Luminance-based Embedding Approach for Color Image Watermarking

Jamal Ali Hussein
Computer Science Department, University of Sulaimani
Sulaimani, Iraq
Email: jamal.hussein@univsul.net

Abstract—In this paper a new non-blind luminance-based color image watermarking technique is proposed. The original 512×512 color host image is divided into 8×8 blocks, and each block is converted to YC_bC_r color space. A 32×32 monochrome image is used as a watermark and embedded in the selected blocks of the original image. The selected blocks must have log-average luminance that is closer to the log-average luminance of the image. DCT transform is applied to the Y component of each selected block. Each four values of the watermark image are embedded into each selected block of the host image. The watermark values are embedded in the first four AC coefficients leaving the DC value unchanged. The watermark is extracted from the watermarked image using the same selected blocks and DCT coefficients that have been used in the embedding process. This approach is tested against variety of attacks and filters: such as, highpass, lowpass, Gaussian, median, salt and peppers, and JPEG compression. The proposed approach shows a great ability to preserve the watermark against these attacks.

Index Terms— Color Image Watermarking, YC_bC_r, Log-Average Luminance, DCT, PSNR.

I. Introduction

Unauthorized distribution of digital media such as audios, videos and images has become the most important challenge that faces media copyright protection. In digital watermarking the copyright data is embedded inside the host images in a manner that is difficult to extract or alter the watermark using various attacks. Watermarking embedding is done either in spatial domain or frequency domain. There are many papers in the literature that use these approaches. A spatial domain method is proposed in [1] by adding the watermark to the two chrominance channels in YC_bC_r color space. The size of pixels needed to add the watermark is reduced to half by dividing the watermark into two parts and each part is added to one chrominance channel. In [2] a 32×32 monochrome image is used as the watermark and embedded in spatial domain to the Y component of YC_bC_r of the host color image. This method shows the robustness when changing the host image to grayscale and some persistency against JPEG compression.

There are many papers that used Discrete Cosine Transform DCT and/or Discrete Wavelet Transform DWT to embed the watermark in the frequency domain. The approach proposed in [3] uses DCT transform to embed an M×M color watermark image in an N×N original image using some secret keys. The permutated watermark is embedded into middle -frequency DCT coefficients of Y component in NTSC color space.

In [4] a scrambled color watermark image is added to luminance component of HVS color space. The luminance is classified into weak, medium and strong luminance. The watermark is added to these luminance values in a manner that is less perceptible to human eyes. The DCT frequency domain is also used in [5] to embed the watermark. This method uses a monochrome watermark image and adds it to a grayscale image using three techniques: Torus Automorphism (TA) permutation, a pseudo-random number generator, and visual cryptography. Experimental results show robustness against several types of intentional and unintentional attacks. However, adding the watermark to arbitrary values of the host image with very high or very low luminance values may affect the imperceptibility of the embedded watermark, especially for color host images. The proposed algorithm in [6] used a combination of DWT and DCT to embed the high energy content of color watermark (i.e. low frequency DCT coefficients) into mid frequency DCT coefficients of high frequency components of multi resolved host color image.

The proposed method in this paper is robust against perceptibility and also robust against various attacks. Watermark transparency (imperceptibility) is achieved by adding the watermark to blocks that have log-average luminance values closer to the log-average luminance of the image. Also, embedding the watermark to the low frequency DCT coefficient values makes this approach robust against different attacks.

II. The Proposed Approach

After dividing the 512×512 host image into 8×8 blocks the following operations are needed and must be performed to accomplish the embedding and extracting processes:

A. RGB to YC$_b$C$_r$ conversion

The RGB color space is converted to YC$_b$C$_r$ color space for each 8×8 block using the equations (1):

$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$

$C_b = 0.596 \times R - 0.275 \times G - 0.321 \times B$

$C_r = 0.212 \times R - 0.523 \times G - 0.311 \times B$ (1)

where R, G, and B are red, green and blue components of RGB color space respectively.

B. Log-average Luminance

The block selection criteria are depended on log-average luminance for the entire image and log-average luminance for each block. The log-average luminance Lavg is calculated as shown in the equation (2):

$$L_{avg} = \exp\left(\sum \log(\delta + Y_{x,y}) / N \right) \quad (2)$$

where

L$_{avg}$: Log-average luminance

Y$_{x,y}$: Luminance Y of the pixel at x,y

δ: A small value to avoid taking the log of a completely black pixel whose luminance is zero

N: The number of pixels in the image

C. Block selection criterion

When 512×512 host image is divided into 8×8 blocks, 4096 blocks are produced. Since the watermark size is 32×32 pixels and only 4 values are added to each block, so to accomplish the watermark embedding process 256 blocks are required. After finding the log-average luminance for the entire image and for each block; the best 256 blocks are chosen from the blocks that have log-average luminance closer to the log-average luminance of the entire image. To do that we select blocks with log-average luminance in the range [L$_{avg}$-β, L$_{avg}$+ β] where L$_{avg}$ is the log average luminance of the image and β is the minimum floating-point value that is enough to determine adequate number of blocks (about 256 blocks). See figure 1.

Image: Fruits	Image: Peppers	Image: Baboon	Image:Whitewater
Size: 512x512	Size: 512x512	Size: 512x512	Size: 512x512
β: 3.224	β: 4.490	β: 2.288	β: 4.099
L$_{avg}$: 154.027	L$_{avg}$: 102.342	L$_{avg}$: 121.019	L$_{avg}$: 80.077
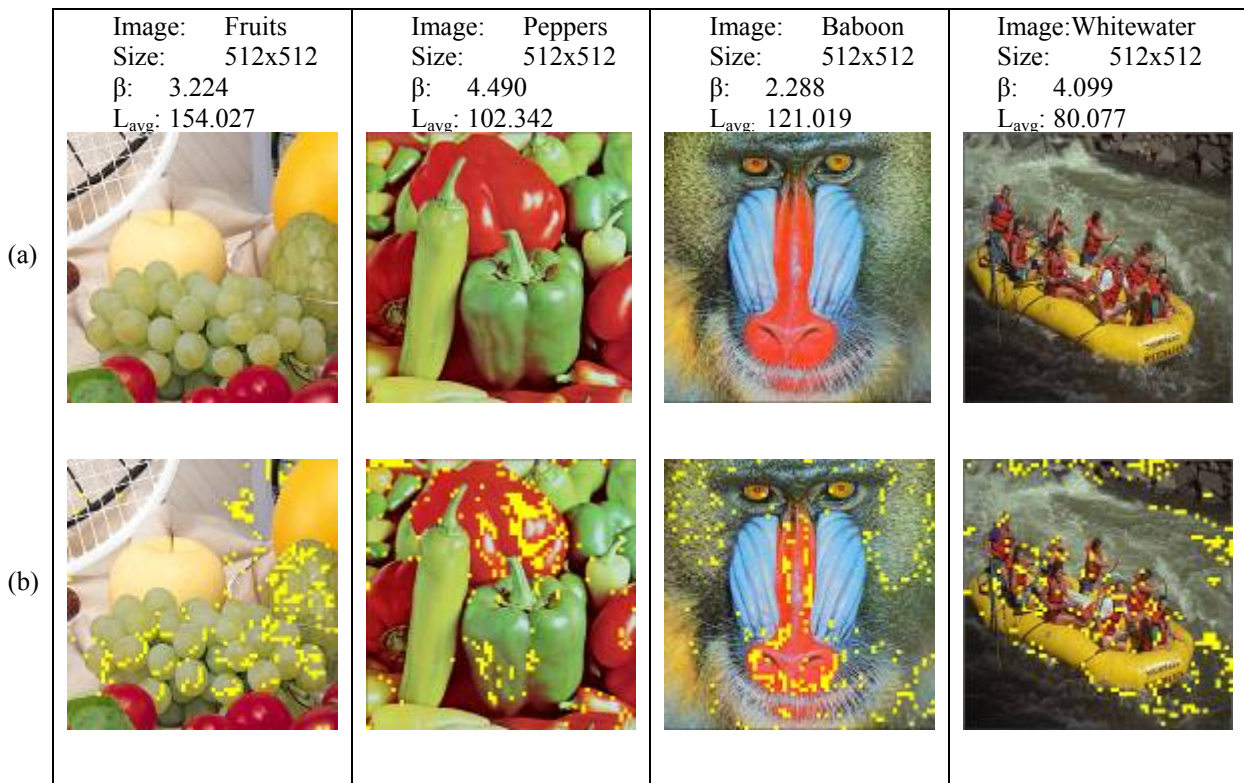
Figure 1. Block Selection Criteria (a) the original images (b) selected blocks (yellow)

D. Forward DCT transform

The DCT transform is applied to the Y component of each selected block using the forward DCT transform shown in the equation (3):

$$D_{u,v} = \frac{1}{\sqrt{2N}} C_u C_v \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} Y_{x,y} \cos\left[\frac{(2y+1)v\pi}{2N}\right] \cos\left[\frac{(2x+1)u\pi}{2N}\right] \quad (3)$$

where

$$C_i = \begin{cases} 1/\sqrt{2} & \text{if } i = 0 \\ 1 & \text{if } i > 0 \end{cases}$$

and $Y_{x, y}$ is the block's Y value and N = 8 for 8X8 blocks.

F. Inverse DCT transform

After the watermark embedding process, The DCT coefficients are converted back to the Y component using the inverse DCT equation (4):

$$Y'_{x, y} = \frac{1}{\sqrt{2N}} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C_u C_v D'_{u,v}$$
$$\cos\left[\frac{(2y + 1)v\pi}{2N}\right] \cos\left[\frac{(2x + 1)u\pi}{2N}\right] \quad (4)$$

where
$$C_i = \begin{cases} 1/\sqrt{2} & \text{if } i = 0 \\ 1 & \text{if } i > 0 \end{cases}$$

and $D'_{u,v}$ is the watermarked block's DCT coefficient and N = 8 for 8X8 blocks.

G. YC_bC_r to RGB conversion

Finally the watermarked image is constructed by converting the YCbCr to RGB using the equations (5):

R' = Y' + 0.956 × C_b + 0.620 × C_r

G' = Y' - 0.272 × C_b - 0. 647 × C_r

B' = Y' - 1.108 × C_b + 1.705 × C_r (5)

III. watermark embedding

The figure 2 illistrates the steps of watermark embedding process.

Each four values of the watermark data is added to the first four coefficient values of each selected block except the upper-left value (DC). The DCT coefficients are selected using a zigzag algorithm as shown in the figure 3.

An addition factor α = 10 is added to or subtracted from the DCT coefficients. If the watermark's pixel is white (i.e. 255) then α is added to the original DCT coefficient. If the pixel is black (i.e. 0) then α is subtracted from the original DCT coefficient. See equation (6).

$$D'_{i,j} = \begin{cases} D_{i,j} + \alpha & \text{where } W_{x,y} = 255 \\ D_{i,j} - \alpha & \text{where } W_{x,y} = 0 \end{cases}$$
                        (6)
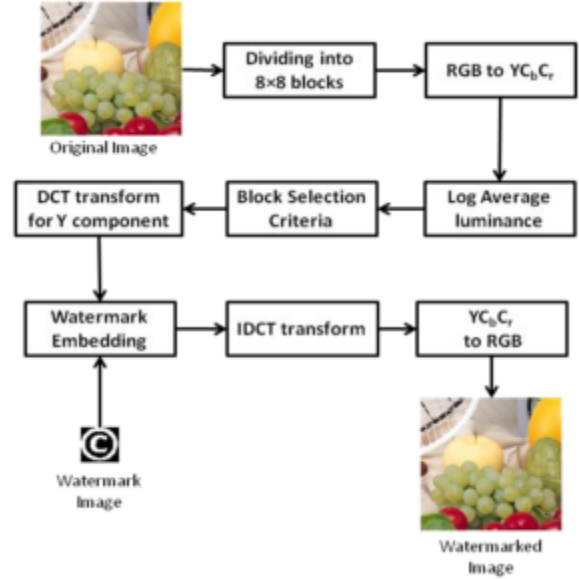
Figure 2. The Embedding Process
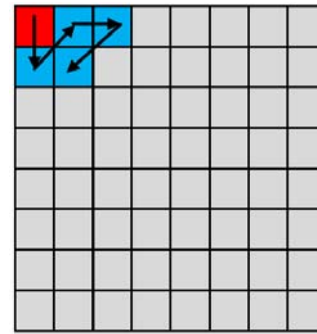
Figure 3. Selected DCT coefficients (blue)

where

$D'_{i,j}$: the selected DCT coefficient after embedding

$D_{i,j}$: the selected DCT coefficient before embedding

α: the addition factor

$W_{x,y}$: the x,y value of the watermark

IV. Quality Measurement

The quality of the watermarked image is measured by finding Mean Square Error MSE and Peak Signal-to-Noise Ratio PSNR values of the watermarked image and the original image. The equations (7) and (8) are used to calculate MSE and PSNR. Lower MSE values and higher PSNR values imply good embedding results.

$$MSE = \frac{1}{NM} \sum_{x=0}^{N} \sum_{y=0}^{M} (I(x, y) - I_w(x, y))^2$$
                        (7)

Where I is the original image and Iw is the watermarked image. The two images are of size (NxM).

$$PSNR(dB) = 10\log_{10}(\frac{(MAX_i)^2}{MSE})$$
(8)

where MAX_i is the maximum possible pixel value of the image.

Figure 4 shows the PSNR value of the watermarked images for five images (Fruits, Peppers, Baboon, Whitewater, and Lena).
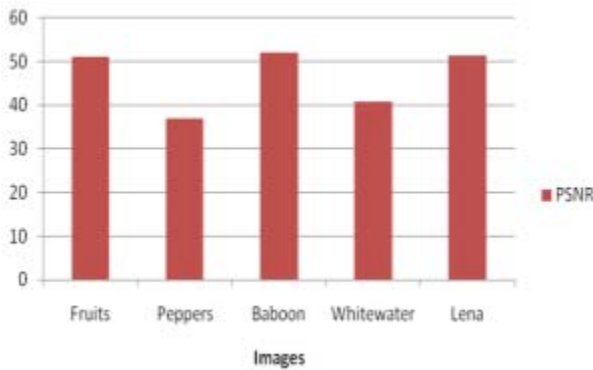
Figure 4. PSNR of five images after embedding the watermark

V. Watermark extraction

The steps of watermark extraction process are shown in figure 5.
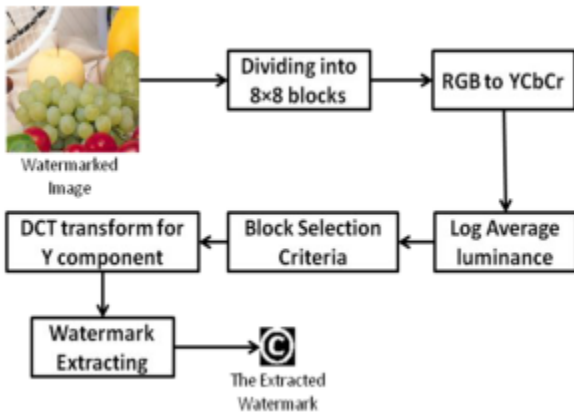
Figure 5. The Extracting Process

To extract the watermark the same steps of embedding process is performed; the watermarked image is converted to YC_bC_r, Y component transformed to DCT coefficients. The watermark is extracted from the same blocks that have been used in the embedding process. The difference between the value of watermarked DCT coefficient and the original corresponding value is calculated. If the result is greater than zero then the watermark color is white, otherwise the black color is assumed. See equation (9).

$$W_{x,y}^{'} = \begin{cases} 255 & \text{where } D_{i,j}^{'} - D_{i,j} \geq 0 \\ 0 & \text{where } D_{i,j}^{'} - D_{i,j} < 0 \end{cases}$$
(9)

where

$W_{x,y}^{'}$: The extracted watermark x,y pixel

$D_{i,j}^{'}$: The i,j pixel Y value of watermarked image

$D_{i,j}$: The i,j pixel Y value of original image

VI. Watermark Similarity

After extracting the watermark image from the watermarked host image, the extracted watermark must be compared to the original one to determine the similarity σ between them using equation (10):

$$\sigma = \frac{1}{NM}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} S_{x.y}$$
(10)

where $S_{x,y} = \begin{cases} 1 & \text{if } W_{x,y}^{'} = W_{x,y} \\ 0 & \text{if } W_{x,y}^{'} \neq W_{x,y} \end{cases}$

and N×M is the size of the watermark (32×32).

W' is the extracted watermark and W is the embedded watermark. The similarity σ varies in the interval [0, 1].

VII. Experimental Results

This approach was evaluated under different type of attacks: filters such as Highpass, Lowpass, Gaussian (radius = 4 and 12), Median (size = 3 and 5), Impulse (salt & pepper, amount = 0.01 and 0.05), and JPEG compression with various qualities, 0.75, 0.50 and 0.25. The chart in the figure 6 shows the PSNR values of three images after the attacks have been applied, while the chart in figure 7 shows the similarity between the extracted watermark and the original embedded watermark for the same attacks.

The efficiency of the proposed method in this paper is compared against JPEG compression with four previous methods. The results, as presented in Table I, show good improvement in similarity between the original and the extracted watermark after compression of the proposed method in comparison with the previous methods.

In Table II, five different images (Fruits, Peppers, Baboon, Whitewater, and Lena) have been used to test the robustness of the proposed watermarking method against variety of attacks. The extracted watermark image, MSE, PSNR, and Similarity are presented for each test as test parameters. These test parameters show reasonable watermark resistance against most of the attacks.

As we see, the attack that mostly distorts the watermarked image is the Impulse (salt and pepper noise) which also affects the embedded watermark. However, this type of noise can be reduced using median filter. Resolving the image using median filter also recovers the embedded watermark. The extracted watermark from the noisy image and the extracted watermark from the recovered image after applying the median filter are shown in Table III.
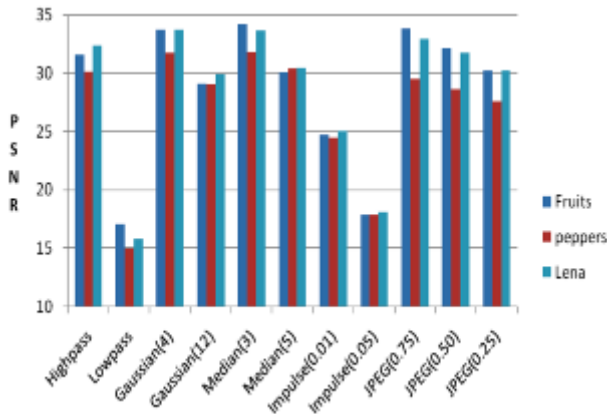
Figure 6. PSNR versus various attacks applied on three images (Fruits, Peppers, and Lena)
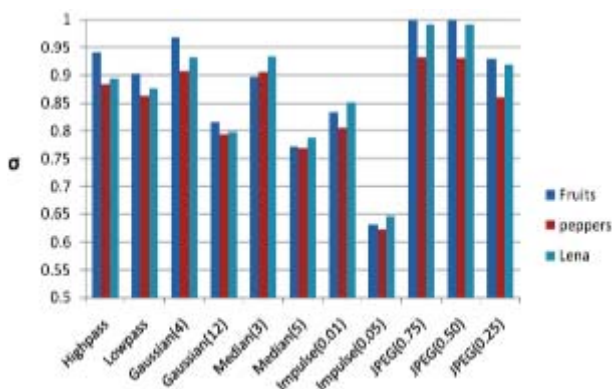
Figure 7. Watermark Similarity versus various attacks applied on three images (Fruits, Peppers, and Lena)

Table I.
JPEG Compression Comparison with previous methods (quality = 0.70)

Methods	PSNR	Similarity
[1]	30.000	0.800
[2]	32.512	0.689
[3]	35.000	0.980
[4]	34.990	0.780
Proposed	32.650	0.986

VIII. Conclusion

In this research paper a new approach is proposed based on log-average luminance and DCT transform. This approach achieves high imperceptibility and robustness of the embedded watermark. The watermark imperceptibility is obtained by using log-average luminance. Only the blocks with log-average luminance close to the log-average luminance of the entire image are used to embed the watermark. The test results show that these blocks do not degrade the image when the pixels' luminance value is increased or decreased. Also, these blocks are less affected when various filters are applied to the image. The robustness is achieved by embedding the watermark in the low-frequency coefficients of block's DCT transform performed on the Y components of YCbCr color space. Adding the watermark to the low-frequency coefficients not only achieve robustness but also prevent the watermark to be affected by JPEG compression.

So, the experimental results show a high degree of robustness against a number of unintentional attacks such as highpass filter, lowpass filter, Gaussian, median, impulse (salt and pepper) and JPEG compression. The efficiency of the embedding process is measured by using PSNR value between the original and the watermarked image. The quality of the watermark image is measured by using similarity degree between the original and the extracted watermark.

The proposed method in this paper is compared with four previous papers in the watermarking field and shows a significant enhancement of the watermark robustness against JPEG compression.

The future work may focus on finding a criterion to select best DCT coefficients for watermark embedding. This selection criterion must balance between watermark robustness and imperceptibility.

References

[1] K. Thongkor and T. Amornraksa , "Method of Image Watermarking by Dividing Watermark into Two Chrominance Channels", IEEE ISCIT conference 2010, pp. 498-503.

[2] J. A. Hussein, "Spatial Domain Watermarking Scheme for Colored Images Based on Log-average Luminance", Journal of Computing, vol. 2, No. 1, 2010, pp. 100-103.

[3] N. Ahmidi and R. Safabakhsh, "A Novel DCT-based Approach for Secure Color Image Watermarking", Proceedings of the International Conference on Information Technology: Coding and Computing, 2004, Vol.2, pp. 709 - 713.

[4] Ting Zhang and Yi Du, "A Digital Watermarking Algorithm for Color Images Based on DCT", International conference on Information Engineering and Computer Science, 2009, pp. 1 – 4.

[5] H. Elazhary and S. Morkos, "Blind Robust Transparent DCT-Based Digital Image Watermarking for Copyright Protection", (IJCSIS)

International Journal of Computer Science and Information Security, Vol. 2010, 8, No. 7, pp. 183-188.

[6] S. Chavan, R. Shah, R. Poojary, J. Jose and G. George, "A Novel Robust Colour Watermarking Scheme for Colour watermark images in Frequency Domain", International Conference on Advances in Recent Technologies in Communication and Computing, 2010, pp. 96-100.

[7] J. Manoharan, C.Vijila and A.Sathesh, "Performance Analysis of Spatial and Frequency Domain Multiple Data Embedding Techniques towards Geometric Attacks", International Journal of Security (IJS), Vol. 4, No. 3, 2010, pp. 28-37.

[8] P. Mohanty, P. Guturu, E. Kougianos, and N. Pati, "A Novel Invisible Color Image Watermarking Scheme using Image Adaptive Watermark Creation and Robust Insertion-Extraction". In proceeding of the 8th IEEE International Symposium on Multimedia (ISM), 2006, pp. 153-160.

[9] Ali Al-Haj, "Combined DWT-DCT digital image watermarking", Journal of Computer Science, vol. 3, no. 9, 2007, pp. 740-746.

[10] Y. Wang, J.F. Doherty and R.E.V. Dyck, "A Wavelet Based Watermarking Algorithm for Ownership Verification of Digital Images," IEEE trans. Image Processing, 2002, vol. 11, no. 2, pp. 77-88.

[11] Saraju P. Mohanty, Parthasarathy Guturu, Elias Kougianos, and Nishikanta Pati, "A Novel Invisible Color ImageWatermarking Scheme using Image Adaptive Watermark Creation and Robust Insertion-Extraction", 8th IEEE International Symposium on Multimedia, 2006, pp. 153 – 160.

[12] Yigang ZHOU and Jia LIU, "Blind Watermarking Algorithm Based on DCT for Color Images", IEEE Image and Signal Processing Conference, 2009, pp. 1-3.

Jamal A. Hussein he has MSc in Computer Science 2007 from Computer Deparment / College of Science / University of Sulaimani. He has been teaching Computer Graphics and OOP. In his research work has been focusing on ecommerce security and Digital Watermarking. He is a Member of Scientific Committee at Computer Department.

Table II.
The watermark quality against number of attacks(W' is the extracted watermark and σ is the similarity)

Images		Attacks										
		Highpass	Lowpass	Gaussian (4)	Gaussian (12)	Median (3)	Median (5)	Impulse (0.01)	Impulse (0.05)	JPEG (0.75)	JPEG (0.50)	JPEG (0.25)
Fruits	W'											
	MSE	43.051	1229.736	26.474	77.225	23.664	61.349	210.588	1026.843	26.049	38.324	59.362
	PSNR	31.619	17.061	33.731	29.081	34.218	30.081	24.724	17.844	33.801	32.124	30.224
	σ	0.941	0.902	0.968	0.816	0.929	0.771	0.834	0.631	0.999	0.999	0.929
Peppers	W'											
	MSE	59.910	1963.667	41.830	77.658	40.785	57.008	221.959	1028.014	69.950	85.348	109.584
	PSNR	30.184	15.028	31.744	29.057	31.854	30.399	24.496	17.839	29.511	28.647	27.561
	σ	0.884	0.863	0.907	0.793	0.905	0.769	0.805	0.623	0.931	0.930	0.861
Baboon	W'											
	MSE	335.741	5155.979	163.552	388.756	310.985	532.158	192.001	954.222	156.344	213.435	291.041
	PSNR	22.699	10.836	25.822	22.062	23.031	20.698	25.126	18.162	26.018	24.666	23.319
	σ	0.788	0.749	0.849	0.679	0.792	0.620	0.851	0.639	0.993	0.994	0.938
Whitewater	W'											
	MSE	124.864	3018.278	82.752	178.865	99.107	198.630	207.686	1002.831	47.131	78.952	125.751
	PSNR	26.994	13.161	28.781	25.434	27.998	24.978	24.785	17.947	31.226	28.985	26.964
	σ	0.805	0.693	0.868	0.677	0.815	0.640	0.852	0.642	0.984	0.985	0.916
Lena	W'											
	MSE	36.116	1661.394	26.378	64.050	26.815	56.614	196.465	961.241	31.793	41.582	59.601
	PSNR	32.382	15.754	33.746	29.894	33.675	30.430	25.026	18.130	32.936	31.770	30.206
	σ	0.894	0.877	0.932	0.797	0.933	0.788	0.852	0.646	0.992	0.992	0.919

Table III.
Recovering the watermark using median filter after adding different Impulse noises
(W' is the extracted watermark and σ is the similarity)

Images		Impulse (0.01)	Median (3)	Impulse (0.05)	Median (3)	Impulse (0.10)	Median (3)
Fruits	W'						
	MSE	210.588	24.490	1026.843	28.337	1997.946	35.523
	PSNR	24.724	34.069	17.844	33.435	14.953	32.454
	σ	0.834	0.923	0.631	0.931	0.591	0.905
Peppers	W'						
	MSE	221.959	42.015	1028.014	48.247	2013.059	57.009
	PSNR	24.496	31.725	17.839	31.124	14.920	30.399
	σ	0.805	0.900	0.623	0.883	0.585	0.873
Baboon	W'						
	MSE	192.001	314.689	954.222	331.136	1835.237	355.958
	PSNR	25.126	22.980	18.162	22.759	15.322	22.445
	σ	0.851	0.794	0.639	0.737	0.596	0.742
Whitewater	W'						
	MSE	207.686	101.632	1002.831	109.689	1936.842	120.566
	PSNR	24.785	27.888	17.947	27.557	15.088	27.147
	σ	0.852	0.804	0.642	0.777	0.574	0.758
Lena	W'						
	MSE	196.465	27.453	961.241	30.074	1874.236	36.557
	PSNR	25.026	33.573	18.130	33.177	15.231	32.329
	σ	0.852	0.934	0.646	0.914	0.601	0.884