Digital Image Watermarking using Inner Product and Quantization

Seung-Wook Lee[1], Kyung-Ae Moon[1] and Yo-Sung Ho[2]

ETRI, 161 Kajong-Dong Yusong-Gu, Taejon, 305-350, Korea[1]
K-JIST, 1 Oryong-Dong Puk-Gu, Kwangju, 500-712, Korea[2]
tajinet@etri.re.kr, kamoon@etri.re.kr, hoyo@kjist.ac.kr

ABSTRACT

Due to the wide spread of the internet, electronic data can be distributed and accessed in an effortless manner. Content providers are faced with a challenge of how to protect their electronic data, and various solutions have been proposed for copyright protection. One of them is digital watermarking.

In this paper, we propose a robust data embedding scheme using quantization and data perturbation. Our proposed scheme can embed a relatively large amount of information, including images. A secrete key is used to design random direction vectors, which take inner product with the transformed feature set. In order to achieve robustness against malicious attacks, we exploit properties of the human visual system in designing the random direction vectors. Experimental results show that the ownership assertion is possible even though the watermarked data may undergo common signal processing operations.

1. INTRODUCTION

Recent advances in information processing systems and networked databases continue to spur rapid growth of digital media applications. Production and distribution of multimedia data are all done in the digital form. Digital media provide several merits over analog media. For example, quality of digital signals is better than that of their analog counterparts. The copy process of digital media is very simple without loss of fidelity, and a copied version is identical to the original one. This feature of the digital media can open various new possibilities. For instance, because we know the exact value at a certain position, data modification is easy. It can also generate undesirable problems of illegal copies. Digital watermarking techniques have been proposed to discourage unauthorized distribution of digital media.

A watermark is a perceptually invisible signal inserted into the host data, such as image, audio, video or any other multimedia data [1]. A typical watermark contains the information of the origin, state, or the recipient of the data. The purpose of watermarking is to protect the copyright of the creator or the owner. In order that the owner could verify the copyright ownership, the embedded signal should survive under various circumstances, such as malicious attacks or intentional data modifications.

Recently, various digital watermarking techniques have been proposed. At the beginning, time-domain insertion methods were proposed; however, they are not very efficient because the inserted watermark can be easily removed. In order to overcome this problem, Cox, *et al.* introduced a frequency-domain watermarking technique [1], where the watermark signal is inserted into visually important frequency components. The watermark consists of a sequence of numbers $x = x_1, x_2, x_3, \cdots, x_n$ with a given statistical distribution. The Cox's algorithm provides improved results, but the extracted watermark signal, i.e., a sequence of numbers, does not contain any meaningful information. Furthermore, it needs the host data to detect the embedded watermark.

In this paper, we propose a new watermarking scheme that embeds a meaningful watermark signal into DCT coefficients of the feature set. We define a random directional vector, where the feature set data is projected using the modulation transfer function (MTF) of the human visual system [2]. Because the watermark data are not directly related to the private key, we can embed any kind of watermark data. We also propose how to calculate the threshold value for determining whether watermark is embedded or not.

2. WATERMARK INSERTION

Although the Cox's watermarking algorithm is popular, it needs the original image for watermarking detection and it can embed only one bit information, indicating whether a watermark exists or not [1].

Our proposed scheme can contain more than one bit information in the watermark signal. In addition, the original image is not required for the watermark recovery process. We insert the watermark information in the DCT domain. Fig. 1 shows the block diagram of the proposed scheme.

2.1. Feature Set Determination

We can partition the DCT coefficients into two regions; the region where the watermark is embedded, and the region where the watermark is not embedded. We call the watermark-embedded region as a feature set.

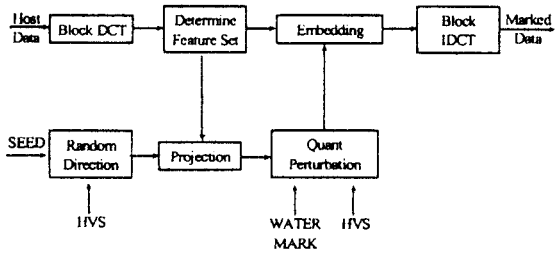


Figure 1. Block Diagram of Proposed Scheme

We choose the low frequency band as a feature set to achieve robustness [1]. By dividing the feature set into three or four disjoint groups, we can obtain higher watermark embedding capacity.

For four-bit insertion per block, the feature set can be divided as follows:

$$F_{1b}=(c_{10b},c_{20b})\quad F_{2b}=(c_{01b},c_{02b},c_{03b})$$
$$F_{3b}=(c_{11b},c_{12b})\quad F_{4b}=(c_{21b},c_{30b}), \tag{1}$$

where F_{ib} means the i-th feature set of the b-th block and c_{ijb} is the (i,j)-th DCT coefficient of the b-th block.

For three-bit insertion per block, we exploit the characteristics of the JPEG quantization tables. A large value in the quantization table indicates that the corresponding DCT coefficient can be modified significantly during the quantization operation, whereas a small value implies a small data change. In general, we have relatively small changes in low frequency components in DCT-based coding. Therefore, the watermark signal embedded in the low frequency region can survive strongly. In our proposed scheme, we try to distribute low and high frequency components in different groups, and avoid placing extreme frequency components into the same group. By experiments, we choose the following feature set.

$$F_{1b}=(c_{02b},c_{03b},c_{21b})\quad F_{2b}=(c_{10b},c_{11b},c_{30b})$$
$$F_{3b}=(c_{01b},c_{12b},c_{20b}). \tag{2}$$

2.2. Random Direction Table

The watermark embedding process can be viewed as an operation of adding invisible noise into the host data. For watermark insertion, we need to design the noise to be embedded. In this paper, we define this noise as a random direction table (RDT).

Chiprasert *et al.* [2] proposed to use the modulation transfer function (MTF) of the human visual system. Since the MTF represents the frequency sensitivity of the human eye, we can achieve invisibility of the embedded watermark by defining the noise as follows:

$$z_{ij\_row}=seed\_vector_i(j)/H(u,v), \tag{3}$$

where $z_{ij\text{-}row}$ is the j-th row of the i-th RDT and $H(u,v)$ is the MTF. $seed\_vector_i(j)$ is generated randomly by a private key. By Eq. (3), a small noise is generated in the sensitive region. For blind watermarking, each row of the RDT is normalized.

2.3. Inner Product and Quantization

We take the inner product of the feature set and the row of the RDT in order to compute the projection value, p_{ib}, which is the magnitude of the projection of the feature set onto the RDT.

$$p_{ib}=F_{ib}\cdot z_{ij\_row}\quad \text{for } i=1,2,3. \tag{4}$$

Then, we perform uniform quantization of the projection value with a step size T, i.e., $p_{ib\_q}=kT$, where k is an integer. Since the quantized value can take only multiples of T, if we add or subtract a certain pre-defined noise as a watermark signal, the modified value lies on either sides of p_{ib_q}. Thus, it is easy to determine what value is added or subtracted.

2.4. Perturbation and Embedding

During the watermark embedding operation, we modify the quantized projection, p_{ib_q}, as follows:

$$p_{ib}'=\begin{cases}p_{ib\_q}+0.25T & \text{if watermark=1}\\ p_{ib\_q}-0.25T & \text{if watermark=0.}\end{cases} \tag{5}$$

After the modification, the feature set data is altered by

$$\overline{v_{ib}}'=\overline{v_{ib}}+(p_{ib}'-p_{ib})\cdot\overline{z_{ij\_row}}. \tag{6}$$

We obtain the watermarked image by the inverse DCT. Fig. 2 illustrates the embedding operation.

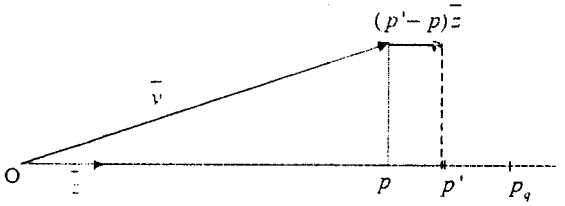


Figure 2. Watermark Embedding

In Fig. 2, we note that the watermark-embedded value p' can be computed by taking inner product of the watermarked data and each row of the RDT without using the original data. This implies that blind watermarking is possible in our proposed scheme.

3. WATERMARK EXTRACTION

The watermark extraction process is very similar to the watermark embedding process. In fact, it is the inverse process of the embedding process.

3.1. Computation of Modified Inner Product

A single watermark bit is embedded in p_{ib_q} to generate p'_{ib_q}. Thus, if we can recover p'_{ib}, the embedded bit is easily obtained. For blind watermarking, p'_{ib} should be computed without the original data. Since the RDT is normalized, we can compute p'_{ib} without the original unwatermarked data by Eq. (7).

$$\bar{v}'\cdot\bar{z}=[\bar{v}+(p'-p)\bar{z}]\cdot\bar{z}=p+(p'-p)=p'. \quad (7)$$

3.2. Watermark Extraction

We can extract the embedded watermark by Eq. (8).

$$b^{*}=\begin{cases}1 \text{ if } \dfrac{p_{ib}'}{T}-\left[\dfrac{p_{ib}'}{T}\right]>0\\ 0 \text{ if } \dfrac{p_{ib}'}{T}-\left[\dfrac{p_{ib}'}{T}\right]<0\end{cases}, \quad (8)$$

where $[x]$ represents the rounding operation of the element x to the nearest integer. As we can see in Eq. (5), p'_{ib} is modified from p_{ib_q} by $T/4$. Therefore, the embedded watermark information can be easily recovered by checking whether the modified one (p'_{ib}) is bigger or smaller than the original one (p_{ib_q}).

3.3. Determination of Detection Threshold Value

We can determine whether the watermark is inserted or not according to the following two criteria: (a) quality of the extracted watermark, and (b) similarity between the embedded watermark and the extracted watermark.

Since the second criterion requires a detection threshold value, we can define a similarity measure by

$$X(similarity)=\sum_{i=1}^{M}s_i, \quad (9)$$

where s_i is the correlation between the i-th embedded bit and the i-th extracted bit. It takes '1' if two bits are the same, and '0' otherwise.

In order to compute the detection threshold value, we denote p for the probability of the i-th embedded watermark bit to be '1', and r for the probability of the i-th extracted watermark bit to be '1'. Assuming that the extracted and the embedded watermark bitstreams are independent, we can obtain $Pr(s_i=1)=pr+(1-p)(1-r)$.

Here, we know that s_i is a Bernoulli random variable that has only two values, i.e., '0' or '1' [3]. The mean (m) of s_i is $pr+(1-p)(1-r)$, and the variance (σ^2) is $(1+2pr-p-r)(p+r-2pr)$. By Eq. 9, the similarity X is a summation of independent identically distributed (*iid*) variables of Bernoulli distribution. Thus, X is a Binomial random variable. By the central limit approximation, it has the Gaussian distribution.

$$\Pr(X_G=k)=\int_{k}^{k+dk}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)dx. \quad (10)$$

In the proposed method, we conclude that the watermark is embedded if the similarity X_G is larger than the detection threshold value Th. Therefore, the false alarm rate is calculated by

$$\Pr(error)=\int_{Th}^{\infty}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(x-m)^2}{2\sigma^2}\right)dx, \quad (11)$$

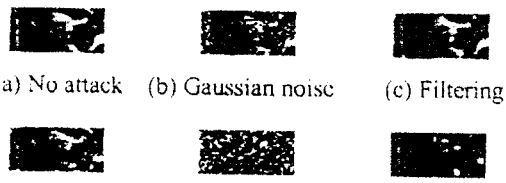
which is represented by $Q((Th-m)/\sigma)$. In this paper, we set the detection threshold value by $Th=4\sigma+m$, and have 3.17×10^{-5} for the false alarm rate.

4. SIMULATION RESULTS

We perform computer simulations to evaluate the proposed watermarking algorithm with two test images, 'Lenna' and 'Barbara', of 512x512 pixels and eight-bit gray levels. The total number of embedded bits is $\lfloor 512/8\rfloor \times \lfloor 512/8\rfloor \times 3$ or $\lfloor 512/8\rfloor \times \lfloor 512/8\rfloor \times 4$ for the three-bit or four-bit insertion per block, respectively. The threshold values are 6365.7 and 8448, respectively.

(a) Lenna (b) Barbara

Figure 3. Original Test Images



(a) No attack (b) Gaussian noise (c) Filtering

(d) JPEG (e) Resizing (f) Cropping

Figure 4. Extracted Watermark

Fig. 3 shows the original test images, and Fig. 4 shows extracted watermarks under various conditions for four-bit insertion per block. Fig. 4(a) is the extracted watermark when there was no attack. Fig. 4(b) is the result when the Gaussian noise is added. As we can see in Fig. 4(b), the reconstructed image is contaminated by the salt-and-pepper noise, which can be removed by median filtering, as shown in Fig. 4(c). Fig. 4(d) displays the extracted watermark after JPEG compression. An extracted image after a resizing attack is shown in Fig. 4(e), where the ownership assertion is

still possible because the similarity measure is 9126. Fig. 4(f) shows a reconstructed watermark image under a cropping attack.

Table 1 summarizes similarity measures for JPEG compression, and Table 2 shows similarity measures when cropping and JPEG compression are applied at the same time.

Table 1. Similarity Measures for JPEG Compression

Test Images		Q-factor			
		60	70	80	90
3 bits per block	Lenna	10088	8526	6692	6489
	Barbara	10336	8832	6717	6564
4 bits per block	Lenna	12936	9694	8599	8780
	Barbara	13519	9779	8512	8666

Table 2. Similarity Measures for Cropping and JPEG Compression with fixed Q-factor 70

Test Images		Cropping Rate			
		0.6	0.7	0.8	0.9
3 bits per block	Lenna	7287	7047	6791	6518
	Barbara	7095	6832	6569	6293
4 bits per block	Lenna	8890	8782	8594	8384
	Barbara	8746	8599	8425	8293

In order to extract the watermark signal from the cropped image, missing portions of the image were replaced by the portions from the original image. Cropping rate 0.6 means that 60% of the watermarked data is lost.

Table 3 shows that we can still recover the embedded watermark up to noise factor 12, when we have 26.3 dB of PSNR for both Lenna and Barbara images.

Table 3. Similarity Measures for Noise Addition

Test Images		Noise Factor			
		10	11	12	13
3 bits per block	Lenna	7095	6790	6502	6381
	Barbara	7039	6795	6479	6290
4 bits per block	Lenna	9383	8953	8676	8441
	Barbara	9304	8881	8544	8440

Similarity measures for other attacks are summarized in Table 4, where AF, MF and HE imply 5x5 averaging filter, 5x5 median filtering and histogram equalization, respectively.

Table 4. Similarity Measures for Other Attacks

Test Images		Attacks		
		AF	MF	HE
3 bits per block	Lenna	7253	7653	7338
	Barbara	6821	7045	8530
4 bits per block	Lenna	10371	10849	9926
	Barbara	9650	9862	11867

5. CONCLUSIONS

In this paper, a new image watermarking algorithm has been developed. Since the embedded watermark is not directly related to the private key, we can embed any kind of multimedia data into the host data as a watermark signal. We consider properties of the human visual system in designing the embedded noise to make the algorithm robust to various attacks. The detection threshold value is computed from a probability model of the similarity measure. Simulation results show that the proposed watermarking algorithm is quite robust to common image processing operations.

6. ACKNOWLEDGMENT

This work was supported in part by KOSEF through UFON, and in part by MOE through BK21.

7. REFERENCES

[1] I. J. Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. on Image Processing*, vol. 6, no. 12, pp. 1673-1687, Dec. 1997.

[2] B. Chiprasert and K. R. Rao, "Human Visual Weighted Progressive Image Transmission," *IEEE Trans. on Communications*, vol. 38, no. 7, pp. 1040-1044, July 1990.

[3] A. Leon-Garcia, *Probability and Random Process for Electrical Engineering*, Addison-Wesley Publishing Company, 1994

[4] M. D. Swanson, B. Zhu and A. H. Tewfik, "Data Hiding for Video-in-Video," *International Conference on Image Processing*, vol. 2, pp. 676-679, Oct. 1997.

[5] S. W. Lee, "Digital Image Watermarking using Inner Product and Quantization," M.S. Thesis, Kwangju Institute of Science and Technology, Feb. 2001.