A Novel Robust Watermarking Technique Using IntDCT Based AC Prediction

Kuo-Ming Hung[12]
[1]Department of Information Management, Kainan University, Taiwan
[2] Department of Electrical Engineering, TamKang University, Taiwan .
[1]No.1 Kainan Road, Luzhu Shiang, Taoyuan 338, Taiwan.
[2]#151, Ying-Chuan Road, Tamsui, Taipei Hsien 25137, Taiwan.
hkming@mail.knu.edu.tw

Abstract: -Because of the blocking artifacts resulted from 8x8 Discrete Cosine Transform (DCT) , most watermarking technologies have been proposed using DCT whose image quality was not very good. In 1990, Gonzales et al. described a technique which predicts a few low frequency AC coefficients. The AC predictor uses the dequantized DC values of a 3x3 neighborhood of 8x8 blocks to predict the AC values in the center block. Wang proposed a data hiding scheme using the AC prediction technology in 2005. But it is unable to be suitable for the image of all types to predict AC coefficients accurately.

We propose a new watermarking system by using the technologies of 4x4 integer DCT transform and adaptive AC estimation. We use 4x4 INTDCT transform to reduce blocking artifacts caused from 8x8 DCT transform and improve Imperceptibility and watermark capacity greatly. Moreover, we utilize AC prediction value as error-checking code to enhance robustness of watermark.

Key-Words:-Watermark;DCT;IntDCT;AC prediction; H.264

1 Introduction

Digital watermarking is a technique for embedding a watermark into a digital image to protect the owner's copyright of the image. Many watermarking techniques have been proposed in recent years. Digital watermark strategies fall into two major categories: spatial-domain and transform-domain techniques. In spatial domain techniques,one of the simplest methods of inserting a digital watermark in a still image is called Least-Significant-Bit (LSB) Watermarking [1]. However, this technique has relatively low information hiding capacity and can be easily erased by lossy image compression. Techniques like superimposing a watermark image over an area of image to be watermarked[2] and signal-adaptive addition [3] are used to embed watermarks in the spatial domain. Watermarks can also be inserted in the frequency domain by applying transforms like Fast Fourier Transform (FFT) [4], Discrete Cosine Transform (DCT) [5],and Discrete Wavelet Transform[6] and then altering the values of selected transform coefficients to store the watermark in still images. Watermarking in the frequency domain is more robust than watermarking in the spatial domain [7], because the watermark information can be spread out to the entire image

[8].In this paper, we propose a new watermarking system by using the technologies of 4x4 Integer Discrete Cosine Transform (Integer DCT) and adaptive AC estimation. We use 4x4 INTDCT transform to reduce blocking artifacts caused from 8x8 DCT transform. And we modify the AC estimation scheme[23] [24] to predict AC(1,1) as error-checking code. Using the error-checking code, we can enhance the robustness of watermark.

The reminder of this paper is organized as follow. Section 2 describes IntDCT and AC Prediction. Section 3 presents the watermarking algorithm. Section 4 presents the experimental results. Section 5 provides concluding remarks.

2 IntDCT and AC Prediction

2.2.1 Integer Discrete Cosine Transform

The proposed watermarking technique uses the Integer DCT(IntDCT) which is used in H.264 video standard[25]. H.264 applies this new 4x4 transform instead of the 8X8 discrete cosine transform (DCT) traditionally used in many video coding standards. IntDCT has at least two advantages over DCT. First, IntDCT needs no floating-point multiplications. The floating-point multiplications are replaced by lifting steps that need only integer operations and shifting. This is very important for applicationsin mobile devices since it is easier and cheaper (power saving) to realize integer operations than to implement float-point multiplications. Second, if sufficient word length is used to represent the intermediate data of IntDCT, the round-off error can be eliminated completely. There is no information lost after the transform even if it is computed in a fixed-point computer.

The 4x4 IntDCT transform can reduce blocking artifacts resulted from 8x8 DCT transform. The integer transform is designed such that the transformation involves only additions and shift operations, and no mismatch exists between the forward and inverse transforms. This reduces the computational complexity and it is much easier for hardware implementation. J. Zhang and Anthony T. S. Ho[26] describe DCT as follows:

To develop this integer DCT, they examine the 4x4 DCT of a 4x4 matrix X:

$$Y=\begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} * X * \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}^T \tag{1}$$

A=1/2, $b=\sqrt{1/2}\cos(\pi/8)$, $c=\sqrt{1/2}\cos(3\pi/8)$

The matrix multiplication can be expressed to the following form:

$$Y=(B*X*B^T)\otimes Q \tag{2}$$

where

$$B=\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix},\quad Q=\begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

$(B*X*B^T)$ is a "core" 2-D transform. ***Q*** is a matrix of scaling factors and the symbol $\otimes$ indicates that each element of $(B*X*B^T)$ is multiplied by the scaling factor in the same position in matrix *Q*. *d* is *c/b* (approximately 0.414).

To simplify the implementation of the transform, *d* is approximated by *0.5*. To ensure that the transform remains orthogonal, b also needs to be modified so that:

$a=1/2$, $b=\sqrt{2/5}$, $d=1/2$

The 2^{nd} and 4^{th} rows of matrix *B* and the 2^{nd} and 4^{th} columns of matrix B^T a r e scaled by a factor of **2** and the post-scaling matrix ***Q*** is scaled down for compensation. This avoids multiplications by **1/2** in the core transform $(B * X * B^T)$ **,** which would result in loss of accuracy using integer arithmetic. The final forward 4x4 IntDCT becomes:

$$Y = (B_f * X * B_f{}^T) \otimes Q_f \tag{3}$$

Where

$$B_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}, \quad Q_f = \begin{bmatrix} a^2 & ab & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}$$

This transform is an approximation to the 4x4 DCT.

The inverse transform is given by

$$X' = B_i^T * (Y \otimes Q_i) * B_i \tag{4}$$

where

$$B_i = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix}, \quad Q_i = \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

2.2 Data embedding technique based on the prediction of AC coefficients

2.2.1 AC Prediction

A DCT is basically a structural decomposition in terms of basis images shown in Fig.1. The left basis image represents the frequency distribution of the DCT block, and the other basis image on the row (column) represent the vertical (horizontal) edges.

In [23], Gonzales et al. described the concealment technique, which predicts a few low frequency AC coefficients. The AC predictor uses the dequantized DC values of a 3x3 neighborhood of 8x8 blocks to predict the AC values in the center block. The estimation formulae for the first five unquantized DCT AC coefficients are shown in Fig.2.

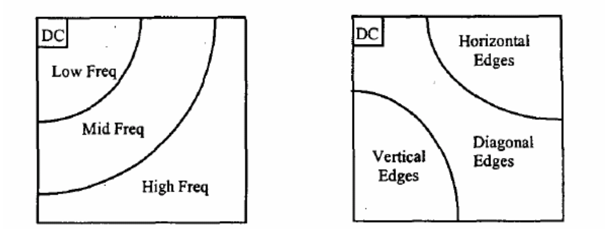


Fig.1 Frequency Distribution in a DCT block (left), Structural decomposition in a DCT block(right)

$$AC(0,1) = 0.14235 * (DC_4 - DC_6) \tag{5}$$

$$AC(1,0) = 0.14235 * (DC_2 - DC_8)$$

$$AC(0,2) = 0.03485 * (DC_4 + DC_6 - 2 * DC_5)$$

$$AC(1,1) = 0.02026 * (DC_1 + DC_9 - DC_3 - DC_7)$$

$$AC(2,0) = 0.03485 * (DC_2 + DC_8 - 2 * DC_5)$$

Block 1 DC1	Block 2 DC2	Block 3 DC3
Block 4 DC4	Central Block DC5	Block 6 DC6
Block 7 DC7	Block 8 DC8	Block 9 DC

DC	AC(0,1)	AC(0,2)					
AC(1,0)	AC(1,1)						
AC(2,0)							

Fig.2 Neighborhood of DCT blocks used for AC prediction

2.2.2 Yulin Wang and Alan Pearmain's watermark embedding technique

Yulin Wang and Alan Pearmain's [24] used Gonzales 's[23] AC prediction technique to propose a robust watermarking technique. They select every nine 8x 8 blocks as one group, in which 5 watermark bits can be embedded by modulating the above 5 AC components in Eq.(5) referring to its predicted values with the following translation rule.

Set $AC_i \geq AC_i^{'} + \Delta$ to embed bit '1' (6)

Set $AC_i < AC_i^{'} - \Delta$ to embed bit '0'

In Eq.(6), AC_i is the real value of one of the 5 AC components: AC(0,1), AC(1,0), AC(0,2), AC(1,1) and AC(2,0). $AC_i^{'}$ is the predicted value of AC_i by using Eq.(5). Δ is a reference threshold. From their experiment, Δ can be chosen as 5–15% of the original AC_i value.

3. The proposed technique

3.1 Embedding the watermark

Fig.4 shows the procedure of embedding the watermark and the steps of embedding approach are listed as the following:

Step1. The original host image f(x,y) is divided into k non-overlapping blocks of size 4x4 pixels.

$$f(x,y) = \bigcup_{k=0}^{K-1} B_k = \bigcup_{k=0}^{K-1} f_k(i,j) \qquad 0 \leq i,j < 4 \quad (7)$$

Step2. Transform each 4x4 block by Integer DCT

$$F_k^*(u,v) = IntDCT(f_k^*(i,j)) \, 0 \leq u,v < 4 \qquad (8)$$

Step3. convert the original watermark image into original watermark series and Repeat the original watermark series W n times to obtain repeated watermark series Wn. Then the watermark series Wn will be permuted and its random seed will be recorded.

Step 4. Insert repeated watermark series Wn into the coefficients of DC value. The embed method is described as follows

If $Wn(k) = -1$

$$F_k^{'}(u,v) = \begin{cases} \Delta * Q_0\left(\dfrac{F_k(u,v)}{\Delta}\right) & if \quad u=v=0 \\ F_k(u,v) & otherwise \end{cases} \qquad (9)$$

If $Wn(k) = 1$

$$F_k^{'}(u,v) = \begin{cases} \Delta * Q_1\left(\dfrac{F_k(u,v)}{\Delta}\right) & if \quad u=v=0 \\ F_k(u,v) & otherwise \end{cases} \qquad (10)$$

Where $Q_0(x)$ indicates to turn the value of x to the most approximate even number, and $Q_1(x)$ indicates to turn the value of x to the most approximate odd number, and △ is the parameter of quantization.

Step5 predict the coefficients of AC(1,1),and replace the actual value with its predicted values.

We modify Gonzales et al.'s[23] method described in section 2.22. Eq.(11) is the modified Estimation method.

$$AC(1,1) = \alpha * (DC_1 + DC_9 - DC_3 - DC_7) \qquad (11)$$

Where α =0.02, which is experimental result that we use ten natural images to test. When α =0.02, the average quality of image is the best as Fig.3.

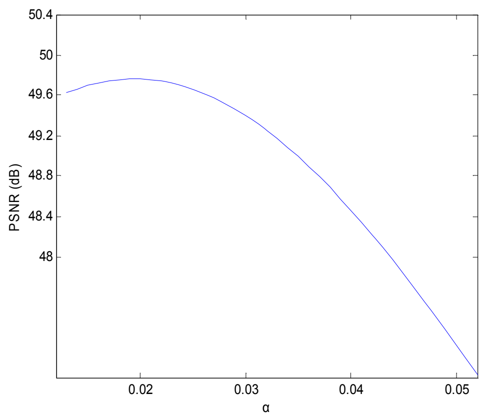


Fig.3 the average PSNR of the ten natural images

Step6. Inverse IntDCT transform to obtain watermarked image

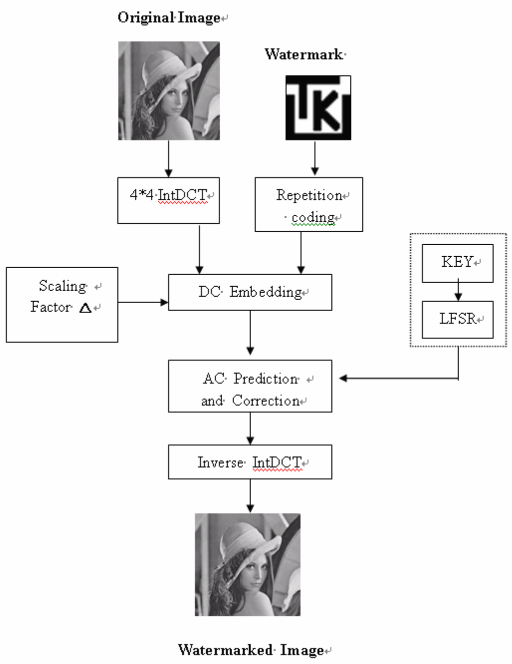


Fig.4 the procedure of embedding the watermark

3.2 Extracting the watermark

Fig.5 shows the procedure of extracting the watermark and the steps of extracting approach are listed as the following:

Step1. The watermarked image $f^*(x,y)$ is divided into k non-overlapping blocks of size 4x4 pixels. Each 4x4 block is denoted by B_k^*.

Step2. Transform each 4x4 block by Integer DCT

$$F_k^*(u,v) = IntDCT(f_k^*(i,j)) \qquad 0 \le u,v < 4 \quad (13)$$

Step3. extracted the repeated watermarking series from coefficients of DC as Eq.(14)

$$if \quad Q(\frac{F_k^*(0,0)}{\Delta}) = even \quad then \quad w_k^* = -1 \quad (14)$$

$$if \quad Q(\frac{F_k^*(0,0)}{\Delta}) = odd \quad then \quad w_k^* = 1$$

where $Q(x)$ is round x to integer.

Step4 Compare AC^* and $F_k^*(1,1)$ to find if the 4x4 block is attacked

$$d_k^*(u,v) = \begin{cases} F_k^*(u,v) & , \quad F_k^*(1,1) = AC^* \\ 0 & , \quad otherwise \end{cases} \quad (15)$$

$$d^*(x,y) = \bigcup_{k=0}^{K-1} d_k^*(u,v) \qquad 0 \le u,v < 4 \quad (16)$$

$$E^*(k) = \begin{cases} 0 & , \quad |F_k^*(1,1) - AC^*| > \rho \\ 1 & , \quad otherwise \end{cases} \quad (17)$$

where $F_k^*(1,1)$ is the actual AC(1,1) vale of each 4x4 block by step 2., AC^* is its predicted value by Eq.(11), $d_k^*(u,v)$ is 4x4 denoted sub-image, $d^*(x,y)$ is the denoted image, ρ is the parameter of sensitivity to attacking. We can find attacked area from image $d^*(x,y)$ .we use $E^*(k)$ as error-checking code. When $E^*(k)$ =0, indicted that the block is attacked.

Step5. Use the random seed recorded to retrieve the repeated watermark $w^*(k)$.

Step6: Use error-checking code $E^*(k)$ and the repeated watermark series $w^*(k)$ to find extracted original watermark series $w^{**}(k)$, the algorithm is shown as Eq.(18).

$$w^{**}(i) = \begin{cases} -1 & if \quad \sum_n W_i^*(n) * E_i^*(n) < 0 \\ 1 & if \quad \sum_n W_i^*(n) * E_i^*(n) \ge 0 \end{cases} \quad (18)$$

where $w^{**}(i)$ is the i[th] bit of the original watermark series, n is the repeated times of watermark series.

Step7: convert the extracted original watermark series into watermark image

$$f^{*}(x,y)=\bigcup_{k=0}^{K-1} B_k^{*}=\bigcup_{k=0}^{K-1} f_k^{*}(i,j) \qquad 0\le i,j<4 \quad (12)$$

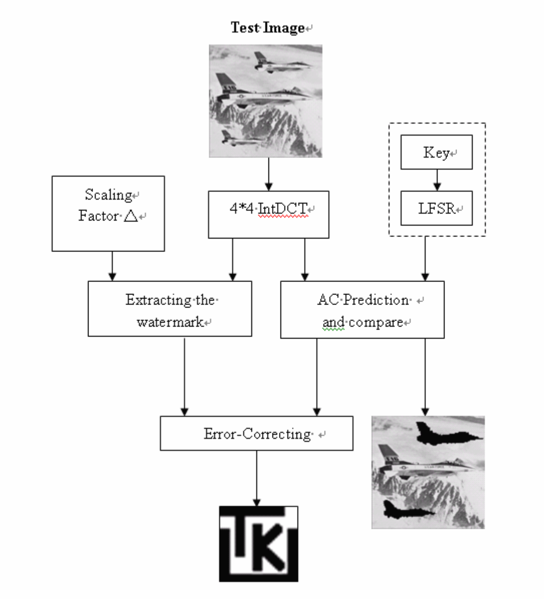


Fig.5 the procedure of extracting the watermark

4. EXPERIMENT RESULT

To test the algorithm, we use nine 512x512 images(Lenna、Baboon、F16、Aerial、Pepper、Couple、Splash、Sailboat on Lake、Stream and Bridge) as host images. Watermarks(64x64) are embedded in these images. Fig.6 shows the nine 512x512 host images and Fig.7 shows the 64x64 watermark.

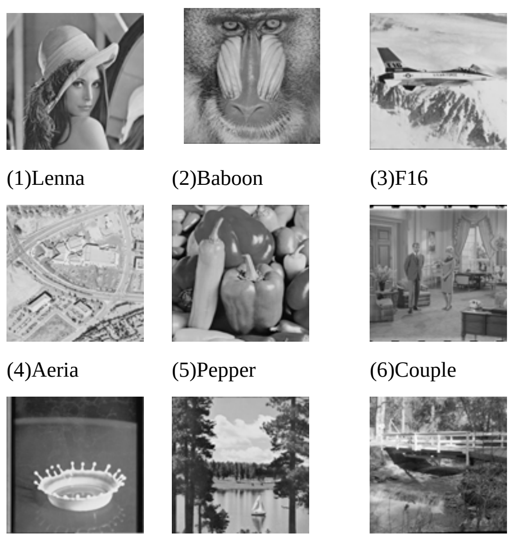
In the general digital image processing, we usually measure the degree of the image distortion with the PSNR value (the peak signal-to-noise ratio). The PSNR value was defined as

$$PSNR(dB)= 10\log_{10}\frac{I_{\max}^{2}}{\frac{1}{M\times N}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f'(x,y)-f(x,y)]^{2}} \qquad (19)$$

where I_{max} =255,when the image is a 8 bit grey image. $M\times N$ is the size of image. f(x,y) is the original image and $f'(x,y)$ is the watermarked image. And we use Correct Decoding Rate to express the quality of extracted watermark. The Correct Decoding Rate was defined as

$$\gamma=\frac{\sum_i\sum_j XNOR[w(i,j),w^{*}(i,j)]}{\sum_i\sum_j[w(i,j)]^{2}} \qquad (20)$$

where W(i, j) is the original watermark and $w^{*}(i,j)$ is the extracted watermark.



(1)Lenna (2)Baboon (3)F16

(4)Aeria (5)Pepper (6)Couple

(7)Splash (8)Sailboat on Lake (9)Stream and Bridge

Fig.6 Nine 512x512 host images

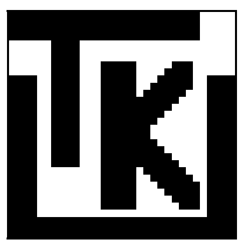


Fig.7 watermark(64x64)

4.1 Imperceptibility and watermark capacity

In order to compare watermark imperceptibility among different images, we calculate the PSNR by using the original images and the watermarked images and list the results in table 1. Table 1 shows the PSNRs of Baboon and Aerial are worse. That is because the pixel value is very different to two sides near the content boundary. There are a lot of edges in the both images.

Table 1 Imperceptibility using PSNR

Image no.	image	PSNR
1	Lenna	42.98 dB
2	Baboon	37.22 dB
3	F16	42.80 dB
4	Aerial	38.07 dB
5	Pepper	44.21 dB
6	Couple	42.50 dB
7	Splash	47.54 dB
8	Sailboat on Lake	39.94 dB
9	Stream and Bridge	38.91 dB

We compare the imperceptibility and watermark capacity of our watermark embedding technique and Yulin Wang and Alan Pearmain's watermark embedding technique[24]. The results are listed in Table 2 and Table 3.

Table 2 Watermark capacity(bit) Image Size 512*512

Y. Wang's technique	Our IntDCT technique
2205	16384

Table 3 PSNR(db)

Image	Y. Wang's technique	Our IntDCT technique
Lenna	39.1	42.9
Baboon	37.2	37.1
F16	38	42.8
Pepper	41	44.2

4.2 Image processing and geometrical attacking

We test ten different attacking methods to the watermarked image Lenna such as (1)blurring(5x5),(2)median filtering(7x7),(3)scale to 25% of original watermarked image,(4)high pass filtering(5x5),(6)histogram equalization,(7)gauss noise adding,(8)dust and scratches,(9)mosaic and (10)pinch. The correct decoding Rates of extracting watermarks are shown in Fig.8.

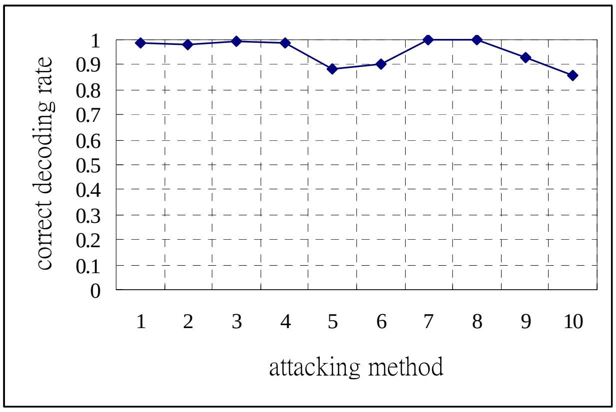


Fig.8 The correct decoding Rates of extracting watermarks

4.3 JPEG compression attacks

Fig.9 shows the correct decoding Rates between the original watermarks and the extracted watermark that the watermarked image Lenna is attacked by various JPEG compression. The experiments show that the watermark is extracted successfully under attacks of different JPEG quality compression

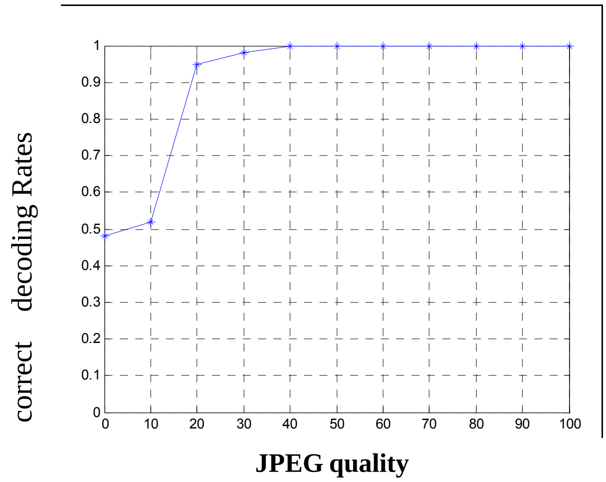


Fig.9 The correct decoding rates under attacks of different JPEG quality compression

5 Conclusion

This paper has present an watermarking technique bases on the 4x4 Integer Cosine Transform(IntDCT) and AC prediction. The 4x4 IntDCT transform can reduce blocking artifacts caused from 8x8 DCT transform and improve Imperceptibility and watermark capacity greatly. Moreover, we utilize AC prediction value as error-checking code to enhance robustness of watermark. For future work, the proposed watermarking technique will be extend to video watermarking based on the H.264 standard.

Acknowledgment

The research is supported by National Science Council, Taiwan, at the contract number NSC 95-2221-E-032-047

References

[1] R G. van Scbyndel, A. 2. Tirkel, and C. F. Shamoon,“Secure spread specbum watermarking for multimedia,”IEEE Trans. Image Processing, vol. 6, pp. 1673-1687, Dec.1997.

[2] C. Langelaar, J. C. A. Vea der Lubbe, and R L. Lagendijk, “Robust labeling method for copy protection of images,” Proceedings of SPWIS&T Electronic lmaging, San Jose, CA, vol. 3022, pp. 298-309, Feb. 1997.

[3] M. Bami, F. Bartolini, V. Cappelhi, and A. F’iva, “A DCT domian system for robust image watermarking,” Signal Processing (Special Issue on Watermarking), vol. 66, no. 3, pp. 357-372, May 1998.

[4] C.-W. Tang and EL-M. Hang “A feature-based robust digital image watermarking scheme,” IEFE Transactions on Signal Processing, vol. 51, no. 4, pp. 950-959, Apr.2003.

[5] M. Bami, F. Bartolini, V. Cappellini, and A. Piva, “A DCT-domian system for robust image watermarking,” Signal Processing (Special Issue on Watermarking), vol. 66, no. 3, pp. 357-372, May 1998.

[6] Tay,R.; Havlicek, J.P. “Image Watermarking Using Wavelets”, Proc. IEEE. Vol. 3, pp.III-258 - III-261, 2002.

[7] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for Data Hiding,” IBM Systems Journal, vol. 35, no. 3-4, 1996, IBM,USA, pp. 313-336.

[8] I.J. Cox, J. Killian, T. Leghton, and T. Shamoon, “Secure Spread Spectrum for Multimedia,” IEEE Trans. on Image Processing, vol. 6, no. 12, 1997, pp. 1673-1687.

[8] Yiwei Wang, Doherty J.F., Van Dyck R.E., “A POCS-based algorithm to attack image watermarks embedded in DFT amplitudes”, Circuits and Systems, Proceedings of the 43rd IEEE Midwest Symposium on Volume 3, pp. 1100 - 1103 vol.3, 2000.

[9] Nikolaidis A., Pitas I., “Asymptotically optimal detection for additive watermarking in the DCT and DWT domains” Image Processing, IEEE Transactions on Volume 12, Issue 5, pp. 563 – 571, 2003.

[10] Ng T.M., Garg H.K., “Maximum-likelihood detection in DWT domain image watermarking using Laplacian modeling”, Signal Processing Letters, IEEE Volume 12, Issue 4, pp. 285 – 288, 2005.

[11] Shih-Hsuan Yang, “Filter evaluation for DWT-domain image watermarking” Electronics Letters Volume 39, Issue 24, pp. 1723 – 1725, 2003.

[12] Serdean C.V., Ambroze M.A., Tomlinson M., Wade J.G., “DWT-based high-capacity blind video watermarking, invariant to geometrical attacks”, Vision, Image and Signal Processing, IEE Proceedings-Volume 150, Issue 1, pp. 51 –

58, 2003.

[13] Chiou-Ting Hsu, Ja-Ling Wu, "DCT-based watermarking for video", Consumer Electronics, IEEE Transactions on Volume 44, Issue 1, pp. 206 – 216, 1998.

[14] Chuan-Fu Wu, Wen-Shyong Hsieh, "Digital watermarking using zerotree of DCT", Consumer Electronics, IEEE Transactions on Volume 46, Issue 1, pp. 87 - 94, 2000.

[15] Hernandez J.R., Amado M., Perez-Gonzalez F., "DCT-domain watermarking techniques for still images: detector performance analysis and a new structure" Image Processing, IEEE Transactions on Volume 9, Issue 1, pp. 55 – 68, 2000.

[16] S.D. Lin, Chin-Feng Chen, "A robust DCT-based watermarking for copyright protection", Consumer Electronics, IEEE Transactions on Volume 46, Issue 3, pp. 415 – 421, 2000.

[17] Barni M., Bartolini F., De Rosa, A., Piva A., "Capacity of full frame DCT image watermarks", Image Processing, IEEE Transactions on Volume 9, Issue 8, pp. 1450 – 1455, 2000.

[23] Gonzales, C.A., Allman, L., Mccarthy, T., Wendt, P., "DCT coding for motion video storage using adaptive arithmetic coding", Signal Processing: Image Comm. 2 (2), 1990.

[24] Wang Yulin, Pearmain, Alan, "Blind image data hiding based on self reference", Pattern Recognition Letters Volume: 25, Issue: 15, November, pp. 1681-1689, 2004

[25] "H.264/MPEG-4 Part 10: Transform & quantization," ITU-T Rec. H264/ISO/IEC 11496-10, Final Committee Draft, Document JTV-F100, Dec. 2002. (Latest version modified on 19.03.2003) [Online] Available: http://www.vcodex.fsnet.co.uk/h264.html

[26] J. Zhang and Anthony T. S. Ho," An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform (IntDCT)", ICICS-PCM 2003,15-18 December 2003,Singapore.

[27] H. Zhang, L.Z. Cai, X.F. Meng, X.F. Xu, X.L. Yang, X.X. Shen and G.Y. Dong, "Image watermarking based on an iterative phase retrieval algorithm and sine–cosine modulation in the discrete-cosine-transform domain", Optics Communications, In Press, Uncorrected Proof, Available online 4 May 2007.

[28]Santa Agreste, Guido Andaloro, Daniela Prestipino and Luigia Puccio, "An image adaptive, wavelet-based watermarking of digital images", Journal of Computational and Applied Mathematics, In Press, Corrected Proof, Available online 1 February 2007.