix} i \\ j \end{pmatrix} \bmod m \qquad (2)$$

Equation (2) indicates that each bit of the watermark at location (i, j) will be moved to a new location (i*, j*). Parameter m is obtained from the size m*n of the watermark while parameter k is arbitrarily chosen by the user. Parameters m and k are secret keys needed for both the scrambling and reconstruction of the watermark. Even with TA permutation, the proposed algorithm is still blind.

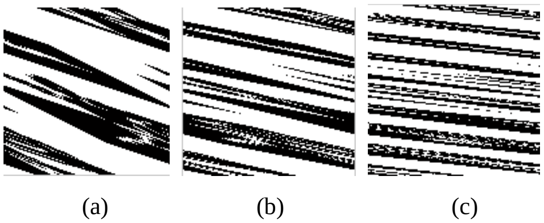
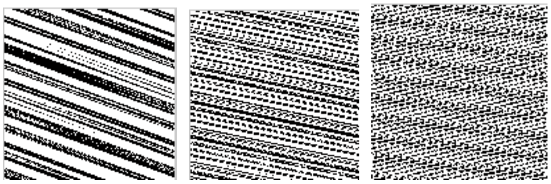


Fig 3: The rose watermark image scrambled using TA permutation with parameters m=128, t=1; and (a) k=2, (b) k=4, and (c) k=8.

Another parameter t can be the number of iterations of TA permutation [1]. Assuming m=128 and t=1, the scrambled rose watermark using TA permutation is shown in Figure 3 for k=2, 4, and 8. Similarly, Figure 4 shows the same watermark scrambled using parameters m=128, k=2; and t=2, 4, and 8.

In this paper, we propose an extended version of TA permutation where each iteration has a different value for k. Figure 5 shows the scrambled rose watermark using TA permutation using parameters m=128, t=3; and k=2, 4, and 8 in iterations 1, 2, and 3 respectively. Since the number of parameters increases, the robustness of the embedded watermark against intentional attacks also increases. It is worth noting that when reassembling the extracted watermark, the values of k have to be applied in the reverse order.



(a) (b) (c)

Fig 4: The rose watermark image scrambled using TA permutation with parameters m=128, k=2; and (a) t=2, (b) t=4, and (c) t=8.

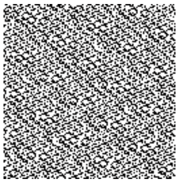


Fig 5: The rose watermark image scrambled using extended TA permutation with parameters m=128, t=3; and k(1)=2, k(2)=4, and k(3)=8.

3. EXPERIMENTAL RESULTS

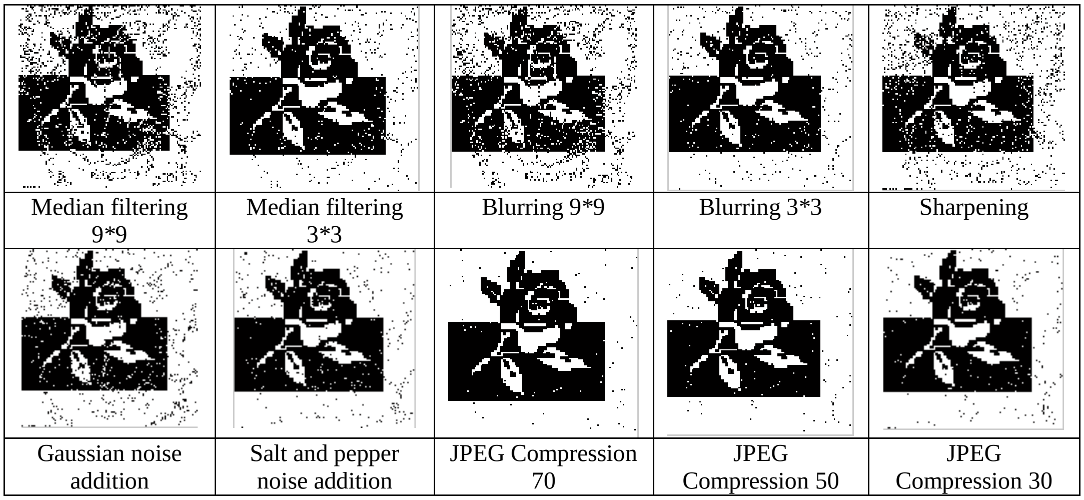
This section provides experimental results that demonstrate the capabilities of the proposed algorithm. Two host images have been used in the experiments. These are the baboon image and the boat image shown in Figures 1(a) and 2(a) respectively. Each of these images is of size 512*512. The rose watermark shown in Figures 1(c) and 2(c) is of size 128*128. It has been used for watermarking these two host images.

Figures 6 and 7 show the extracted watermarks from the baboon host image and the boat host image respectively under different unintentional attacks. It is clear that the watermark, embedded using the proposed algorithm, is robust to these different types of attacks. This is not only due to embedding the watermark bits in the low frequencies, but also due to intelligently utilizing the very robust signs of the DC components in the DCT-transformed host image blocks.

The algorithm has also been shown to be fairly robust to the less common geometric attacks such as scaling, cropping, and rotation, but the attacked image has to be re-rotated and re-scaled to its original size before watermark extraction. The results are shown in Figures 8 and 9 for the baboon and the boat host images respectively.

Figure 10 shows the extracted watermark from the baboon image when using the secret key image of the boat image or from the boat image when using the secret key image of the baboon image. The same watermark is extracted in both cases since in both cases it is equivalent to the result of XORing the reduced baboon image with the reduced boat image with the rose watermark.

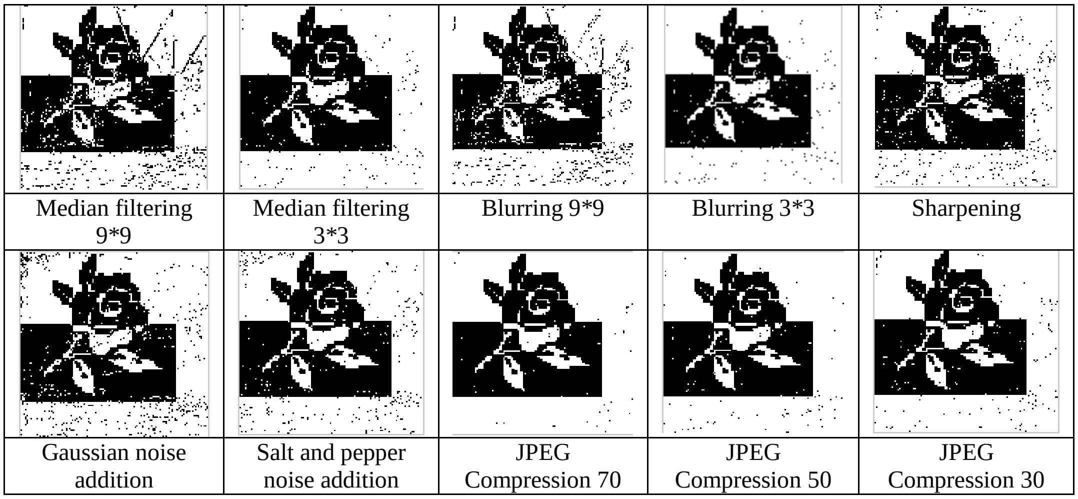
The reported time is as follows on a laptop with Intel Core 2 Duo, 2.40 GHz clock, and 4 GB RAM: An average of only 0.16 seconds for embedding, an average of only 0.13 seconds for extraction, and an average of only 0.08 seconds for one cycle of both TA scrambling and reassembling.



Median filtering 9*9 | Median filtering 3*3 | Blurring 9*9 | Blurring 3*3 | Sharpening

Gaussian noise addition | Salt and pepper noise addition | JPEG Compression 70 | JPEG Compression 50 | JPEG Compression 30

Fig 6: The extracted watermarks from the baboon image under different common attacks.



Median filtering 9*9

Median filtering 3*3

Blurring 9*9

Blurring 3*3

Sharpening

Gaussian noise addition

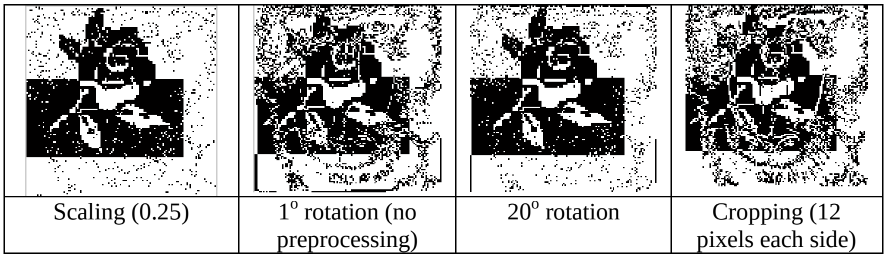
Salt and pepper noise addition

JPEG Compression 70

JPEG Compression 50

JPEG Compression 30

Fig 7: The extracted watermarks from the boat image under different common attacks.



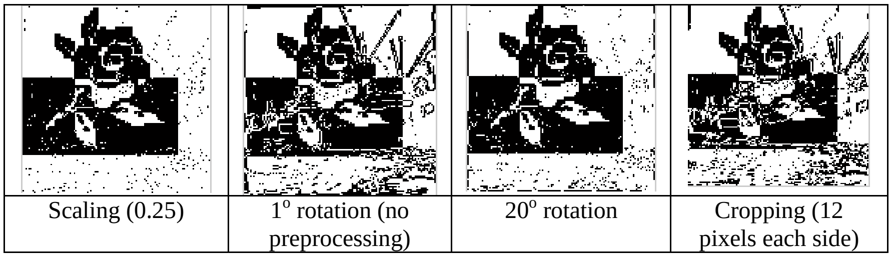
Scaling (0.25)

1^o rotation (no preprocessing)

20^o rotation

Cropping (12 pixels each side)

Fig 8: The extracted watermarks from the baboon image under geometric attacks.



Scaling (0.25)

1^o rotation (no preprocessing)

20^o rotation

Cropping (12 pixels each side)

Fig 9: The extracted watermarks from the boat image under geometric attacks.

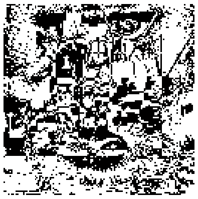


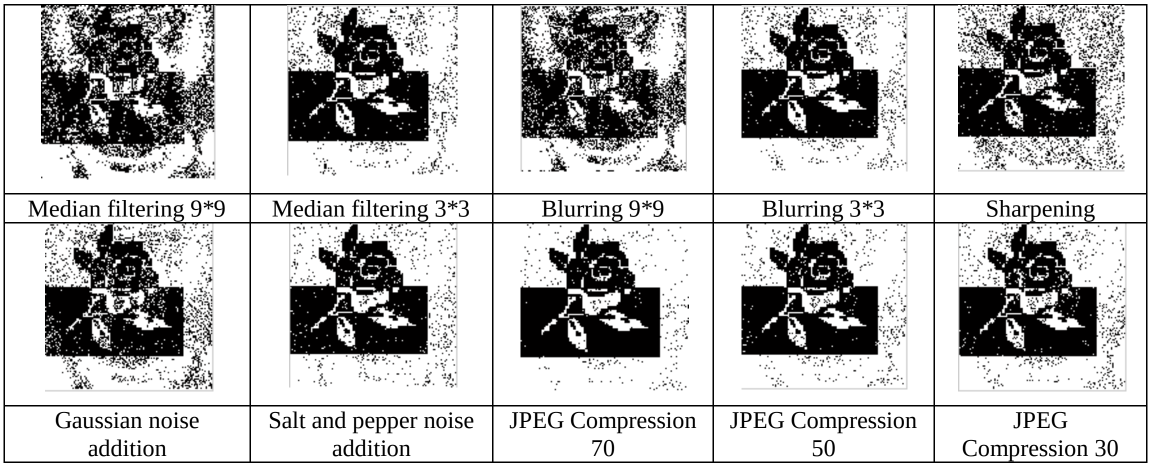
Fig 10: The extracted watermark from the baboon image when using the secret key image of the boat image or from the boat image when using the secret key image of the baboon image.

4. RELATED WORK

Many algorithms in the literature have been proposed for digital image watermarking in the DCT-domain for copyright protection. These algorithms attempted to utilize the low or intermediate frequency components in different ways to achieve robustness against unintentional attacks. For example, Lin et al [5] adjust the DCT low-frequency coefficients by the concept of mathematical remainder to preserve acceptable visual quality of the watermarked image, but it deals only with JPEG compression. Patra et al. [6] randomly select one of the four low frequency locations for embedding a given watermark bit based on the Chinese Remainder Theorem (CRT), but this algorithm is not robust to the most common attack of noise addition. Our algorithm utilizes the most robust signs of the DCT-transformed blocks of the host image in the embedding process to increase robustness against unintentional attacks.

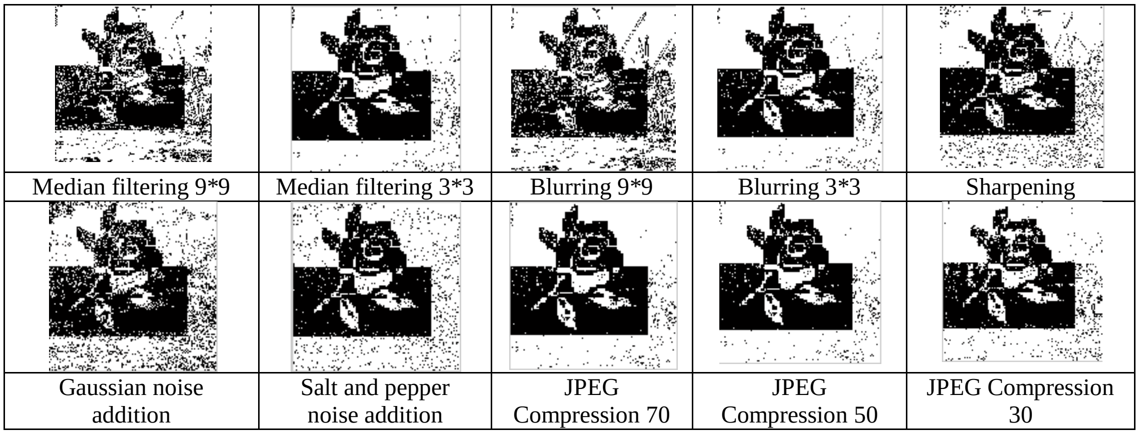
For the sake of transparency, many algorithms in the literature called for avoiding the physical embedding of the watermark into the host image. Instead, verification information is extracted to help in later extraction of the embedded watermark. A common verification scheme is the generation of two transparencies or shares. For example, Naor et al. [17] replaced each bit of a given binary image with 2*2 bits. In other words, a binary image with M by N bits can be divided into two sharing images with 2M by 2N bits. Chang et al. [1] modified this scheme to be suitable for grayscale images. But, the process of generating the shares was very time-consuming and the quality of the extracted watermark was very poor under different common attacks. Alternatively, Hu et al. [2] utilize the pixel values of the original grayscale image to construct a grayscale watermark image. A binary watermark image is further retrieved via the grayscale watermark from the first phase. Though this algorithm is robust, it is semi-blind. The grayscale watermark is needed in the extraction process. The problem with such algorithms is that generating the shares is always time consuming. Besides, the size of each share is usually much larger than the size of the original watermark. In our proposed algorithm, generating the shares is straightforward and the size of each share is equal to half the size of the watermark.

Concerning speed, Patra et al. [6] reported an extraction time similar to that of our algorithm. But, the embedding time is about 50% larger than that of our algorithm. This is in addition to the problems discussed above. Ganesan [11] developed a fast algorithm in the DCT domain, but the reported embedding time is about three times that of our algorithm even on a DSP board. Besides, the algorithm is a non-blind algorithm that requires the host image during extraction. Coltuc et al. [12] claimed a fast algorithm though they did not report the embedding and extraction time. The algorithm, however, is not transparent and its robustness to different types of attacks is not verified. Naderahmadian et al. [13] developed another fast algorithm, but the reported embedding time is about 900 times the speed of the embedding process of our algorithm.



Median filtering 9*9
Median filtering 3*3
Blurring 9*9
Blurring 3*3
Sharpening
Gaussian noise addition
Salt and pepper noise addition
JPEG Compression 70
JPEG Compression 50
JPEG Compression 30

Figure 11. The extracted watermarks from the Boboon image under different attacks using the algorithm in [18].



Median filtering 9*9
Median filtering 3*3
Blurring 9*9
Blurring 3*3
Sharpening
Gaussian noise addition
Salt and pepper noise addition
JPEG Compression 70
JPEG Compression 50
JPEG Compression 30

Figure 12. The extracted watermarks from the Boat image under different attacks using the algorithm in [18].

The algorithm that is closest to ours is that of Elazhary et al. [18]. This algorithm utilized the more robust relative values of the low frequencies instead of the absolute values for embedding the watermark bits. They also used TA permutation to increase robustness. Besides, they generated two shares similar to our algorithm. But, the size of their shares is equal to that of the watermark. The size of the shares in our proposed algorithm is half that of the watermark. Also, their algorithm suffered from the low extracted watermark quality in comparison to ours. Figures 11 and 12 show the watermarks extracted using their algorithm. By comparing these two figures to figures 6 and 7 respectively, it is clear that the quality of the watermarks extracted using our proposed algorithm is superior. Besides, their algorithm is very slow.

Table 1 shows the processing time required by different algorithms discussed above. The table shows the superiority of our algorithm.

Table 1. A comparison of the processing time of different algorithms

Algorithm	Embedding Time	Extraction Time	Notes
Patra et al. [6]	245 ms (0.245 sec)	129 ms (0.129 sec)	smaller watermark
Ganesan [11]	0.5 sec	extraction time not reported	DSP board
Naderahmadian et al. [13]	146 sec	extraction time not reported	smaller watermark; P4, 2.80 GHz clock, 1.25 GB RAM
Elazhary et al. [18]	19.42 min (1165 sec)	81 min (4860 sec)	
The proposed algorithm	0.16 sec	0.13 sec	

5. DISCUSSION & CONCLUSIONS

This paper presented a novel algorithm for grayscale digital image watermarking using monochrome watermarks in the DCT domain. The algorithm embeds the watermark bits in the low frequencies to increase the robustness of the embedded watermark against common unintentional attacks. But, to preserve transparency so as not to degrade the quality of the watermarked host image, the watermark bits are not physically embedded. Rather, the algorithm exploits the most robust signs of the DC components of the DCT-transformed host image blocks together with the watermark bits in extracting a secret key image that can be used later for watermark extraction. Thus, the two contradictory goals of robustness against unintentional attacks and transparency are achieved.

Another advantage of the proposed algorithm is that it is a blind algorithm that requires only two transparencies or shares to extract the embedded watermark. One of the transparencies is public, while the other is secret and is registered to the Certified Authority (CA) for additional security and protection against intentional attacks. Unlike other algorithms in the literature, generating the transparencies is straightforward and fast and the size of each share is equal half the watermark size.

To increase the robustness of the embedded watermark against intentional attacks, Torus Automorphism (TA) permutation is used to scramble the watermark before embedding and reassemble it after extraction. The paper proposes an extended version of TA permutation for additional security against intentional attacks. The use of TA permutation preserves the blindness of the proposed algorithm since it only results in additional parameters needed for watermark extraction.

The proposed algorithm is also practical. On a laptop with Intel Core 2 Duo, 2.40 GHz clock, and 4 GB RAM, it requires an average of only 0.16 seconds for embedding, an average of only 0.13 seconds for extraction, and an average of only 0.08 seconds for one cycle of both TA scrambling and reassembling. The embedding and extraction times are lower than that of all the other algorithms that we have encountered in the literature.

Experimental results proved the robustness of the algorithm against common unintentional attacks such as filtering, noise addition, and JPEG compression. Probably, the main drawback of the algorithm is that it is less robust to the less common geometric attacks such as scaling, cropping, and rotation and the attacked image has to be re-rotated and re-scaled to its original size before watermark extraction. We are now working on modifying our algorithm to increase the robustness of the embedded watermark to geometric attacks without preprocessing of the attacked host image. But this is expected to increase the speed of the algorithm.

6. REFERENCES

[1] Chang, C. and Chuang, J. 2002. An Image Intellectual Property Protection Scheme for Gray-Level Images Using Visual Secret Sharing Strategy. Pattern Recognition Letters, 23 (2002), 931–941.

[2] Hu, M., Lou, D., and Chang, M. 2007. Dual-Wrapped Digital Watermarking Scheme for Image Copyright Protection. Computers & Security, 26 (2007), 319-330.

[3] Hussein, J. 2010. Spatial Domain Watermarking Scheme for Colored Images Based on Log-Average Luminance. Journal of Computing, 2(1) (2010).

[4] Saryazdi, S. and Demehri, M. 2005. A Blind DCT Domain Digital Watermarking. In Proceedings of the 3rd International Conference on Sciences of Electronic, Technologies of Information and Telecommunications, (2005).

[5] Lin, S., Shie, S., and Guo, J. 2010. Improving the Robustness of DCT-Based Image Watermarking against JPEG Compression. Computer Standards & Interfaces, 32 (2010), 54–60.

[6] Patra, J., Phua, J., and Bornand, C. 2010. A Novel DCT Domain CRT-Based Watermarking Scheme for Image Authentication Surviving JPEG Compression. Digital Signal Processing, (2010).

[7] Yongqiang, C., Yanqing, Z., and Lihua, P. 2009. A DWT Domain Image Watermarking Scheme Using Genetic Algorithm and Synergetic Neural Network. In Proceedings of the 2009 International Symposium on Information Processing (ISIP'09), (2009), 298-301.

[8] Temi, C., Choomchuay, S., and Lasakul, A. 2005. A Robust Image Watermarking Using Multiresolution Analysis of Wavelet. In proceedings of ISCIT2005, (2005).

[9] Ganic, E. and Eskicioglu, A. 2004. Robust DWT-SVD Domain Image Watermarking: Embedding Data in All Frequencies. In Proceedings of the 2004 workshop on Multimedia and security, (2004).

[10] Bedi, S., Kumar, A., and Kapoor, P. 2009. Robust Secure SVD Based DCT-DWT Oriented Watermarking Technique for Image Authentication. In Proceedings of the International Conference on IT to Celebrate S. Charmonman's 72nd Birthday, (2009).

[11] Ganesan, S. 2006. Real-Time Digital Image Watermarking. TI Developer Conference, (2006).

[12] Coltuc, D. and Chassery, J. 2007. Very Fast Watermarking by Reversible Contrast Mapping. IEEE Signal Processing Letters, 14(4) (2007), 255-258.

[13] Naderahmadian, Y. and Hosseini-Khayat, S. 2010. Fast Watermarking Based on QR Decomposition in Wavelet Domain. In Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, (2010), 127-130.

[14] Voyatzis, G. and Pitas, I. 1996. Applications of Toral Automorphisms in Image Watermarking. In Proceedings of the Image Processing International Conference, (1996).

[15] Chang, C., Hsiao, J., and Chiang, C. 2002. An Image Copyright Protection Scheme Based on Torus Automorphism. In Proceedings of the 1st International Symposium on Cyber Worlds, (2002), 217–224.

[16] Engedy, M., Munaga, V., and Saxena, A. 2006. A Robust Wavelet Based Digital Watermarking Scheme Using Chaotic Mixing. In Proceedings of the 1st International Conference on Digital Information Management, (2006), 36–40.

[17] Naor, N., and Shamir, A. 1994. Visual Cryptography, Advances in Cryptography. Lecture Notes in Computer Science, (1994), 1–12.

[18] Elazhary, H. and Morkos, S. 2010. Blind Robust Transparent DCT-Based Digital Image Watermarking for Copyright Protection. International Journal of Computer Science and Information Security, 8(7) (2010).