Digital Watermarking Robust Against JPEG Compression

Hye-Joo Lee[1], Ji-Hwan Park[1], and Yuliang Zheng[2]

[1] Department of Computer Science,
PuKyong National University, the Republic of Korea
leehj@woongbi.pknu.ac.kr,
jhpark@dolphin.pknu.ac.rk
[2] School of Comp. & Info. Tech., Monash University, Australia
yuliang@pscit.monash.edu.au

Abstract. Digital watermarking has been considered as an important technique to protect the copyright of digital content. For a digital watermarking method to be effective, it is essential that a watermark embedded in a still or moving image resists against various attacks ranging from compression, filtering to cropping. As JPEG is a dominant still image compression standard for Internet applications, digital watermarking methods that are robust against the JPEG compression are especially useful. Most digital watermarking methods proposed so far work by modulating pixels/coefficients without considering the quality level of JPEG, which renders watermarks readily removable. In this paper, we propose a new method that actively uses the JPEG quality level as a parameter in embedding a watermark into an image. A useful feature of the new method is that the watermark can be extracted even when the image is compressed using JPEG.

1 Introduction

The tremendous development in data compression methods has resulted in the widespread use of digital data such as image, audio and video in every corner of our daily life. Digital data are easy to distribute and duplicate. This gives rise to a serious problem in illegal copying. While encryption is essential to the provision of confidentiality, the same technology does not represent an ideal solution to copyright protection, simply because any user who possesses a decryption key may (re-)distribute decrypted digital images as he or she wishes. This indicates the necessity of embedding information on the rightful owner into an image in such a way that the information and the image cannot be easily separated. To achieve this goal, researchers have proposed to use so-called digital watermarking [1, 2]. The most important requirement of digital watermarking is that embedded watermarks are robust against compression, filtering, cropping, geometric transformation and other attacks. For images that are published on the World-Wide Web (WWW), robustness of watermarks against the JPEG compression standard is particularly important.

In this paper, we propose a method that constructs a watermark by using the quality level of JPEG as a parameter. As a result we obtain a watermarking method that is robust against the JPEG compression. We describe an overview of digital watermarking in Section 2. This is followed by a detailed description of our proposed method in Section 3. A number of simulation results are provided in Section 4 to verify the effectiveness of the proposed method.

2 Digital Watermarking Method

Digital watermarking consists of a pair of matching procedures, one for embedding a watermark into a still or moving image and the other for detecting/extracting the watermark. A number of factors have to be considered while embedding a watermark. These factors include the structure of a watermark, locations where the watermark is embedded, and the level of change in the quality of the image introduced by digital watermarking.

The structure of a watermark generally falls into one of two types. The first type is essentially a random binary sequence which is composed of either 0 and 1 or -1 and 1 [3–6]. The second type is a random real number that is distributed according to N(0,1)[7]. With a random binary sequence, one can apply the sequence in the extraction of an embedded watermark as well as the detection of the presence of the watermark. A disadvantage of the use of a random binary sequence is that the watermark is vulnerable to such attacks as removal and collusion. In comparison, with a random real number, one cannot extract the original image. Nevertheless, this method has the advantage of being more robust against removal and collusion attacks, primarily due to the fact that even though an attacker may have some knowledge on the locations of a watermark, he or she has far greater difficulties in identifying the exact watermark. Random real numbers are being used by researchers more often than random binary numbers recently.

Locations for embedding watermark are often determined by using random sequences together with human visual system(HVS)[8, 9]. For instance, the Podilchuk-Zeng method[8] have utilized the visual model developed by Watson[10] for the JPEG compression. More specifically, the authors have used the frequency sensitivity portion of the model to embed a watermark. In addition, Dittmann[9] et al have developed a robust video watermarking method based on a combination of an error correction code, the Zhao-Koch method[11] in the frequency-domain and the Fridrich method[12] in the spatial-domain. The method proposed in this paper is similar to the above two methods in the use of the HVS. However, as our method embeds copyright information in the spatial domain, it differs from the Podilchuck-Zeng method that embeds in the frequency-domain using DCT or wavelet transform. Compared with the method by Dittmann et al, their method doesn't consider the re-compression of a watermarked video image.

With a method entirely relying on a secure random sequence to determine locations for embedding a watermark, the quality of a watermarked image de-

grades since it does not fully consider the feature of the image. In contrast, by using HVS in deciding locations, it is possible to make the change of image perceptually unrecognizable. Nevertheless an attacker who is familiar with HVS may still be able to estimate the location of the watermark. For this reason, the use of HVS actually decreases the secrecy with respect to the watermark's locations even though it meets the requirement of invisibility.

In view of these observations, we argue that digital watermarking should satisfy the following conditions:

1) If the structure of a watermark is a binary random sequence, the magnitude of the watermark has to be different in all pixels.
2) If locations to embed the watermark are random, the magnitude of the watermark has to be dependent on the image.
3) If it employs HVS, locations to embed the watermark have to be random.

In the following section, we describe our proposed method that converts a text on copyright information into binary random sequences. To satisfy the above conditions, the embedding locations and changes in pixels are determined by using the HVS.

3 Digital Watermarking Using Difference

The JPEG algorithm is one of the loss compression methods for still images. We consider a situation where a watermarked image will be compressed using the JPEG prior to its publication on WWW. Figure 1 illustrates the block diagram of the JPEG algorithm.

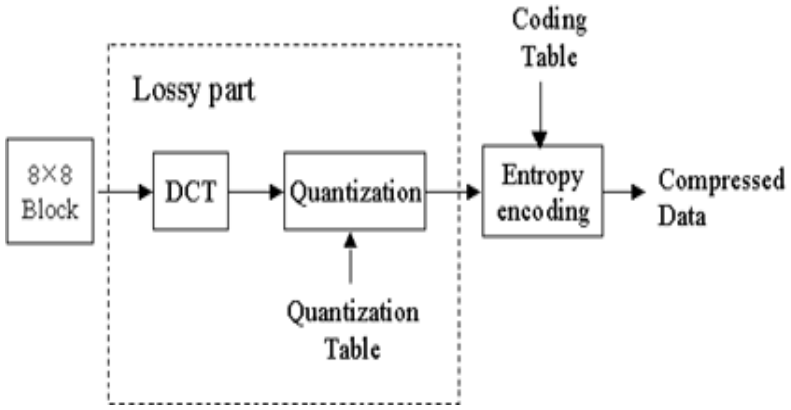

Fig. 1. The JPEG compression

As illustrated in Figure 1, the loss of information is due to quantization. When the JPEG algorithm is performed, *the quality level* is used to construct a quantization table. In general, the higher the quality level, the lower the compression ratio. Now consider a case where users uses only JPEG for compression. When the owner of an image expects to achieve robustness at least for the quality level q_1, he or she can construct a watermark, and embeds it into the original image by using q_1 as a parameter. We will show how this can be done. In addition we will show that it is possible for the owner to extract the watermark from the watermarked image when the watermarked image is compressed with JPEG at a higher quality level $q_2 (> q_1)$.

3.1 Preprocessing

To start the method we construct a watermark as follows. We calculate the differences between the original image and a reconstructed image after compression with q_1, and the visual component from the original. Each parameter is defined as follows.

i, j : denote positions of pixels on the original image of size $N \times M$, $0 \leq i < N$, $0 \leq j < M$
h, v : denote indexes of a 8×8 block which is partitioned from the original image, $0 \leq h < N/8$, $0 \leq v < M/8$
k, l : denote positions within one block, $0 \leq k < 8$, $0 \leq l < 8$.

First, an image I of size $N \times M$ is partitioned into 8×8 blocks. Subsequently, DCT transform is applied to each block and all DCT coefficients are quantized by quantization factors obtained from q_1. Note that the quality level q_1 acts also as an indicator of robustness against the JPEG compression. For the reconstructed image, the difference $D_{h,v}$ of all blocks is calculated using Eqn.(1).

$$D_{h,v} = \{d_{k,l} | x_{k,l} - x'_{k,l}\}, \tag{1}$$

where $x_{k,l}$ and $x'_{k,l}$ denote the value of a pixel in the original image and the value of a reconstructed pixel with respect to q_1. The difference $d_{k,l}$ is used to decide the magnitude of a watermark to be embedded, and this magnitude is then used to obtain the watermark patterns.

The next step in preprocessing is to compute visual components that are dependent on the image. Centering the pixel value $x_{i,j}$ within a $t \times t$ window, the average difference of brightness $\hat{x}_{i,j}$ between the centering pixel and its neighboring pixels is calculated using Eqn.(2).

$$\hat{x}_{i,j} = \frac{1}{t^2 - 1} \left\{ \sum_{s_1=i-t}^{i+t} \sum_{s_2=j-t}^{j+t} |x_{s_1,s_2} - x_{i,j}| \right\} \tag{2}$$

The average brightness $b_{i,j}$ is then calculated for the neighboring pixels using Eqn.(3)

$$b_{i,j} = \frac{1}{t^2 - 1} \left\{ \sum_{s_1=i-t}^{i+t} \sum_{s_2=j-t}^{j+t} x_{s_1,s_2} - x_{i,j} \right\}. \tag{3}$$

From the average brightness $b_{i,j}$, a relative intensity R is obtained via Eqn.(4).

$$R = \left[\frac{x_{i,j}}{b_{i,j}} \right] = [r_{i,j}] \tag{4}$$

It is difficult to perceive the change by human eyes, as the change takes place in the vicinity of edge. Therefore, we can utilize the following as visual components.

$$V_{h,v} = \{v_{k,l} | \log(r_{k,l} \cdot \hat{x}_{k,l})\}. \tag{5}$$

As shown in Eqn.(5), the visual components are obtained by multiplying $\hat{x}_{i,j}$ by $r_{i,j}$ and scaling the resulting value with log. The values $d_{k,l}$ and $v_{k,l}$, which are dependent on the image, are used to produce a watermark that satisfies the conditions 1) and 2) as mentioned earlier.

3.2 Construction of a Watermark

Before locations to embed a watermark are determined, we have to compute the magnitude of the watermark from the values of $d_{k,l}$ and $v_{k,l}$. The values of the watermark are restricted by the difference value $d_{k,l}$ through a modular operation. A pattern $U_{h,v}$ can be obtained from the product of $D_{h,v}$ and $V_{h,v}$. Note that the pattern is actually a matrix of size 8×8 (see Eqn.(6)).

$$U_{h,v} = (D_{h,v} \times V_{h,v}) \bmod D_{h,v}, \tag{6}$$

It should be pointed out that the mod operation is applied to each element of the matrix. The elements $u_{k,l}$ of the matrix $U_{h,v}$ represent the amount of change in a corresponding pixel. Now the watermark $f_{i,j}$ can be defined as follows.

$$f_{i,j} = \begin{cases} 1, \, u_{i,j} \neq 0 \text{ and } d_{i,j} \neq 0 \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

The watermark is embedded into a location where the value of $f_{i,j}$ is one, that is, where the magnitude of the watermark is none zero.

Let $ID = \{id_0, id_1, \cdots, id_{l-1}\}$ be a text that indicates the copyright information. In the following Eqn.(8), C_{ID} is obtained by repeating ID a number of times. The repetition is necessary in order to reduce/eliminate errors during extraction.

$$C_{ID} = \underbrace{ID||ID||\cdots||ID}_{L}, \tag{8}$$

Here L is the number of locations where the value of $f_{i,j}$ is one. We note that clearly repetition can be replaced with a more efficient error correction code.

C_{ID} is then randomized with $m_{i,j}$, an M-sequence with a maximum period. This ensures that the condition 3) discussed above is satisfied. We denote the randomized sequences by $s_{i,j}$ which are defined more precisely as follows:

$$s_{i,j} = \begin{cases} id_y \oplus m_{i,j}, \text{ if } f_{i,j} = 1 \\ m_{i,j}, \qquad \text{otherwise} \end{cases} \tag{9}$$

where id_y is an element in C_{ID} .

Finally a watermark pattern $w_{i,j}$ is constructed via Eqn.(10) and it is embedded into the original image. The watermarked image I_W is obtained by the addition of the pattern $w_{i,j}$ to a corresponding pixel in the original image.

$$w_{i,j} = \begin{cases} (2 \times s_{i,j} - 1) \cdot u_{i,j}, \text{ if } f_{i,j} = 1 \\ 0, \qquad\qquad\quad \text{otherwise} \end{cases} \tag{10}$$

To extract the watermark, we subtract the original image from the watermarked image. Then, a value $\hat{s}_{i,j}$ is derived from the signs of $u_{i,j}$ and $\hat{w}_{i,j}$ of Table 1.

Table 1. The extraction of $\hat{s}_{i,j}$

	$\hat{w}_{i,j} > 0$	$\hat{w}_{i,j} < 0$
$u_{i,j} > 0$	1	0
$u_{i,j} < 0$	0	1

To reconstruct the copyright information ID , $\hat{id}_y$ is extracted using $m_{i,j}$ and $\hat{s}_{i,j}$ as in Eqn.(11).

$$\hat{id}_y = \hat{s}_{i,j} \oplus m_{i,j}, \text{if } f_{i,j} = 1, \tag{11}$$

If $\hat{s}_{i,j}$ is equal to $s_{i,j}$ used in embedding, the extracted $\hat{id}_y$ should be the same as the original id_y. As a result, we are able to reconstruct the copyright information ID correctly.

4 Simulation Results

We simulated the proposed method to evaluate the robustness of the following block data which serves as a simple example.

$$\begin{pmatrix} 20 & 20 & 53 & 79 & 80 & 56 & 21 & 20 \\ 20 & 82 & 110 & 110 & 110 & 110 & 86 & 22 \\ 53 & 110 & 110 & 110 & 110 & 110 & 110 & 59 \\ 79 & 110 & 110 & 110 & 110 & 110 & 110 & 85 \\ 80 & 110 & 110 & 110 & 110 & 110 & 110 & 86 \\ 56 & 110 & 110 & 110 & 110 & 110 & 110 & 62 \\ 21 & 86 & 110 & 110 & 110 & 110 & 91 & 23 \\ 20 & 22 & 59 & 85 & 86 & 62 & 23 & 20 \end{pmatrix}$$

For the above data, the visual components can be easily calculated. These components are shown below, and indicate the edges components in a block.

$$\begin{pmatrix} 2 & 2 & 3 & 3 & 3 & 3 & 2 & 2 \\ 2 & 3 & 4 & 3 & 2 & 3 & 3 & 2 \\ 3 & 4 & 3 & 2 & 2 & 2 & 3 & 3 \\ 3 & 3 & 2 & 0 & 0 & 0 & 2 & 3 \\ 3 & 3 & 2 & 0 & 0 & 0 & 2 & 2 \\ 3 & 3 & 2 & 0 & 0 & 0 & 2 & 2 \\ 2 & 3 & 3 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 & 2 & 3 & 2 & 2 \end{pmatrix}$$

Consider a single bit "1" as copyright information ID to be embedded into a block. The differences and the watermark patterns for a JPEG quality level $q_1 = 25$ are shown in Figure 2.

The length L indicates the number of locations where the watermark is embedded. For the above data, its value is 59 out of 64. That is, 59 data among 64 are used for embedding the watermark. When these watermarked data are compressed with a quality level q_2 that is greater than q_1, we are able to reconstruct the copyright information if the signs of the extracted watermark are the same as those of the original watermark.

In another experiment we use $q_2 = 80$. The results are depicted in Figure 3. To reconstruct the embedded single bit "1" copyright information, the embedded random sequences $s_{i,j}$ are derived from Table 1 by using the signs of the extracted watermark and pattern calculated by Eqn.(6). The id_y can be calculated from $s_{i,j}$ using Eqn.(11). The embedded ID can be extracted by counting the number of 1's and 0's in id_y followed by a majority vote. In the simulation, the numbers of 1's and 0's are 31 and 28, respectively. This gives us "1" as the value of the embedded ID. Note that more efficient error correcting codes, rather than simple repetition, can also used.

In this example, the proposed method is able to reconstruct the copyright information with respect to $q_2 = 80, 75$, but not to $q_2 = 65 \sim 20$, as the size of the block is too small. In general, the larger the size of an image, the higher the chance of extraction. This motivates us to apply the proposed method to the standard image Lena(256×256 , 8bits/pixel).

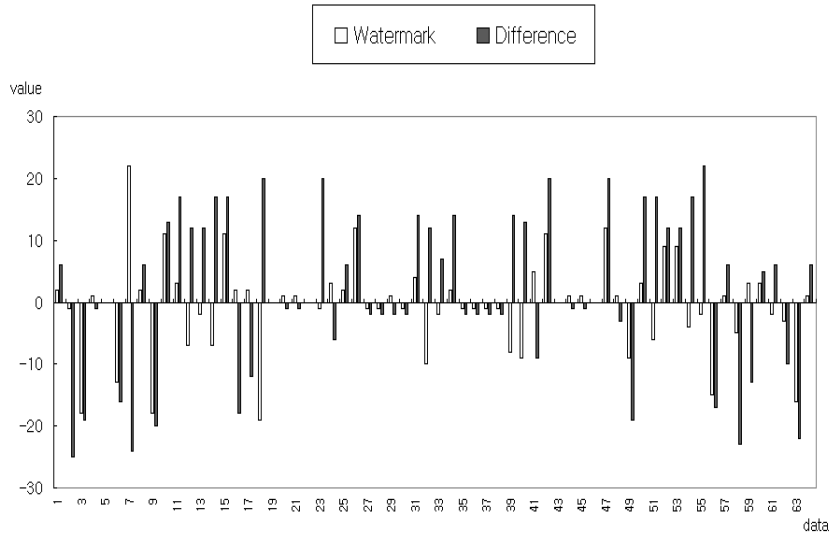

Fig. 2. The difference and watermark pattern for $q_1 = 25$

The watermarked image is shown to Figure 4. We used $q_1 = 25$ as the quality control parameter and the length of copyright information is 112 bits (14 digit ASCII characters). The watermark is embedded into 60,870 pixels among 65,536. Table 2 indicate the numbers of 1's and 0's, for the first 10 bits "0100110000" of the 112 bits.

Table 2. The numbers of 0's and 1's from the extracted bits used to construct ID

quality level	bit order (the num. of 0 : the num. of 1)				
q_2	1	2	3	4	5
80	357:187	201:343	354:190	355:189	197:347
75	329:215	219:325	343:201	339:205	213:331
60	315:229	226:318	310:234	314:230	208:336
quality level	bit order (the num. of 0 : the num. of 1)				
q_2	6	7	8	9	10
80	189:355	356:188	345:199	350:194	344:200
75	206:338	336:208	343:201	325:219	327:217
60	229:315	312:232	325:219	315:229	302:242

From Table 2, one can reconstruct correctly all the bits. Table 3 clearly indicates that it is possible to reconstruct the copyright information when the watermarked image is compressed with a quality level of q_2 greater than q_1 .

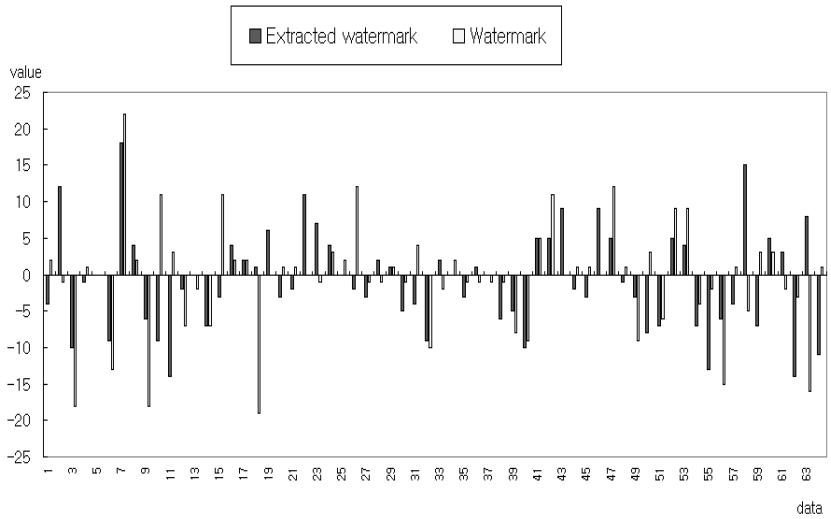

Fig. 3. Extracting a watermark for q_2

Table 3. The reconstruction of ID for $q_1 = 25$

quality level q_2	80	75	60	35	30	25	20
reconstruction	◯	◯	◯	◯	◯	◯	×

5 Conclusion

The proposed method utilizes the quality level of JPEG as a parameter in watermarking to provide robustness against JPEG compression. It calculates visual components using relationships with neighboring pixels. Locations to embed the watermark are derived from these visual components. To minimize extraction errors caused by compression, copyright information is repeated and converted into random sequences by XOR with M-sequences. As the result, the owner can extract the watermark even when the watermarked image is compressed, as long as the quality level is not smaller than the quality level used as a parameter. The copyright information can be extracted from the embedded random sequences at a later stage.

To close this paper, we remark that in this work we have considered only the quality level of the JPEG compression algorithm as a parameter. One may use different parameters related to image processing, and these parameters might provide equal or even stronger robustness against various attacks based on image processing.

Fig. 4. The watermarked image for $q_1 = 25$

Acknowledgment : *This work was supported by GRANT No. KOSEF 981-0928-152-2 from the Korea Science and Engineering Foundation.*

References

1. M.D.Swanson, M.Kobayashi and A.H.Tewfik, *Multimedia Data-Embedding and Watermarking Technologies,* In Proc. of IEEE, Vol.86, No.6, pp.1064-1087, 1998
2. M.M.Yeung et al, *Digital Watermarking*, Communications of the ACM, Vol.41, No.7, pp.31-77, 1998
3. C. T. Hsu and J. L. Wu, *Hidden Signature in Images*, In Proc. of IEEE International Conference on Image Processing, pp.223-226, 1996
4. G.Langelaar, J.van der Lubbe and J.Biemond, *Copy Protection for Multimedia Based on Labeling Techniques.*, http://www-it.et.tudelft.nl/pda/smash/public/benelux_cr.html
5. M.Kutter, F.Jordan and F.Bossen, *Digital Signature of Color Images Using Amplitude Modulation*, In Proc. SPIE-EI97, pp.518-526, 1997
6. K.Matsui and K.Tanaka, *Video-steganography: How to Embed a Signature in a Picture*, In Proc. IMA Intellectual Property, Vol.1, No.1, pp.187-206,1994
7. I.Cox, J.Kilian, T.Leighton and T.Shamoon, *Secure Spread Spectrum Watermarking for Multimedia.*, Tech.Rep., 95-10, NEC Research Institute, 1995
8. C.I.Podilchuk and W.Zeng, *Image-Adaptive Watermarking Using Visual Models*, IEEE Journal on Selected Areas In Communications, Vol.16, No.4, pp.525-539,1998
9. J.Dittmann, M. Stanenau and R. Steinmetz, *Robust MPEG Video Watermarking Technologies*, ACM Multimedia'98, 1998
10. A.B.Watson, *DCT Quantization Matrices Visually Optimized for Individual Images*, Proc. of SPIE, 1992

11. E. Koch and J. Zhao, *Towards Robust and Hidden Image Copyright Labeling*, Proc. of IEEE Workshop on Nonlinear Signal and Image Processing, 1995
12. J.Fridrich, *Methods for Data Hiding*, http://ssie.binghamton.edu/jirif/