

수정된 DCT 계수 기반의 Blind Watermarking 시스템의 하드웨어 설계 및 구현

윤승주, 정진일, 채봉수*, 조용범
건국대학교 전자정보통신공학과, *삼성전자

Blind Watermarking System

Based on the Modified DCT Coefficient

Seung Joo Yoon, Jin Il Chung, Bong Soo Chae*, Yong Beom Cho
Dept. of Electronics Engineering, Konkuk University
System LSI ASIC P/J Group, Samsung Electronics*
E-mail : sjyoon@konkuk.ac.kr

Abstract

본 논문에서는 디지털 데이터 내에 사용자의 정보나 저작권 정보를 나타내기 위해 삽입되는 워터마크를 추출할 때 원본 이미지나 개인 키를 필요로 하지 않는 Blind Watermarking 방법을 개선하였다. 기존의 워터마킹 방법에서는 워터마크를 추출하기 위해 원본 이미지를 사용하거나 원본 이미지를 사용하지 않는 경우에는 개인 키를 사용하여 워터마크를 추출하였다. 제안하는 워터마킹 알고리즘은 워터마크를 주파수 대역 별로 삽입하는 것으로서 수정된 DCT 계수를 기반으로 하였고, 삽입 및 추출 연산의 복잡성을 배제하여 속도가 빠르고 하드웨어의 구조가 간단하다. 또한, 워터마크를 저 주파수 대역과 고 주파수 대역에 삽입하여 압축 및 에러 환경에 강인한 성격을 가진다. 제안한 알고리즘의 FPGA 와 PCI Interface 를 통한 구현 및 검증에 대해서도 논하였다.

I. 서론

워터마킹은 워터마크를 추출할 때 원본 이미지나 원본 워터마크가 필요한지 여부에 따라 Non-Blind, Blind

Watermarking 으로 분류될 수 있다.

Non-Blind Watermarking 은 워터마크의 정확도는 높으나 워터마크를 추출하기 위해서는 항상 원본 디지털 데이터가 요구된다는 단점이 있어 적용할 수 있는 범위가 제한적이다. Non-Blind Watermarking 은 워터마크를 삽입할 때 암호화에서와 같이 개인 키를 사용하거나 워터마크를 의사 랜덤 noise 처럼 생성하여 워터마크를 삽입한다. 따라서 워터마크를 추출하기 위해서는 개인 키 값이나 랜덤 noise 생성 함수를 가지고 있어야 한다. 디지털 방송 등의 여러 사람이 사용할 수 있는 분야에 적용하기 위해서는 원본 영상을 필요로 하지 않고, 암호화에서와 같은 키 관리 문제를 해결할 수 있는 워터마킹 방법이 필요하다.

본 논문에서는 영상을 이산 코사인 변환을 하여 DCT 계수의 LSB(Least Significant Bit) 세 비트에 삽입하는 Blind Watermarking 방법을 제안한다.

II. Digital Watermarking 기법

2.1 공간 영역에서의 Watermarking 기법

워터마크 신호를 영상의 픽셀 값에 직접 삽입하는 방법이다. 대표적인 예로 Caronni 가 제안한 LSB 변환 방

법^[1]이 있다. LSB 변환 방법은 워터마크 영상인 L 비트의 2진 마스크 값과 픽셀의 LSB 값을 비교하여 식(2-1)과 같이 같으면 임의의 값을 더해주고, 그렇지 않으면 빼준다. 워터마크의 추출은 식(2-2)와 같이 원 영상과 워터마크가 삽입된 영상의 차이를 계산하고 화소 별로 차이 값의 부호를 조사하여 추출한다. LSB 기법은 다른 방법에 비해 상대적으로 많은 정보를 영상에 삽입할 수 있으나 시각적인 관련성을 사용하지 않고 고주파수에 noise를 더해주는 것과 같으므로, 저대역 필터링에 약하며 원 영상이 추출 시 항상 필요하다.

$$W \otimes LSB = 1 \rightarrow \tilde{I} = I + \alpha \quad (2-1)$$

$$W \otimes LSB = 0 \rightarrow \tilde{I} = I - \alpha$$

$$\tilde{I} - I > 0 \rightarrow W = 1 \quad (2-2)$$

$$\tilde{I} - I < 0 \rightarrow W = 0$$

2.2 주파수 영역에서의 Watermarking 기법

영상을 DCT, DWT, DFT 등으로 변환하여 변환된 주파수 계수에 워터마크를 삽입하여, 압축 및 여러 가지 에러환경에 강인한 특성을 가지게 한 워터마킹 기법이다.

첫 번째, 저 주파수 영역에서의 Watermarking은 Cox에 의해 제안된 방법^[2]으로, 영상을 DCT한 후, n개의 가장 시각적으로 중요한 주파수 요소에 워터마크를 삽입하는 방법이다. 워터마크의 추출은 삽입된 워터마크를 되찾는 것이 아니라 단순히 삽입된 워터마크의 존재 여부를 문턱 치와 비교하여 판별하여 이뤄진다. 이 방법은 rescaling, JPEG, dithering, clipping, printing/scanning에 강인하나 항상 원 영상에 워터마크가 존재하는지를 확인하여야 하고, 넣을 수 있는 정보량이 너무 적다.

두 번째, 중간 주파수 영역에서의 Watermarking은 Barni, Swanson 등에 의해 제안된 방법^[3]으로, 영상을 8x8 단위로 나누고 각 8x8 단위 블록 별로 DCT를 수행한 후, DCT된 블록 중에서 무작위로 블록들을 선택하고 이 선택된 블록 내에 중간 주파수에 해당하는 부분의 상대적 크기가 1 또는 0을 나타내도록 계수를 바꾼다. 이 방법은 선택된 중간 주파수 영역에 해당하는 부분의 계수 사이의 변이가 작기 때문에 noise나 왜곡에 민감하다.

세 번째, 고 주파수 영역에 워터마크를 삽입하는 방법으로 Langelaar가 제안한 방법^[4]이다. 이 방법은 주파수 영역의 DCT 계수를 선택적으로 손상시켜 삽입하고 워터마크 비트는 고 주파수 영역의 DCT 계수가 제거된 DCT 블록의 패턴에 코딩된다. 고 주파수 영역에서의

워터마크 삽입 방법은 각종 압축에는 강인하나 시각적 저하를 막기 위해 cut-off point는 최소 값보다는 커야 하므로 워터마크 비트가 영상에 삽입되지 못하는 경우가 발생하고, noise 첨가에 민감하다.

III. 수정된 DCT 계수 기반의 Blind Watermarking Algorithm

3.1 적용된 DCT 알고리즘

네 가지 DCT Type(Type-II) 중 식(3-1)의 기저함수를 갖는 DCT-II를 사용하였다. DCT의 계산 중 코사인값의 계산은 실수 연산을 시행하여야 하며, 이러한 계산은 시간이나 하드웨어의 구성 측면에서 구현에 어려움이 많다. 따라서 식(3-3), (3-4)와 같은 DCT/IDCT 행렬식을 이용하여 룬 테이블을 구현하여 사용하였다.

$$X(u) = \sqrt{\frac{2}{N}} C(u) \sum_{i=0}^{N-1} x(i) \cos\left(\frac{(2i+1)u\pi}{2N}\right) \quad (3-1) \text{ DCT-II}$$

$$(C(0) = 1/\sqrt{2}, C(u) = 1 \text{ for } u \neq 0)$$

$$x(u) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} C(u) X(i) \cos\left(\frac{(2i+1)u\pi}{2N}\right) \quad (3-2) \text{ IDCT-II}$$

$$(C(0) = 1/\sqrt{2}, C(u) = 1 \text{ for } u \neq 0)$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} A & A & A & A & A & A & A & A \\ D & E & F & G & -G & -F & -E & -D \\ B & C & -C & -B & -B & -C & C & B \\ E & -G & -D & -F & F & D & G & -E \\ A & -A & -A & A & A & -A & -A & A \\ F & -D & G & E & -E & -G & D & -F \\ C & -B & B & -C & -C & B & -B & C \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \quad (3-3) \text{ DCT 계수 Matrix}$$

$$\begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \end{bmatrix} = \begin{bmatrix} A & D & B & E & A & F & C & G \\ A & E & C & -G & -A & -D & -B & -F \\ A & F & -C & -D & -A & G & B & E \\ A & G & -B & -F & A & E & -C & -D \\ A & -G & -B & F & A & -E & -C & D \\ A & -F & -C & D & -A & -G & B & -E \\ A & -E & C & G & -A & D & -B & F \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} \quad (3-4) \text{ IDCT 계수 Matrix}$$

$$x = \tilde{M} \cdot X$$

[표 3-1] DCT 계수 및 IDCT 계수

A	B	C	D	E	F	G
0.7071	0.9239	0.3827	0.9808	0.8351	0.5556	0.1951
X: DCT values			x: Input values			

식(3-3), (3-4)에서 순방향과 역방향의 DCT 계수와 행렬식은 행과 열의 순서만 바뀌어져 있으므로 하나의 룬 테이블을 사용하여 DCT, IDCT를 행할 수 있다.

3.2 Watermark 삽입 및 추출 방법

워터마크를 삽입하는 첫 단계는 원본 이미지의 각 블록을 DCT를 행하는 것이다. 두 번째 단계로 MxN 형태의 워터마크를 비트스트림 형태로 생성한다. 세 번째

단계는 워터마크 각각의 비트를 zig-zag 스캔 순서로 [그림-1]에서 표시된 부분의 DCT 계수의 LSB 3 비트에 삽입한다. 네 번째 단계는 워터마크가 삽입된 블록을 IDCT를 행하여 워터마크가 삽입된 영상을 만들어 낸다.

워터마크를 추출하는 과정은 워터마크가 된 이미지만을 가지고 진행된다. 추출과정은 워터마크가 된 이미지를 8x8 블록단위로 나누고 DCT를 수행한 뒤 삽입된 위치를 참조하여 각 위치에서 워터마크 비트를 추출해 내게 되고, 추출해 낸 값을 토대로 비트스트림을 만들어 낸 뒤 역 permutation을 하여 워터마크 이미지를 만들게 된다.

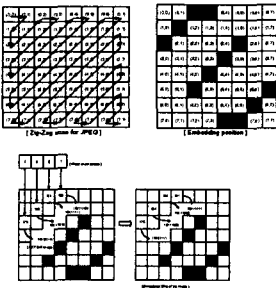


그림 1 Zig-Zag 스캔 방식과 워터마크 삽입 위치 및 방법

IV. Algorithm 검증

4.1 검증조건 및 결과고찰

본 논문에서는 워터마크가 삽입된 영상의 화질 열화에 대한 객관적인 평가에 식(4-1), (4-2)로 표현되는 PSNR을 사용하였다. 원 영상과 워터마크 된 영상의 차이가 적을수록, PSNR 값은 커진다는 것을 알 수 있다.

$$MSE = \frac{1}{n^2} \sum_{i=0}^n \sum_{j=0}^n (S_{ij} - C_{ij})^2 \quad (4-1)$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} [dB] \quad (4-2)$$

{ MSE(mean square error) , S_{ij} : 원 영상
 { PSNR(peak signal-to-noise) , C_{ij} : 워터마크된 영상

또 하나의 평가 기준은 워터마크 추출성능이다. 워터마크의 검출 비트 성공률은 워터마크가 삽입된 영상에 외부적 조작을 가한 후 워터마크를 검출하여, 삽입된 비트가 애러 없이 어느 정도의 비율로 검출될 수 있는지를 의미하며 NC(Normalized Correlation)으로 표현된다.

$$NC = \frac{\sum_{i=1}^N \sum_{j=1}^N M(i,j) * \hat{M}(i,j)}{\sqrt{\sum_{i=1}^N \sum_{j=1}^N M(i,j)^2 * \sum_{i=1}^N \sum_{j=1}^N \hat{M}(i,j)^2}} \quad (4-3)$$

M : 원본마크
 \hat{M} : 복원마크

그림 2 에서 원 영상과 워터마크가 삽입된 영상을 비교하였고, 두 영상간의 PSNR 값은 42.90[dB]로 나타났다.



그림 2 (a) 원 영상 (b) 워터마크 삽입된 영상 (c) 워터마크

워터마크가 삽입된 영상에 noise 삽입, cropping, JPEG 압축 등을 행한 후 워터마크를 추출하여 알고리즘을 검증한 결과를 그림 3, 4, 5 에, DCT를 적용한 기존 알고리즘과의 성능 비교를 [표 4-1], [표 4-2]에 나타내었다.

주위수대역	5	10	15
Low	0.75	0.65	0.60
Middle	0.75	0.66	0.60
High	0.80	0.70	0.65
PSNR	31.01	28.50	26.66

그림 3 Noise 삽입(5%, 10%, 15%) 후 워터마크 추출 결과

Lena	PSNR	14.24
Low	Middle	High
0.92	0.92	0.93

그림 4. Cropping 25% 후 추출된 워터마크(NC: 0.93)

	5	10	15	20	25	30
Lena	0.97	0.98	0.92	0.70	0.61	0.44
Middle	0.97	0.97	0.86	0.23	0.27	0.18
High	1.00	0.91	0.19	0.11	0.09	0.04
PSNR	42.66	39.66	35.82	33.20	31.29	28.95

그림 5. JPEG 압축 후 추출된 워터마크

[표 4-1] Huang' s 방법^[5]과 제안된 방법의 PSNR 비교

	Baboon	Lena	Peppers	Airplane
Huang' s Method	42.00	41.66	41.91	42.00
Proposed Method	42.93	42.91	42.88	42.81

[표 4-2] Wong' s 방법^[6]과 제안된 방법의 PSNR 비교

PSNR [dB]	Baboon	Peppers	Lena
	Proposed Method	42.93	42.91

V. FPGA 구현 및 검증

그림 6에서는 워터마크의 삽입블럭과 추출을, 그림 7에서는 DCT/IDCT 블럭의 구조와 동작흐름을 보여주고 있다.

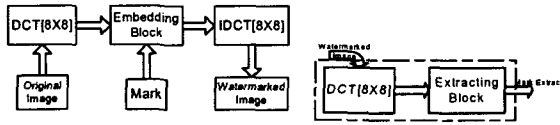


그림 6 (a) 워터마크 삽입블럭 (b) 워터마크 추출블럭

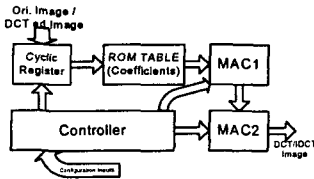
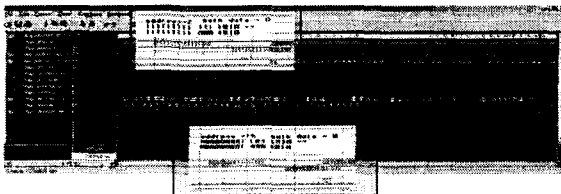
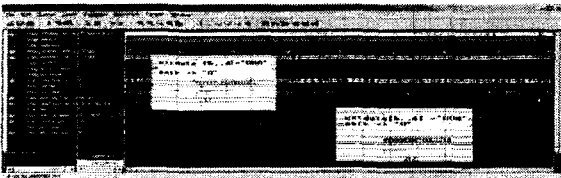


그림 7 DCT/IDCT 블럭구조

그림 8, 9에서는 VHDL 로 구현한 워터마크 삽입과정 및 추출과정에 관한 시뮬레이션 결과를, 그림 10은 FPGA 검증에 위해 사용된 PCI Board 를 나타낸다. (AMCC 5933 PCI 컨트롤러와 FPGA 3 만 게이트 탑재)



(a) 삽입과정 시뮬레이션 결과



(b) 추출과정 시뮬레이션 결과

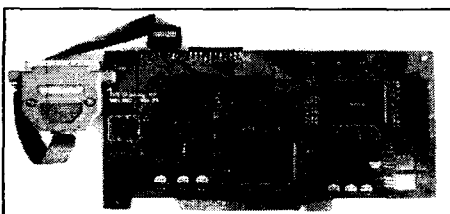


그림 10 검증에 사용한 PCI Board

VI. 결론

본 논문에서는 이산 코사인 변환을 이용한 주파수 영역에서의 워터마킹 알고리즘을 제안하였다. 기존의 복잡한 수식을 요구하는 방법과는 달리 원본 이미지의 DCT 계수의 LSB 값들에 워터마크를 삽입하는 간단한 방법을 적용하여 워터마크를 삽입하였다. 워터마크의 추출은 오직 워터마크가 삽입된 영상만으로 삽입된 워터마크를 추출하도록 하여 실용성을 갖게 하였다. 향후 과제로서 손실 압축이나 잡음 첨가, 크로핑 이외에 회전이나 다른 필터링 기법으로 변형된 영상에 대한 워터마크의 추출 방법에 대해 연구할 수 있을 것으로 본다.

VII. 참고문헌

- [1] G. Caronni, Schuba, C "Enabling Hierarchical and Bulk-Distribution for Watermarked Content", Computer Security Applications Conference, 2001. ACSAC 2001. Proceedings 17th Annual, 2001
- [2] I. J. Cox, M.L. Miller, J.A.Bloom, "Digital Watermarking" Morgan Kaufmann, 2001
- [3] Swanson, M.D.; Zhu, B.; Tewfik, A.H., "Robust data hiding for images" Digital Signal Processing Workshop Proceedings, IEEE, 1996
- [4] Langelaar, G. C.; Langendijk, R. L. "Optimal differential energy watermarking of DCT encoded images and video", Image Processing, IEEE Tran., Vol.10 Issue : 1, Jan. 2001
- [5] Jiwu Huang ; Yun Q. Shi; Yi Shi, "Embedding Image Watermarks in DC Components", IEEE Tran. Vol.10, 2000.
- [6] K. K. Wong; T. H. Lee; K. S. Ng; L. M. Cheng, " Adaptive Watermark Image", IEEE Tran. Vol. 43, 1997