

Error Correction Code를 이용한 워터마킹 방법과 성능분석

심혁재, 전병우

성균관대학교 전기전자 및 컴퓨터공학부

Watermarking Method using Error Correction Code and its Performance Analysis

Hiuk Jae Shim, Byeungwoo Jeon

School of Electrical and Computer Engineering, Sungkyunkwan University

E-mail : watnual@ece.skku.ac.kr, bjeon@yurim.skku.ac.kr

요약 : 영상에 워터마크를 삽입하는 것을 통신채널의 입장에서 해석한다면 워터마크는 신호로, 영상은 잡음으로 모델링이 가능하다. 따라서 이러한 잡음 속에서 신호에 대한 에러를 최소화하는 것이 워터마크의 추출을 최대화하는 것이라 할 수 있다. 통상적으로 Error Correction Code는 에러가 많은 통신채널에서 많이 이용되기 때문에 워터마킹 방법에서도 효과를 기대할 수 있다. 본 논문에서는 DCT 기반의 구간화 워터마킹 방법에 Turbo code를 이용하여 강인성 면에서의 향상된 성능을 실험 결과로 보이며, Turbo code의 해밍거리를 이용하여 워터마킹의 보다 효율적인 검출 방법을 제안한다.

1. 서론

최근 디지털 콘텐츠를 보호하기 위한 목적으로 워터마킹 기술은 방법면에서나 응용면에서나 다양하게 개발되고 있다. 워터마크의 삽입 영역도 공간 영역을 비롯하여 여러 다양한 변환 영역까지 넓게 연구되어지고 있다. 변환 영역 워터마킹 방법은 DCT 영역이나 Wavelet 영역, DFT 영역에서 주된 연구가 이루어지고 있으며, 각각의 영역에서의 강인성을 목적으로 새로운 알고리즘이 개발되어지고 있다. 이러한 강인성의 목적으로 최근에 많이 연구되어지는 부분 중 하나의 예로

기존의 워터마킹 방법과 ECC(Error Correcting Code)와의 접목하는 기술을 들 수 있다. 영상에 워터마크를 삽입하는 것은 통신채널에서의 신호와 잡음의 모델과 유사하다고 할 수 있기 때문에 반복부호 방식, 패리티 비트의 부가, CRC, BCH, RS, Viterbi, Turbo code등 다양한 ECC를 워터마킹 방법에 이용할 수 있다. 따라서 복잡도나 성능, 워터마킹 방법과 효율적으로 적용할 수 있는 코드를 찾거나 개발하는 것으로 워터마크 추출에 성능 향상을 가져올 수 있다. ECC를 워터마킹에 적용한 몇가지 예를 기술하면, [1], [2]에서는 블록 당 반복적인 워터마크를 삽입하여 복호시 성능을 높이는 방법을 제안하였고, [3]에서는 RS 코드를 이용, 이미지의 인종과 변형된 부분을 복구하는 방법을 제안하였다. 또한, [4], [5]에서는 각각 웨이블릿 영역에서 확산대역 워터마킹 방법과 DCT 영역에서 CDMA기반 워터마킹과 Turbo Code를 이용하여 강인성을 높이는 방법을 제안하였다. 본 논문에서는 [2]에서 제안된 워터마크 삽입 방법과 Turbo code를 통합하여 사용할 경우, 성능이 향상됨을 보이며, Turbo code에 해밍 거리를 이용한 결정법을 추가적으로 이용하여 워터마크 추출 성능을 높이는 방법을 제안한다.

2. 워터마크의 삽입과 추출

[2] 에서 제안하는 방법은 양자화, 혹은 구간화를 이용한 워터마킹이라 할 수 있다. 원 영상이 8x8 블록 DCT 처리된 후, 각각의 DCT 계수를 구간 계수로 대응시키는 방법으로 계수를 가까운 대표 계수 중 삽입 정보와 같은 값의 대표치를 갖는 대표 계수로 대응시킨다. 즉, 각각 구간 간격 Δ 의 배수는 0, 혹은 1을 대표하게 된다. DCT 계수 $F(u, v)$ 를 워터마크 wm 에 따라 삽입했을 때, 워터마크가 삽입된 계수를 $F'(u, v)$ 라 하면 다음과 같이 표현할 수 있다.

$$F'(u, v) = \begin{cases} F_0 & \text{if } wm=0 \\ F_1 & \text{if } wm=1 \end{cases} \quad (1)$$

여기서 F_0 와 F_1 은 각각 $F(u, v)$ 와 가까운 구간대표치로 $m \in Z$ 인 $m \cdot \Delta$ 로 표현할 수 있는 계수를 말한다. 이러한 구간화 워터마킹 방법은 $F''(u, v)$ 를 변형된 계수라 할 때, 식 (2)를 만족하면 올바른 워터마크가 추출될 수 있는 장점이 있다.

$$F_{0or1} - \Delta/2 \leq F''(u, v) < F_{0or1} + \Delta/2 \quad (2)$$

여기서 F_{0or1} 는 F_0 와 F_1 중에 현재 $F''(u, v)$ 가 속해 있는 대표치를 말한다.

워터마크의 추출과정은 다음과 같이 요약할 수 있다. 워터마크가 삽입된 영상을 DCT 취한 뒤 i 번째 DCT 계수를 $g_i(u, v)$ 라 한다면 식 (3)과 같이 가까운 대표치로 대응시킨다.

$$G_i = \text{ROUND} \left\{ \frac{g_i(u, v)}{\Delta} \right\} \quad (3)$$

이렇게 대표치 G_i 를 얻은 뒤, 최종적으로 삽입된 워터마크는 식 (4)와 같이 추출될 수 있다.

$$wm'(i) = \text{mod}(G_i, 2) \quad (4)$$

3. ECC를 이용한 워터마크 삽입과 추출

[2]에서는 8x8 DCT 블록 당 1bit의 워터마크를 N 회 반복하여 삽입하고 추출시에는 N 개의 추출된 워터마크 0, 1 중에 1/2를 넘는 신호로 결정하는 간단한 반복 부호어 방법을 취하고 있다. 본 논문에서는 워터마크 삽입과 추출에 Turbo code를 이용하고, 여기서 나온 결과를 해밍 거리를 이용하여 성능을 높이는 방법을 제안한다.

3. 1. Turbo code와 해밍거리를 이용한 결정법

L 의 길이를 갖는 워터마크 $wm = (b_0, b_1, \dots, b_{L-1})$ 는 각각 k 의 길이를 갖는 codeword C 로 매핑이 되며, 매핑된 $C = (c_{0,0}, c_{1,0}, \dots, c_{k-1,0}; \dots; c_{0,L-1}, c_{1,L-1}, \dots, c_{k-1,L-1})$ 는 각 길이가 k bit의 sequence마다 0 혹은 1을 나타낸다. 또 0과 1을 나타내는 코드를 정할 때는 해밍거리가 가장 먼 코드를 부가하게 된다. 즉, $k = 3$ 이라 할 때, 생성 가능한 코드의 쌍은 $MAP = \{(0,0,0;1,1,1), (0,0,1;1,1,0), (0,1,0;1,0,1), (0,1,1;1,0,0)\}$ 이며, 이 중 하나의 쌍을 선택하여 $\{0,1\}$ 에서 k bit의 코드로 매핑하여 0으로 매핑된 코드는 MAP_0 , 1로 매핑된 코드는 MAP_1 로 정한다. 따라서 결과적으로 워터마크 $wm = 0$ 일 경우에는 MAP_0 로, $wm = 1$ 일 경우에는 MAP_1 로 표현할 수 있다. 이렇게 생성된 C 는 Turbo coding을 통하여 부호화되어 영상에 삽입되게 된다. 길이 k 의 코드 C 는 그림 1에서의 PAD에서 길이 $n-k$ 의 tail bit이 부가되어 n bit가 되며 각각 interleaver []와 convolutional encoder를 통해 부호화되어 최종적인 sequence가 된다.

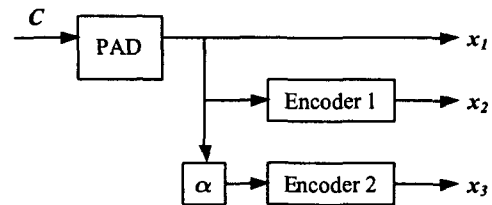


그림 1. Turbo encoder

따라서 그림 1의 Turbo encoder는 C 가 N bit일 때, Turbo encoding이 끝난 비트 스트림이 $3*L$ bit가 되므로 Rate 1/3 Turbo encoder라 할 수 있다. 일반적으로 code rate를 높이기 위해 puncturer를 이용하여 1/2 rate로 만들기도 하나 puncturing을 이용한 비트 스트림은 에러에 조금 약해지는 특성이 있기 때문에 본 논문에서는 puncturer를 추가하지 않았다. Decoding은 SOVA(Soft Output Viterbi Algorithm)를 이용하여 수행되며, 피드백을 사용하기 때문에 5~15의 iteration을 거치게 되는데, 일반적으로 5회 정도가 적당하다. Turbo Decoder의 output을 W 라 할 때,

$$W = (w_{0,0}, w_{1,0}, \dots, w_{k-1,0}, \dots; w_{0,l-1}, w_{1,l-1}, \dots, w_{k-1,l-1})$$

로 표현할 수 있다. 비트 스트림 W 는 처음 매핑했던 쌍 MAP_0 과 MAP_1 과의 비교를 통해서 최종적인 워터마크 wm' 를 추정하게 된다. i 번째 생성된 output bit stream을 워터마크로 추정하는 것은 식 (5)와 같이 표현할 수 있다.

$$wm'(i) = \min \{ d([w_{0,i}, w_{1,i}, \dots, w_{k,i}], MAP_0), d([w_{0,i}, w_{1,i}, \dots, w_{k,i}], MAP_1) \} \quad (5)$$

여기서 $d(x, y)$ 는 벡터 x 와 y 의 해밍 거리를 말하며, 이러한 과정을 통하여 Turbo decoding에서 정정되지 못한 부분을 추가적으로 보완하여 최종적 워터마크 데이터를 추정하게 된다.

3.2. 워터마크 삽입 영역

DCT 기반의 워터마킹 방법은 워터마크 삽입을 중간 주파수 대역으로 하는 것이 일반적이다. 저주파 대역은 강인성 면에서는 좋으나 화질 저하 및 워터마크의 가시화의 문제가 있고, 고주파 대역은 가시화의 문제는 적은 반면, 워터마크가 쉽게 깨어지기 쉽다. 하지만 워터마킹 방법에 ECC를 이용한다는 것으로 어느 정도의 에러는 정정될 수 있다는 특성을 이용하면 보다 더 강인함을 얻을 수 있게 된다. Turbo code는 interleaver를 사용함으로써 데이터를 일정한 패턴으로 재배열하여 군집 에러가 일어났을 경우, 한 블록 내의 에러를 다른 블록 등으로 분산시키는 역할을 하게 된다. 따라서 블록 내

에서 Random하게 interleaving을 해주는 것은 각각의 대역에 삽입할 코드를 고르게 분포시키는 것이라 할 수 있으며, 워터마크의 삽입영역을 기존에 비해 고주파 대역 방향으로 확장시킬 수 있는 장점이 있다. 본 논문에서는 그림 2에 표시된 영역에 워터마크를 삽입하였다.

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| DC | 1 | 5 | 6 | 14 | 15 | 27 | 28 |
| 2 | 4 | 7 | 13 | 16 | 26 | 29 | 42 |
| 3 | 8 | 12 | 17 | 25 | 30 | 41 | 43 |
| 9 | 11 | 18 | 24 | 31 | 40 | 44 | 53 |
| 10 | 19 | 23 | 32 | 39 | 45 | 52 | 54 |
| 20 | 22 | 33 | 38 | 46 | 51 | 55 | 60 |
| 21 | 34 | 37 | 47 | 50 | 56 | 59 | 61 |
| 35 | 36 | 48 | 49 | 57 | 58 | 62 | 63 |

그림 2. 워터마크 삽입 영역

4. 실험결과 및 성능분석

실험 영상은 256x256 lena를 사용하였으며, 각 블록 당 1 비트의 워터마크 정보를 삽입 총 1024 비트를 삽입하였다. 구간 간격 Δ 는 15를 사용하여 워터마크가 삽입된 이미지를 JPEG Quality factor 10에서 90까지 압축율을 변화시켰을 때 워터마크 추출을 BER로 나타내었다. 실험의 대상은 각각 블록 내에 30회를 반복시킨 반복 부호와 Turbo code를 이용한 방법, 그리고 Turbo code와 해밍 거리를 이용하여 워터마크를 결정하는 방법의 3가지 방법을 비교하였다.

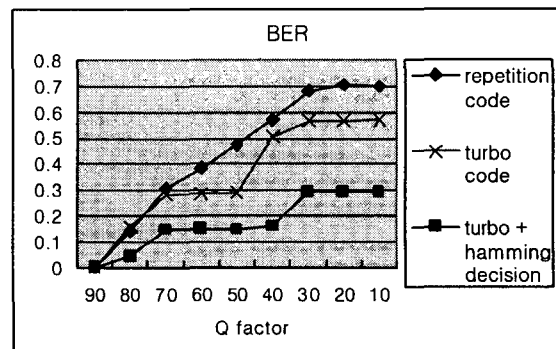


그림 3. 추출된 워터마크의 BER

그림 3에서 반복부호 방식으로 워터마크를 삽입한 방법보다 Turbo code를 이용한 방법이 워터마크 추출에 있어 BER을 향상시키는 것을 볼 수 있다. 하지만 Quality factor가 50보다 낮아지면서 워터마크 추출에 큰 효과를 주지 못하는 것을 볼 수 있다. 하지만 해밍거리에 의한 워터마크 결정방법을 Turbo code에 부가적으로 이용함으로써, Turbo code를 통해서도 정정되지 못한 부분을 대략 10~20% 정도 정정할 수 있다.

5. 결론 및 향후 연구 과제

본 논문에서는 [2] 에서 사용된 DCT 기반의 구간화 워터마킹 방법에 ECC 중 Turbo code를 이용하여 워터마크 추출에 있어서의 성능 향상을 보였다. 하지만 구간화 워터마킹 방법은 본래 연속 워터마킹 방법이었던 때문에 Turbo code를 이용해서도 큰 효과를 얻지 못하는 것을 볼 수 있었으나, 제안한 해밍거리를 이용한 결정법을 접목하여 Turbo code의 성능이 향상됨을 보였다. 하지만 Turbo code 자체가 가지고 있는 복잡도가 상당히 크기 때문에 이를 간소화하는 방법이 필요하다. 이러한 복잡도는 Turbo decoding 시 반복적인 피드백을 이용하여 soft decision을 내리는 부분에서 가장 크게 나타나며, 워터마킹에 이용하기 위해서는 복잡도를 낮출 수 있는 방법의 연구가 필요하고, 또한 Turbo code는 additive Gaussian 분포를 갖는 잡음 환경에 강한 반면, 워터마킹의 채널은 단순한 additive 뿐 아니라, multiplicative의 성격 등 여러 다른 성질을 가지는 환경이기 때문에 code 자체를 워터마킹 채널에 적합한 모델로 연구할 필요성이 있다. 또한 Turbo code 자체적인 interleaver 뿐 아니라, 생성 비트 스트림을 삽입할 저주파, 혹은 고주파 대역의 영역에 각각 다른 가중치를 주어 부가적으로 interleaving 기능을 부여하는 연구도 필요할 것이라 생각되며, 근본적으로 Turbo code를 기존 방법에 접목시키는 접근보다는 Turbo code의 성질을 이용하는 새로운 워터마킹 방법의 연구가 성능향상에 효과를 가져다 줄 것이라 생각된다. 따라서 이러한 부분들을 향후 연구 과제로 삼고 있다.

REFERENCES

- [1] C.-T. Hsieh et al., "A Study of Enhancing the Robustness of Watermark," *Multimedia Software Engineering 2000. Proceedings. International Symposium*, pp. 325-327, 2000.
- [2] 임재현, 심혁재, 전병우, "영상 변형 검출을 위한 사용자 key기반 Fragile 워터마킹," 대한전공학회 논문지 sp편, 9월, 2001.
- [3] J. Lee and C. S. Won, "A Watermarking Sequence using Parities of Error Control Coding for Image Authentication and Correction," *Consumer Electronics, IEEE Transactions*, Vol 46(2), pp. 313-317, May 2000.
- [4] N. K. Abdulaziz and K. K. Pang, "Performance Evaluation of Data Hiding System Using Wavelet Transform and Error-control," *Image Processing, 2000. Proceedings. 2000 International Conference*, vol.1, pp. 605-608, 2000.
- [5] A. Ambroze et al., "Turbo code protection of video watermark channel," *Vision, Image and Signal Processing, IEE Proceedings*, Volume: 148(1), pp. 54-58, Feb. 2001.