7

Using Digital Watermarking for Copyright Protection

Charlie Obimbo and Behzad Salami
University of Guelph
Canada

1. Introduction

Without a doubt, the Internet has revolutionized the way we access information and share our ideas via tools such as Facebook, twitter, email, forums, blogs and instant messaging. The Internet is also an excellent distribution system for digital media. It is inexpensive, eliminates warehousing and delivery, and is almost instantaneous. Together with the advances of compression techniques such as JPEG, MP3 and MPEG; the Internet has become even faster, easier and more cost effective to distribute digital media such as audio, video, images and documents over the World Wide Web.

In addition to existing web sites and shared networks, the recent development of peer-to-peer (P2P) file distribution tools such as Kazaa, Limewire, Exceem or eMule enables a copious number of web users to easily access and share terabytes of digital media across the globe. These technologies also significantly reduce the efforts of pirates to illegally record, sell, copy and distribute copyright-protected material without compensating the legal copyright owners.

Today, content owners are eagerly seeking technologies that promise to protect their rights and secure their content from piracy, unauthorized usage and enable the tracking and conviction of media pirates. Cryptography is probably the most common method of protecting digital content [Koch & Zhao, 1995], where the content is encrypted prior to delivery and a decryption key is provided to those who have purchased legitimate copies. However, cryptography cannot help the content providers monitor their goods after the decryption process; a pirate could easily purchase a legit copy and then re-sell it or distribute it for free over a shared network.

It is therefore important to find a way to protect these digital media with a more stringent method, which would enable the vendors and artists / photographers / directors get confidence in placing and distributing their material over the Internet. Watermarking could be such a vehicle.

2. Overview

Digital watermarking is a field that refers to the process of embedding digital data directly onto multimedia objects such that it can be detected or extracted later.

It has three unique advantages over other techniques such as cryptography. First of all, it is imperceptible and does not affect the aesthetic of the digital data. Secondly, watermarks become fused with the actual bits of the work, unlike headers they do not get removed when the work is displayed, copied or during format changes. Lastly, they undergo the same transformation as the work itself and sometimes the extracted mark can be used to learn about the history of transformations that the work has undergone.

In general any watermarking system consists of three components

a. Watermark generation stage,
b. encoding and
c. decoding [12].

Watermarking can be applied to various digital multimedia such as images [Wolfgang et. al, 1999 & Hartung & Kutter, 1999], videos [Ren-Hou et. al., 2005 & Lie et. al., 2006], audio [Liu & Innoue, 2003 & Berghel, 1997], or text [Huang & Wu, 2004]. Image watermarking is either perceptible or imperceptible to the human eye and can be designed to be robust, fragile or semi-fragile [Koch & Zhao, 1995].

An example of a basic visible watermark would be placing a text or logo onto an image to identify it's rightful copyright owner (see Figure 1). As seen in Figure 1, an image can be placed on the web in low resolution as an advertisement. The purchaser would then receive a copy minus the watermark, on completion of the purchase, from the vendor.

Fig. 1. Example of a visible watermark

Visible marks are usually embedded in the spatial domain, that is, directly onto the pixel values of an image. Clearly, this method is fragile and can easily be compromised by cropping or replacing the text using either a basic image processing tool such as Microsoft Paint, advanced software such as Adobe Photoshop or sophisticated Algorithms such as Huang & Wu's [Huang & Wu, 2004 & Baaziz, 2005].

As a result various other domains have been proposed. In current literature, the watermark is added to the image either in the spatial domain or in a transform domain [Leighton et. al. 1997]. Example of transform domains are discrete Fourier transform (DFT), the full-image discrete cosine transform (DCT) [Bartolini et. al., 2001], the block-wise DCT [Wolfgang et. al, 1999], the discrete wavelet domain (DWT) [Cappellini et. al., 1998], fractal domain [Puate et. al., 1996 & Shahraeini and Yaghoobi, 2012], the redundant contourlet transform [Leighton et. al., 1997], the Hadamard domain, Fourier- Mellin domain or the Radon domain [Lie et. al., 2006]. It has been shown that embedding the mark in the mid-frequencies of a transform domain is advantageous in terms of visibility and security over the spatial domain [Cheng et.al., 1999].

In the embedding stage visibility artifacts must be avoided and thus the Human Visual System (HVS) must be taken into account. The watermark is generally shaped using spatial or spectral shaping to reduce it's energy in areas where the mark would become visible [Lie et. al.]. An image adaptive watermarking scheme uses the local or global characteristics of the original image to determine the maximum strength that can be achieved in each area without introducing visible artifacts [Cappellini et. al., 1998]. Image-adaptive watermarking Algorithms have been proposed in [Hartung et. al., 1999, Podilchuk et. al., 1998, Cappelini et. al., 1998].

Watermarking techniques that do not require the original image for verification or extraction of the watermark are called "blind" watermarking as opposed to "informed" watermarking [Liu & Innoue, 2003, Cappellini et. al., 1998, Anderson & Petitcolas, 1998, Koch & Luo, 1998.].

The functions of the digital watermarking technology can be classified in four broad categories [Miller et. al.]:

a. Copyright Protection,
b. Monitoring,
c. Authentication, and
d. Secure and Invisible Communications.

Each individual application area desires its own set of special requirements with regards to robustness, fidelity and capacity [Lie et. al.].

In spite of the fact that digital watermarking has been an active area of research for decades, there is still a lot of room for improvements. One main reason for this is the limitations associated with each technique and the need to find the best balance between the three conflicting requirements (robustness, fidelity and capacity).

Robustness calls for the watermark to be as strong as possible where the fidelity requirement asks the watermark to be invisible.

It is difficult to satisfy all the requirements to their maximum at the same time. In current systems image watermarks are typically a pseudo-random signal with much lower amplitude, compared to the original image amplitude and usually with distribution of each bit into a group of pixels [Wolfgang et. al]. The pseudorandom signal is generally generated with Gaussian, uniform or bipolar probability density distribution using a secret seed.

Watermarks could also be a string of bits or a pseudo-randomly generated set of real numbers or a small image such as a company logo.

These watermarks however, often carry no extra information and are not very useful. On the other hand, multi-bit watermarks typically include a second signal used as error correction and thus decrease the amount of useful information or the payload that can be embedded.

Below are some Watermarking applications.

2.1 Watermark applications

Copyright Protection: Content providers such as individual artists or large-scale broadcast companies are interested in enforcing copyright protection of digital media [Koch & Luo, 1998, Berghel, 1997]. Authors wish to be ensured that their products are not commercially used without the payment of royalties. Another branch of this technology is fingerprinting. A product is marked with a unique label or fingerprint and then distributed to the rightful customer. Fingerprinting and Copyright applications require a high degree of robustness, and should be imperceptible but may have low capacity.

2.2 Monitoring

Digital watermarks can also be used to track and monitor digital content. In medical applications, watermarks might be used for identification and accessing of individual patient records. This particular application may prevent human errors such as record mismatching therefore preventing fatal mistakes [Koch & Luo, 1998]. In broadcast monitoring, companies like to confirm that their advertisements receive the full amount of airtime purchased. They have a desire to ensure that their product is broadcasted with the full duration, at the most optimal time of the day, and at preferred strategic frequencies [Cox, 2008]. Also, companies may wish to monitor the advertisement of the competition to predict future business strategies or explore competitive marketing techniques.

2.3 Authentication

For proof of authentication watermarks can be used not only to identify if a digital file has been tampered with, but also to determine how it has been tampered with. Such information can possibly give clues on how to reverse the malicious tampering to recover the original data. Authentication of surveillance cameras can be of importance if authorities question the reliability of such evidence in courts [Koch & Luo, 1998].

2.4 Communication

The idea of covert or secret communication is as old as communication itself [Hartung & Kutter, 1999] and is used frequently by defence and intelligence sectors. Digital watermarking continue to exist even after the receiver has obtained the information. If sensitive data is leaked out to unauthorized personal, the digital watermark contained in them can be used to trace back to the original owner or the intended receiver [Koch & Luo, 1998].

Digital watermarking used as covert communication adds an extra level of security compared to cryptography. In cryptography, the data is encrypted and can only be decrypted using a secret key. However, the attacker is aware of the existence of such data and can be certain that with enough time, he can decrypt the data, where as in digital watermarking, the attacker can never be certain that secret information is being transmitted.

Another advantage of digital watermarks is that it continues to exist even after the receiver obtains the information. Digital watermarking combined with cryptography is highly desired.

In this Chapter we will describe a watermarking algorithm for digital images for the purpose of copyright protection.

3. The watermarking process

In general any watermarking system consists of three components Watermark generation stage, encoding and decoding [Bartolini et. al., 2001].

3.1 Watermark generation

The watermark signal is typically a pseudo-random signal with much lower amplitude, compared to that of the original image and usually with distribution of each bit into a group of pixels [Hartung & Kutter, 1999]. The pseudo-random signal is generally generated with Gaussian, uniform or bipolar probability density distribution using a secret seed. Watermarks could also be a string of bits or a pseudo-randomly generated set of real numbers or a small image such as a logo.

3.2 Watermark encoding

The general idea is to embed a unique mark into a digital image such that it cannot be perceived by the Human Visual System but can be extracted at a later time using the content owner's secret key to prove ownership. Figure 2 shows the general example of encoding and decoding of a 4096 bit mark into the image "Lena". The mark is a binary image that has been uniquely generated by the watermarking system.

The cover image is first transformed into a domain that facilitates data embedding. The watermark can be embedded or encoded generally by adding or multiplying the signal to the cover image's luminance channel, the colour channels or both. For increased security and invisibility a spread spectrum coding with combination of a shaping technique is applied. In spread spectrum coding the watermark signal is spread over another known signal and then added to the image. Shaping can be done by increasing and decreasing the watermark's energy in some areas to adapt (become less visible) to the original work. In the DCT-Block domain, coefficients are modified according to the watermark content either by re-quantization, substitution or modification to impose a relationship [Bartolini et. al., 2001], [Koch & Zhao, 1995]. A General Watermarking encoding is described in Figure 3.

Watermark decoding

In the extraction stage some watermarking techniques need the original host image for subtracting the watermarked images, such techniques are referred to as "Informed" or

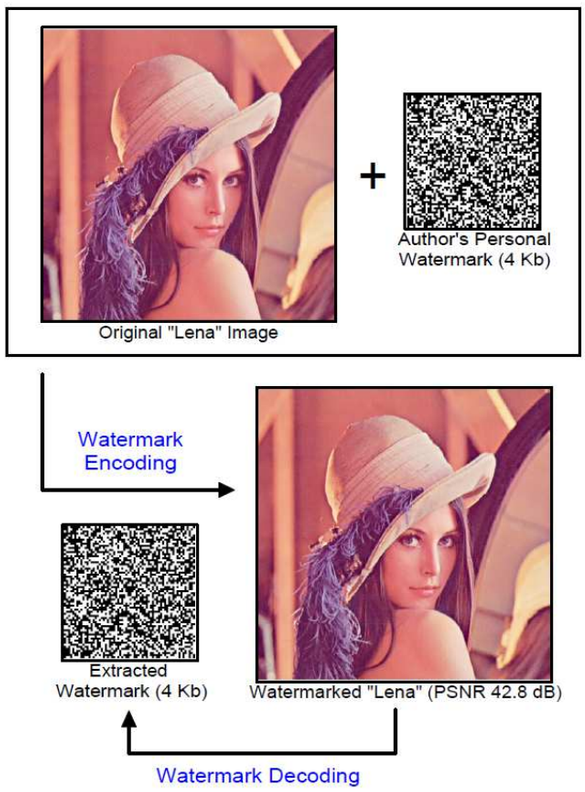


Fig. 2. General example of watermarking an image

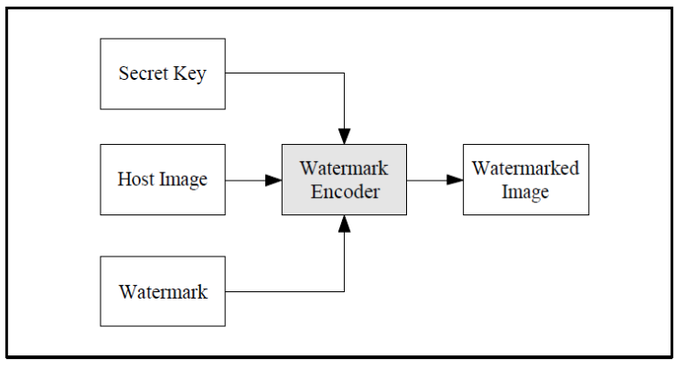


Fig. 3. General watermark encoding diagram

private watermarking [Miller et. al., 2002]. Other techniques do not need the original host image but need a secret seed to generate the original watermark for comparison. Such systems are referred to as "blind" watermarking. A watermarking system is "semi-blind" if it relies on some data or features derived from the original host image.

It is important to distinguish between watermark verification and watermark extraction. In most of literature the watermark is only verified, that is a correlation between the potential watermarked image and the original watermarked image is performed using the normalized correlation defined in Equation 1.

$$SIM(W, W^*) = \frac{W^* \cdot W}{\sqrt{W^* \cdot W^*}} \quad (1)$$

The output of a verification system is a yes/no answer. Extraction of the watermark is performed by reconstructing the watermark bit by bit from a potential watermarked image and comparing it with the original watermark. A threshold is defined for the percentage of similarity (Bit Error Rate) between the two. The basic process is depicted in Figure 4. An image marked with a watermark and a secret key are used by the watermark decoder to extract the original watermark signal.

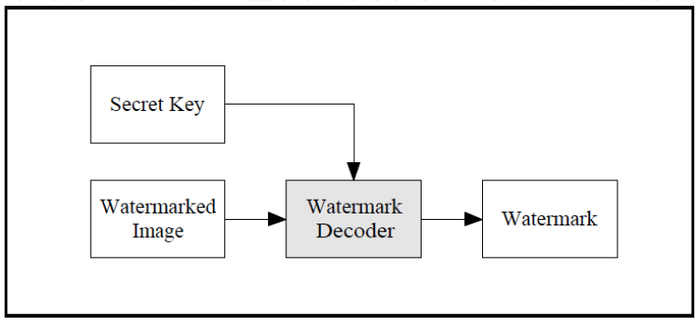


Fig. 4. General watermark decoding diagram

3.3 Watermark generation algorithm

Watermarks can take many shapes such as a company logo, image of a text or a pseudo-randomly generated sequence of bits or real numbers. We propose a new watermark with properties of self-correction. The Error Correction stage performs without any additional sources or reference marks. The author of the host image has the ability to specify personal information such as name, creation date, transaction ID or image ID as a human readable string of characters.

The provided information string is denoted as S, where S_i represents the ith characters in the string. First, each character S_i is converted to it's binary representation B_i and all B_i's are concatenated to form a sequence of bits denoted as B. For example, the binary representation of the string "Ben" is "01000010 01100101 01101110", where the spaces are only added for ease of visual distinction.

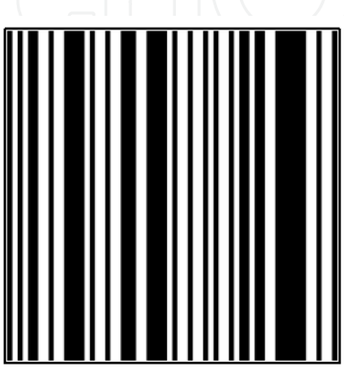


Fig. 5. Personal watermark of the string "Salami06" before encryption

In the next stage the sequence B is converted to a binary image, where a "0" represents a white pixel and a "1" represents a black pixel. The sequence is repeated vertically generating a barcode like image, illustrated in Figure 5. The mark uses a 64 bit information, duplicated 64 times, resulting in 4096 individual bits or 4 kilobytes. The vertical dimension of the mark depends on the height of the host image I , the larger the image dimensions the more the string can be repeated, thus increasing the robustness.

The dimensions of the image is determined by the Equation

$$W_h = \frac{I_h \times I_w}{\text{Strlen}(S) \times 8} \tag{2}$$

where W_h is the height of the watermark image, I_h and I_w are the dimensions of the host image and Strlen(S) is a function that returns the number of characters in the string S provided by the owner or author.

The watermark image is further encrypted using a user specified seed K_{mark} into a fast uniform pseudo-random number generator called "Mersenne Twister" with a period of $2^{19937} - 1$. The algorithm was developed by M. Matsumoto and T. Nishimura [Matsumoto & Nishimura, 1998] in 1998 and improved in 2002 [Matsumoto & Nishimura, 2002]. The generator is implemented to generate fast output by completely avoiding divisions and multiplications. It generates an array at one time and takes the full advantage of cache memory and pipeline processing if supported. Figure 6 depicts an example of an encrypted watermark

Fig. 6. Example of an encrypted watermark

Experts consider this an excellent random number generator. Using the seed Kmark, a sequence of N long-integer values ranging from 0 to $N - 1$ is generated where $N = W_h \times W_w$. The result is a 1-Dimensional array of pseudo-randomly generated values denoted as R, where R_i denotes the ith value in the list.

In order to encrypt the watermark the forward-scrambling Algorithm 1 is used, where each individual pixel W_i is exchanged with the corresponding pixel WR_i defined by R_i. The

Decryption method is very similar except that the shuffling is performed in the reverse order, for more details see Algorithm 2.

Algorithm 1 (Encrypt) *Encrypts the Watermark using Mersenne Twister*

ENCRYPT-MARK*(W, K_{mark}, N)*
1 R ← GENERATERANDOMS*(N,K_{mark})*
2 **for** *i ← 0 to N*
3 **do** *Temp ← W_i*
4 *W_i ← W_{Ri}*
5 *W_{Ri} ← Temp*
6 DELETE*(R)*

Algorithm 2 (Decrypt) *Decrypts the Watermark using Mersenne Twister*

DECRYPT-MARK*(W, K_{mark}, N)*
1 R ← GENERATERANDOMS*(N,K_{mark})*
2 **for** *i ← N − 1 to 0*
3 **do** *Temp ← W_i*
4 *W_i ← W_{Ri}*
5 *W_{Ri} ← Temp*

3.4 Watermark encoding algorithm

We embed the watermark into the DCT-Block domain of the host image. The DCT-Block has the advantage of revealing the local image characteristics [Cox & Li, 2005] and unlike using the full frame DCT, the watermark strength can be adapted to each local frequency content. This method proves to achieve maximum watermark fidelity [De Rosa et. al., 2000].

At first, the general encoding procedure is briefly described to allow the reader a broad conceptual view of the algorithm. Then in subsequent sections the algorithm is disassembled in individual components and each is further described in greater detail. The algorithm can be divided in three general stages. Image Preparation: The image is segmented into individual non-overlapping blocks, the colour space is converted from RGB to YCrCb (YUV) and each 8 × 8 block is transformed from the spatial to the frequency domain.

Watermark Encoding: The properties of the Human Visual System is explored and image adaptive strengths are determined for each block, the blocks are checked for potential edges before the pixels of the watermark image can be embedded. A testing mechanism ensures that the pixel was correctly embedded. Image Finalization: This stage is exactly the same as "Image Preparation" only in the reverse order. Each block is transformed back from the frequency to the spatial domain and the colour space is converted back from YCrCb to RGB. Lastly, all blocks are re-assembled to form the final watermarked image.

A pseudo-code of the encoding method is described in Algorithm 3 and for a more visual representation please refer to Figure 7.

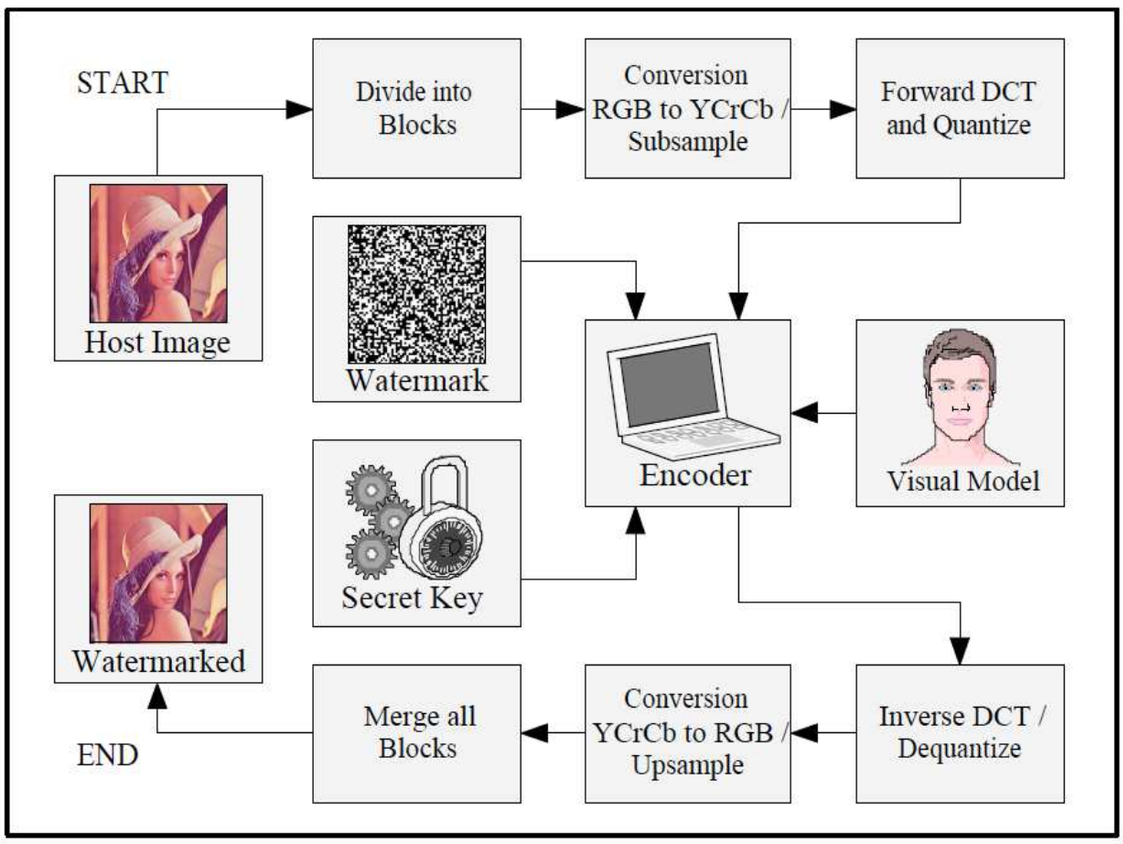


Fig. 7. Proposed watermark encoding diagram

Algorithm 3 (Encode) *Encodes the watermark W in image I*

ENCODE$(W, I, K_{image}, \alpha, Target, ET)$
1 PREPARE-IMAGE(I, N, Y, B, I^{crcb})
2 SHUFFLEBLOCKS(Y, K_{image})
3 CALCWATSONSSLACKS(Y, S)
4 ENCODE-WATERMARK$(N, Y, W, S, ET, \alpha, Target)$
5 UNSHUFFLEBLOCKS(Y, K_{image})
6 $I^W \leftarrow$ FINALIZE-IMAGE(Y, I^{crcb}, N, B)
7 **return** I^W

In Algorithm 3 *W* is the encrypted watermark and I is the original host image. K_{image} is used for shuffling of blocks, α is the user defined watermark strength, Target is a value that can be toggle between 0 and 255 to minimize the number of changes that the encoding algorithm must perform. ET is the edge threshold used in edge classification of blocks. The image I is first segmented via a call to Prepare-Image(*I*, *N*, *Y*, *B*, I^{crcb}) where *N* luminance blocks denoted as *Y* together with the chrominance components I^{crcb} are extracted.

Algorithm 4 (Prepare) *Prepares image I for encoding*

PREPARE-IMAGE(I, N, Y, B, I^{crcb})
1 $N \leftarrow 0$ and $B \leftarrow 0$
2 $I^{Ycrcb} \leftarrow$ NIL and $Y \leftarrow$ NIL
3 **for each** 16×16 *Block* **in** I
4 **do** CONVERT-COLORSPACE(I_B^{RGB}, I_B^{Ycrcb})
5 $I_B^{crcb} \leftarrow I_B^{Ycrcb} - I_B^Y$
6 $B \leftarrow B + 1$
7 **for each** 8×8 *Luminance Block* **in** I_B^Y
8 **do** FORWARDDCT(I^{DCT})
9 $Y^N \leftarrow$ QUANTIZE(I^{DCT})
10 $N \leftarrow N + 1$

The image I is first segmented into 16 × 16 non-overlapping *RGB* blocks, and for each the colout space is converted from *RGB* to *YCrCb* with a subsampling ratio of (4:1:1) obtaining I^{Ycrcb}. The chrominance I^{crcb} and luminance I^{Y} components are separated. Then the Block-DCT is applied to transform each 8 × 8 luminance block from the spatial domain to the frequency domain, followed by a lossy quantization step similar to JPEG compression.

The final quantized luminance blocks are saved in the set Y for the embedding procedure. In the next subsection and the discrete cosine transform (DCT) is described in more detail.

3.4.1 Discrete cosine transform / quantization

After the colour conversion, the luminance (Y) component is extracted and a 2- Dimensional Discrete Cosine Transform (DCT) is performed on every 8 × 8 (Y) block. The DCT is an invertible function that transforms the data from the spatial domain to the frequency domain and helps to separate the image into parts (spectral sub-bands) of differing importance with respect to the image's quality. The JPEG, MPEG-1, MPEG-2 and MPEG-7 encodings use the DCT domain to discard high frequency information that are not important to the human perception. The Forward DCT is defined in Equation 3 and it's inverse in Equation 3.3.

$$C(u,v) = \alpha(u,v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} S(x,y) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{3}$$

where $C(u, v)$ is the resulting DCT coefficient at the coordinates (u, v), $\alpha(u, v)$ is defined by Equation 5, S is the two dimensional square array of size $N \times N$ and in this case $N = 8$.

$$S(x,y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \alpha(u,v) C(u,v) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \tag{4}$$

where N, S and C are as described in Equation 3.

$$\alpha(u,v) = \begin{cases} 1/N & \text{for } u = 0 \text{ and } v = 0 \\ 2/N & \text{otherwise} \end{cases} \tag{5}$$

The first transform coefficient in the block is the average value of the sample so at location (0, 0) in the two dimensional 8 × 8 block the value for (u, v) is $1/N$. This value is referred to as the "DC" coefficient. All other transform coefficients are called the "AC" coefficients and have $\alpha(u, v)$ equal to $2/N$.

The lossy JPEG compression uses an 8×8 quantization matrix of step sizes (quantums), one element for each DCT coefficient, to further increase the compression ratio by discarding the high frequency coefficients. In this watermarking algorithm it is important to ensure that the coefficients used for embedding are not affected by JPEG's lossy-quantization step, therefore the embedding will occur after the lossy-quantization. Great care was taken not to affect the compression ratio. The luminance quantization matrix "Q" obtained from the (Independent Jpeg Group) IJG JPEG library is shown below.

$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Each DCT coefficient in the block is divided by the associated element in the quantization matrix and rounded to the nearest integer. The higher frequency coefficients which are located towards the lower part of the block are divided by higher values forcing them to become 0's. The lower frequencies (upper left) which are the perceptually significant part of the image are divided by smaller values, maintaining their accuracy. After quantization, usually more than half of the DCT coefficients are equal to zero. Therefore, it is impractical to modify any of the high frequency coefficients during watermark embedding for two main reasons. The first and most important reason is that JPEG compression will wipe out these values, destroying the watermark completely. The second reason is that it will disallow JPEG to perform an optimal compression using run-length coding of the zero coefficients.

3.4.2 Pixel encoding

The next step in the encoding process is the call to ShuffleBlocks(Y,Kimage) in which quantized luminance blocks are shuffled using a user defined key, very similar to the algorithm described in Algorithm 3.1. This step adds extra security to protect the watermark from intentional removal so that it will be very difficult for an attacker to guess or statistically show in which block which pixel of the encrypted watermark has been embedded.

Finally the encoding algorithms are given below. Algorithm 5 embeds the mark into the entire image and Algorithm 6 embeds on bit into one DCT block.

Algorithm 5 (Encode-Watermark) *Encodes W in Luminance component Y*

```
ENCODE-WATERMARK(N, Y, W, S, ET, α, Target)
1  for i ← 0 to N − 1
2      do GETRANDOMINDICES(C_X, C_Y, Y_i)
3         GETASSOCIATEDSLACKS(i, S_{C_X}, S_{C_Y})
4         δ ← min((S_{C_X} + S_{C_Y} + 1) × γ , α)
5         ID ← i
6         SETCOEFFICIENTS(C_X, C_Y, C_1, C_2, Target, W_{(i%W_Dim)}, ID)
7         ENCODE-INBLOCK(ID, Y_i, ET, C_1, C_2, δ)
```

Algorithm 6 (Encode-InBlock) *Encodes one DCT block β with strength*

```
ENCODE-INBLOCK(ID, β, ET, C_1, C_2, δ)
 1  if HASRELATIONSHIP(β_{C_1}, β_{C_2}, δ)
 2    then return
 3  if ISEDGEBLOCK(β, ET)
 4    then ADDTOBUCKET(ID) and return
 5    else repeat
 6                ForceRelationShip(β_{C_1}, β_{C_2}, δ)
 7                COPY(β, μ)
 8                IDCT(μ) and DEQUANTIZE(μ)
 9                FDCT(μ) and QUANTIZE(μ)
10                if |μ_{C_1}| > |μ_{C_2}|
11                  then Bit Embedding Is Confirmed
12                  else COPY(μ, β)
13           until Bit Embedded
```

In Algorithm 5, Y is the set of *N* luminance blocks, W is the encrypted watermark, *ET* is the threshold used for edge detection, α is the maximum watermark strength and Target is a value that can be toggled between 0 and 255 for minimizing the number of changes a single run of embedding creates.

3.5 Watermark decoding algorithm

The watermark decoding stage is very similar to the procedures described in the encoding except that now the original image and the original watermark are not available. The watermark bits are constructed bit by bit using the watermarked image. In order to extract the watermark successfully, several requirements must be met. One of the requirements is that the author's key file must be present. A key file is an encrypted binary file that has been written at the time of watermark encoding. This file contains the secret keys used by the author, the target value used for embedding, several templates such as image patches that facilitates in synchronization of the image in the spatial domain and several indices of rejected DCT blocks that can be included for a more robust decoding.

Furthermore, if the dimensions of the image have been altered, the image must be rescaled to the exact same dimensions as when it was marked. The algorithm will also need to be informed by a human user if a potential cropping has occurred, which in that case the synchronization templates provided in the key file will be matched against

the cropped image. If the matching has been successful, the remaining (uncropped) image is pasted against a black background in the exact same position that the templates suggest.

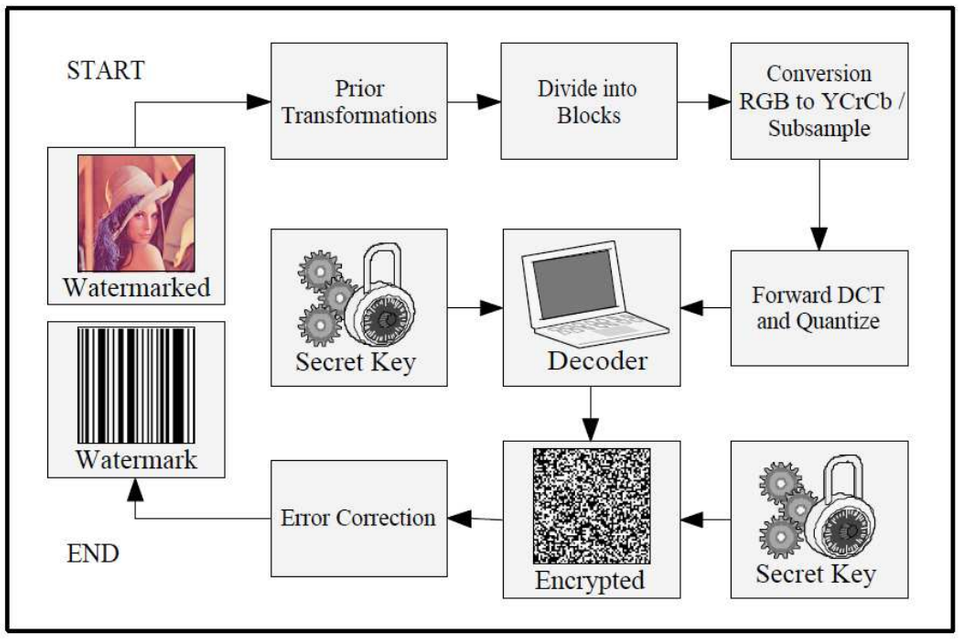


Fig. 8. Proposed watermark decoding diagram

The watermark is constructed pixel by pixel according to the relationship between specific DCT coefficients in a block. Before the relationships can be tested for, the image is segmented into blocks of 16 × 16 for a colour space conversion from RGB to YCrCb (YUV). Next, all blocks from the luminance (Y) component are extracted and a Forward DCT is performed on each 8 × 8 non-overlapping block using Equation 3.

The DCT blocks are shuffled using a secret key K_{image} obtained from the key file. Now the DCT blocks are ready for the watermarking extraction routine outlined in Algorithm 7.

The Target value is read from the key file and can either be 0 or 255, the value for AntiTarget is always calculated to be the opposite of the Target. The decoding procedure in Algorithm 7 traverses each block β in the luminance component Y, choosing the same two random coefficients C_1 and C_2 as used in the encoding procedure described in Algorithm 6. The structure Bucket read from is used to check if the current block has been previously discarded by the encoding algorithm, in which case the sign of the value in $Bucket_j$ will decide if the pixel W_i is equal to 255 or 0. After all blocks have been processed, the watermark W has to be decrypted to reveal the original Bar-code like image created by the author.

Algorithm 7 (Decode) *Extracts the embedded watermark from image I*

```
DECODE(Y, K_image, K_mark, W, Target, Bucket)
 1  Y ← SHUFFLEBLOCKS(Y, K_image)
 2  AntiTarget ← |Target − 255|
 3  j ← 0
 4  for i ← 0 to N − 1
 5      do if Bucket not empty and |Bucket_j| = i
 6            then if Bucket_j ≥ 0
 7                    then W_i ← Target
 8                    else W_i ← AntiTarget
 9                 j ← j + 1
10            else β ← Y_i
11                 GETRANDOMINDICES(C_1, C_2, β)
12                 if |β_C1| ≥ |β_C2|
13                    then W_i ← Target
14                    else W_i ← AntiTarget
15  W ← DECRYPTMARK(W, K_mark, N)
```

Watermarked images are usually posted on web sites (internet) or distributed to individual customers sometime after the encoding process. Between the time of encoding and the time of decoding the watermarked image may undergo many possible manipulations. Some of these attacks are intentional such as cropping and others are unintentional like the collection of channel noise. In addition, the lossy quantization step during JPEG compression and decompression is a major source for error in the decoding process. Therefore, the decoded watermark may not always appear 100% identical to the original embedded mark. One method of determining the similarity of two given signals is known as Bit Error Rate (BER). The BER is the ratio of the total bit error to the total number of bits embedded and is given by:

$$BER = \left(\sum_{I=0}^{N-1} B_I^* \oplus B_I \right) \; / \; N \tag{6}$$

where B_I^* and B_I are the bits at the i^{th} position of the decoded watermark and the original watermark respectively. The symbol $\oplus$ is the binary XOR operation and N is the total number of bits in B.

For a perfect decoding step the BER would equal to 0, indicating no lost bits. Since no algorithm can claim to be perfect it is important for every watermarking system to expect such bit errors and facilitate a method of Error Correction.

4. Experiments and results

This Section presents the results found using the methodology described in Section 3. One Thousand color images are used to test and obtain results from the proposed system "Digital Image Copyright Protector" (DIGI-COP). In Section 4.1 we explore the fidelity of the marked images, in Section 4.2 Error Rates are analyzed and in the subsequent sections various attacks are explored as listed below.

4.1 Fidelity

One of the major requirements of digital watermarking systems is the ability to hide the mark within the cover work such that it becomes perceptually invisible to a human observer. The proposed image watermarking method (DIGI-COP) presented in this Chapter makes use of local characteristics of the image to achieve higher invisibility rates than it's base algorithm (BWM). Consider an image I and it's watermarked version IW, then the standard deviation between I and IW is defined by the MSE (Mean Square Error):

$$\text{MSE}\ (I, I^W) = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I(i,j) - I^W(i,j)||^2 \tag{7}$$

Peak Signal-to-Noise Ratio (PSNR) is often used for global evaluation of the quality of reconstruction in image compression techniques. It is expressed in terms of the logarithmic decibel (dB) scale and defined as:

$$\text{PSNR}\ (I, I^W) = 10 \log_{10} \left(\frac{255^2}{MSE} \right) = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \text{dB} \tag{8}$$

Thus PSNR is the ratio between the maximum possible power of a signal and power of corrupting noise that affects the fidelity of its representation. The original image and the watermarked image are denoted by I and by I^W respectively, and M and N are the dimensions of the images. A lower value of MSE means less distortion, and therefore the higher the PSNR value is, the better or closer the watermarked image is to the original. Generally a PSNR larger than 32 dB means invisible visual degradation and a human observer perceives both images as indistinguishable [Wilson & Martinez 97].

For the fidelity test 1000 RGB images were marked with various watermark strengths, $\alpha \in \{10, 20, 30, 40\}$, using both the base method (BWM) and the proposed watermarking method (DIGI-COP). The PSNR values obtained are graphically produced in Figure 9 and the average PSNR values together with the standard deviations σ are presented in Table 1.

α (strength)		Digi-Cop			BWM	
		PSNR (dB)	σ		PSNR (dB)	σ
10		41.76	2.37		39.97	2.75
20		41.24	2.07		38.71	2.22
30		40.72	1.92		37.39	1.75
40		40.35	1.83		36.12	1.38

Table 1. Average PSNR of 1000 watermarked images DIGI-COP BWM

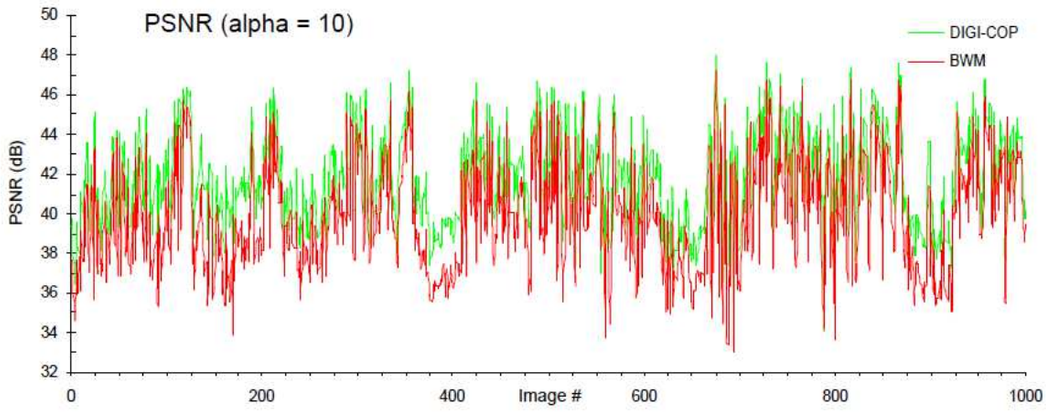


Fig. 9. PSNR values of 1000 watermarked images with α = 10

It is clear from the results that DIGI-COP achieves higher PSNR values than the BWM even when the watermark strength (α) is increased. Next, the PSNR results for classical images marked with α = 40 are illustrated in Table 2 and Figure 10. The PSNR values are calculated in the YCrCb domain and reflects the similarities in the luminance components.

Image	DIGI-COP PSNR (dB)	BWM PSNR (dB)
Boys	42.56	37.67
Boat	39.11	35.73
Peppers	40.31	36.67
Baboon	36.78	34.60
Plane	40.48	36.05
Tiffany	40.99	37.15
Drop	42.12	37.35
Lena	41.30	37.09

Table 2. PSNR values of classical images with α = 40

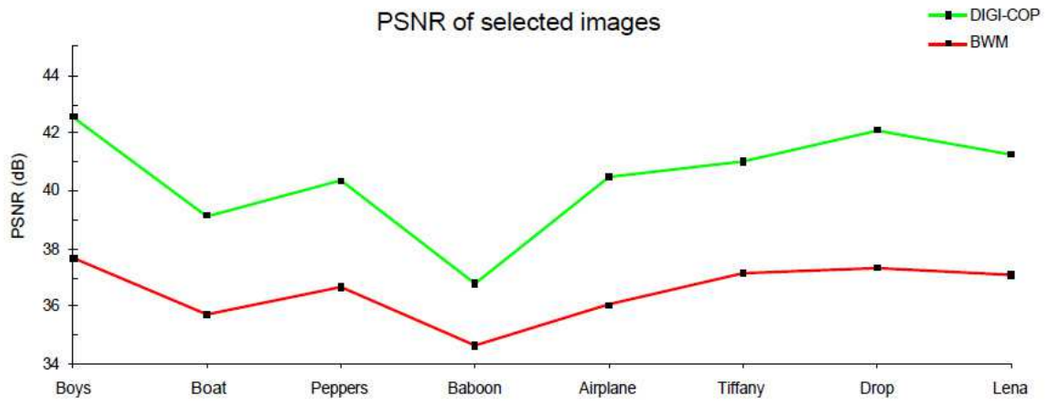


Fig. 10. PSNR values of selected watermarked images with α = 40

In literature [Meerwald 2001, Jellinke 2000, Mohanty 1999, Guo 2003], it has been previously reported that the base watermarking method (BWM) occasionally produces small spatial defects around edges and an undesired blocking effect in smooth regions of the image. Figure 11 demonstrates that DIGI-COP protects the image from such unwanted artifacts.

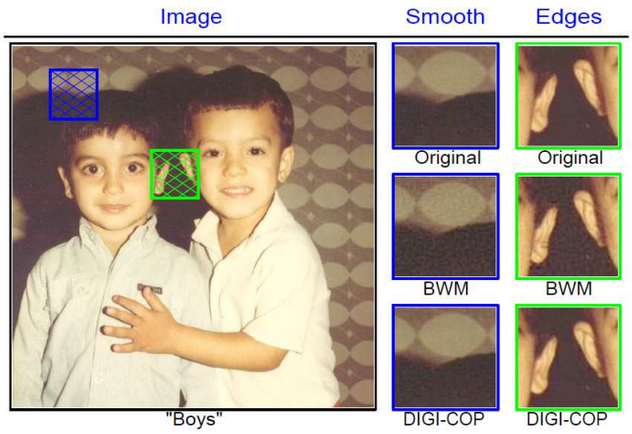


Fig. 11. Smooth and edge regions of the image "Boys" after watermarking

In Figure 11 the shaded blue and green regions in the image represent the selected locations of smooth and edge regions. The enlarged views of the smooth and edge region are presented towards the right of the image and compared to the original unwatermarked areas.

4.2 Error rates

There are two types of errors that can occur during the watermark extraction stage. The first error is a false-negative (FN), in which a watermark decoder fails to identify a watermarked image as a legit or "marked" copy. The decoder's ability of correctly extracting the watermark W from a marked image I is calculated in terms of a BER (Bit Error Rate) value described in Equation 6 on Page 17. The FN test was performed on 1000 watermarked images with α of 20 using the decoder of BWM and DIGI-COP. The BER values are illustrated in Figure 12.

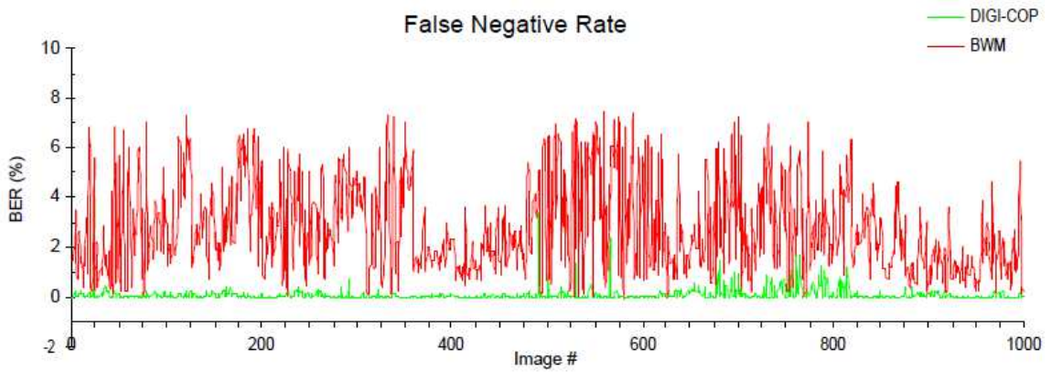


Fig. 12. False negative rate

The graph indicates that DIGI-COP achieves a lower BER in the decoding process and extracts the watermark with higher precision.

The second type of a decoding error is a false-positive, where the watermark decoder incorrectly detects the presence of a watermark in an image. There are two types of false-positives, the first type (FP-I) occurs when a watermark decoder extracts a watermark in an image I that has not been marked previously. The second type (FP-II) of false positives is when a decoder extracts watermark W from an image I that has been marked with a different mark W . Both types of false-positives are undesired in a reliable system. The results for both types of false-positives tests are presented in Figure 4.9.

In Figure 13, the FP-I results suggests that the BWM has a higher accuracy in determining unwatermarked or incorrectly marked images. On the other hand, FP-II results show clearly that both watermark decoders can correctly extract the watermark W from an image marked with W placed at location 500. Further, BWM and DIGI-COP show high BER values when attempting to extract watermark W from images marked with different watermarks such as W*.

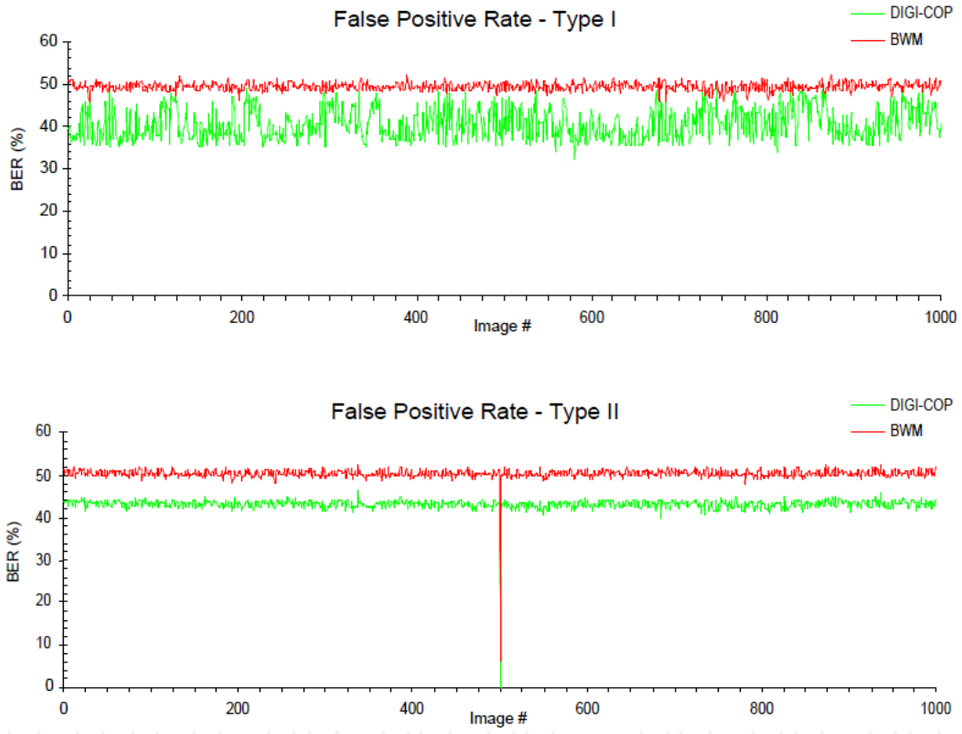


Fig. 13. False positive rates types I and II

4.3 Image processing attacks

4.3.1 JPEG compression

JPEG compression is one of the main reasons for the success of the internet and must be taken into account when designing an image watermarking system. JPEG compression will attempt to remove the perceptual unimportant elements from an image and may render the imperceptible watermark undetectable. Image compressions are considered one of the strongest enemies of digital watermarking techniques today.

The JPEG compression resistance test presented in this thesis is performed on 1000 color images over 11 different JPEG quality factors ranging from 100% to 0%. The BER value is

calculated for each individual image and averaged over the entire image set for that particular quality as illustrated in Table 3.

	DIGI-COP		BWM	
JPEG Quality	BER (%)	σ	BER (%)	σ
100	0.29	1.61	3.74	4.25
90	2.53	1.78	10.38	4.78
80	3.81	2.14	10.86	4.97
70	6.91	3.49	17.05	6.71
60	9.30	4.82	17.09	6.64
50	11.36	5.57	14.32	5.88
40	14.25	6.28	11.70	4.57
30	19.88	7.37	18.63	5.50
20	26.85	7.69	29.29	5.88
10	33.14	6.66	38.41	3.90
0	37.84	4.41	44.44	0.83

Table 3. Effects of JPEG compression on BER

Both the BWM and DIGI-COP decoders show high resilience against this attack up to a JPEG quality factor of 40. In addition, an Error Correction (EC) technique is used to achieve a much lower BER value than the original proposed method in it's essence. The illustration in Figure 14 is the direct result of a total of 50,000 individual decoding operations and shows the advantage of the Error Correction technique.

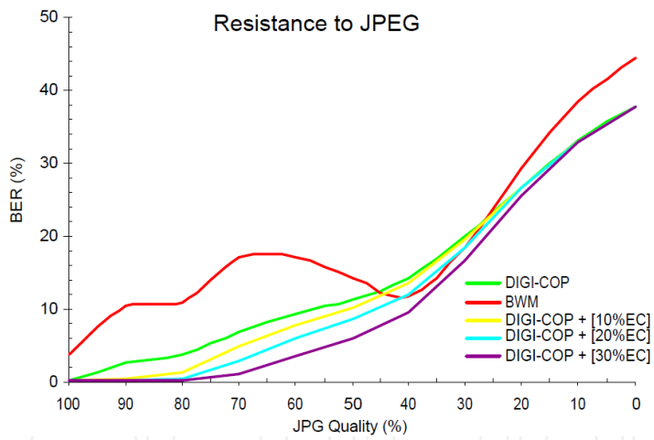


Fig. 14. JPG Compression with Error Correction (EC)

The Error Correction was set to 10%, 20% and 30% to see the changing effects of the BER value over the entire image set. In Figure 4.10 the BER value is plotted against the various JPEG qualities, and it can be seen that as images are compressed at higher rates (lower qualities), the Bit Error Rate also increases. However, DIGICOP's Error Correction feature significantly reduced the decoding BER.

5. Conclusion and future research

A new adaptive and invisible digital watermarking system ("DIGICOP") in the DCT-Block domain is discussed as a method for protecting copyright for digital images. The performances of DIGI-COP and the classical DCT-Block technique are compared.

Extensive results show that DIGI-COP is preferred over the classical method in terms of it's fidelity and robustness. The method can embed large quantities of data into the cover image without any noticeable changes. The new embedding technique facilitates embedding the right amount of watermark at the most advantageous locations in the image without causing visual artifacts. The improvement is achieved by exploiting the characteristics of the cover image in the DCT-Block domain, as well as the sensitivity of the HVS to small changes in smooth regions and edges of the image. The fidelity test of DIGI-COP achieved on average 41 dB over one thousand encodings where the classical method achieves on average 38 dB on the same set of images.

In addition, both false-negative and false-positive rates of the two algorithms were compared over a set of thousand images. DIGI-COP's false-negative rates show to be more reliable than the classical algorithm. It correctly extracts 99.9% of the data with a standard deviation of 0.26 as compared to the classical method with 97.2% decoding rate and a deviation of 1.8. False-positive rates were compared and both algorithms show good performance. Although both algorithms can identify a legit watermark from a set of thousand randomly marked images and deny all unmarked or incorrectly marked images, the classical algorithm shows a slight better performance. This is due to DIGI-COP's feature of storing discarded bits in the key file and assuming it's presence in unmarked or incorrectly marked images.

6. References

Anderson, R. J. & Petitcolas, F. On The Limits Of Steganography. (1998). IEEE J. Select. Areas Communication. (Special Issue on Copyright and Privacy Protection), 16, 474–481, May 1998.

Baaziz, N. (2005). Adaptive Watermarking Schemes Based On A Redundant Contourlet Transform. IEEE International Conference on Image Processing ICIP 2005, 1, , 11–14 September 2005, pp 221–224.

Bartolini, F.; Barni, M.; Podilchuk, C. I. & Delp, E. J. Watermark Embedding: Hiding A Signal Within A Cover Image. IEEE Communications Magazine, 39, August 2001. pp 102–108,

Berghel. (1997). Watermarking Cyberspace. Communications of the ACM, 40, 1997, pp 19–24.

Cappellini, V.; Barni, M.; Bartolini F.; & Piva, A. (1998). A DCT Domain System For Robust Image Watermarking, Signal Processing (Special Issue on Watermarking), 66, May 1998. pp 357–372

Cappellini, V.; Piva, A.; Barni, M.; Bartolini F. & Rigacci, F. (1998). A M.A.P. Identification Criterion For DCT-Based Watermarking, Proceedings Europe: Signal Processing Conference (EUSIPCOŠ98), September 1998.

Cheng, L.L.; Ng, K.S.; Cheng L. M. & Wong, M.K. (1999). Adaptive Watermarking By Using Pixel Position Shifting Technique IEEE Transactions on Consumer Electronics, 45, November 1999. pp 1057–1064.

Cox, I. J. Digital Watermarking and Steganography. Morgan Kaufmann, 2008.

Cox and Q. Li. Using Perceptual Models To Improve Fidelity And Provide Invariance To Value-Metric Scaling For Quantization Index Modulation Watermarking. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing 2005' (ICASSP), March 18-23 2005.

De Rosa M. Barni, F. Bartolini and A. Piva. Capacity of Full Frame DCT Image Watermarks. In IEEE Transactions on Image Processing, vol. 9, August 2000, pp 1450–1455.

Guo, H. Digital Image Watermarking for Ownership verifciation. PhD thesis, University of Ottawa, 2003.

Hartung, F. & M. Kutter, Multimedia Watermarking Techniques. (1999). Proceedings of the 1999 IEEE, 87, July 1999, pp 1079–1107.

Huang, C.-H. and Wu, J.-L. Attacking visible watermarking schemes. In IEEE Transactions on Multimedia, volume 6, pages 16–30, February 2004.

Jellinek, B. Invisible watermarking of digital images for copyright protection. Master's Thesis, University of Salzburg, January 2000.

Koch, E. & Zhao, J. (1995) Towards Robust And Hidden Image Copyright Labeling. Proceedings of 1995 IEEE Workshop on Nonlinear Signal and Image Processing, pages 452–455. Halkidiki, Greece, June 1995.

Koch E.; Zhao, J. & Luo, C. (1998). In business today and tomorrow Communications of the ACM, 41, July 1998. pp 66–72.

Leighton, F. T.; Cox, I. J.; Kilian, J. & Shamoon, T. (1997). Secure Spread Spectrum Watermarking For Multimedia. IEEE Transactions Image Processing, v6, December 1997. pp 1673–1687.

Lie, W. & Chang, L. (1997). Robust And High-Quality Time-Domain Audio Watermarking Based On Low-Frequency Amplitude Modification, IEEE Transactions on Multimedia, vol 8, February 2006. pp 46 – 59.

Liu, Z. & Inoue, A. (2003). Audio Watermarking Techniques Using Sinusoidal Patterns Based On Pseudorandom Sequences. IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, August 2003, pp 801–812.

Matsumoto, M. & Nishimura, T. (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. ACM Transactions on Modeling and Computer Simulation, vol 8, January 1998, pp 3–30.

Matsumoto, M. & Nishimura, T. (2002). A nonempirical test on the weight of pseudorandom number generators. Monte Carlo and Quasi-Monte Carlo methods, 2002, pp 381–395.

Meerwald, P. Digital image watermarking in the wavelet transform domain. Master's Thesis, University of Salzburg, January 2001.

Miller, M. L.; Cox, I. J. & J. A. Bloom. (2002). Digital Watermarking. Academic Press, 2002.

Mohanty, S. P. Watermarking of digital images. Master's Thesis, University of Salzburg, January 1999.

Podilchuk, C. I.; & Zeng, W. (1998) Image-Adaptive Watermarking Using Visual Models, IEEE Journal on Selected Areas in Communications, 16, May 1998. pp 525–539.

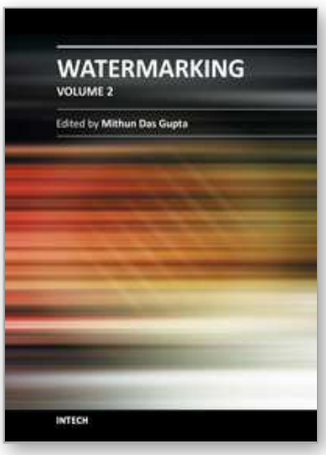
Puate, J. and Jordan, F. "Using fractal compression scheme to embed a digital signature into image," in Proc. SPIE Photonics East Symp., Boston, MA, Nov. 18–22, 1996. Available http://iswww.epfl.ch/~jordan/watermarking.html.

Ren-Hou, L.; Lian-Shan, L. & Qi, G. (2005). A Robust Video Watermarking Scheme Based On DCT. Proceedings of 2005 International Conference on Machine Learning and Cybernetics, 8, 5176-5180, August 2005. pp 18–21.

Shahraeini, S. and Yaghoobi, M. A Robust Digital Image Watermarking Approach against JPEG Compression Attack Based on Hybrid Fractal-Wavelet. Advanced Materials Research, 403-408, 2012.

Wilson, D. R. and Martinez, T. R. Improved heterogeneous distance functions. In Journal of Artificial Intelligence Research., 1997.

Wolfgang, R. B.; Podilchuk, C. I. & Delp, E. J. (1999). Perceptual Watermarks for Digital Images and Video, vol 87:7, July 1999, pp 1108–1126.



Watermarking - Volume 2

Edited by Dr. Mithun Das Gupta

ISBN 978-953-51-0619-7
Hard cover, 276 pages
Publisher InTech
Published online 16, May, 2012
Published in print edition May, 2012

This collection of books brings some of the latest developments in the field of watermarking. Researchers from varied background and expertise propose a remarkable collection of chapters to render this work an important piece of scientific research. The chapters deal with a gamut of fields where watermarking can be used to encode copyright information. The work also presents a wide array of algorithms ranging from intelligent bit replacement to more traditional methods like ICA. The current work is split into two books. Book one is more traditional in its approach dealing mostly with image watermarking applications. Book two deals with audio watermarking and describes an array of chapters on performance analysis of algorithms.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Charlie Obimbo and Behzad Salami (2012). Using Digital Watermarking for Copyright Protection, Watermarking - Volume 2, Dr. Mithun Das Gupta (Ed.), ISBN: 978-953-51-0619-7, InTech, Available from: http://www.intechopen.com/books/watermarking-volume-2/using-digital-watermarking-for-copyright-protection

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen