

프레임 움직임 예측을 이용한 워터마크 추출

이혜주*, 강현수*, 홍진우*

한국전자통신연구원 무선방송기술연구소 방송미디어부

Watermark Extraction Using Motion Estimation of Frames

Hye-Joo Lee*, Hun Su Kang*, Jin Woo Hong*

*Broadcasting Media Technology Department,

Electronics and Telecommunications Research Institute

hyejoo@etri.re.kr, hskang@etri.re.kr, jwhong@etri.re.kr

요약

비디오 워터마킹 기술은 비인지성, 안전성 및 강인성의 일반적인 요구조건 이외에 실시간 처리의 요구 조건들이 추가된다. 또한 공격에 있어서 프레임 정보를 이용한 워터마크 제거와 같은 공격이 가능하기 때문에 정지 영상과 비교하여 설계 시 많은 고려가 요구된다. 본 논문에서는 이전의 삽입 정보가 삽입된 비디오 데이터로부터 양극성 랜덤 계열을 이용하여 워터마크를 추출하는 경우에 비디오 프레임 간의 움직임 예측에 의해 구성된 프레임을 이용하여 추출 성능을 향상시키기 위한 추출 방법을 제안하였다. 제안 방식의 실험 결과로부터 움직임 예측을 이용하지 않은 결과와 비교하여 워터마크 추출 성능이 향상되었음을 확인하였다.

1. 서론

디지털 콘텐츠의 이용 증대와 함께 콘텐츠 보호에 관한 관심이 점차 증대되고 있다. 현재 콘텐츠 보호를 위한 연구의 하나인 디지털 워터마킹은 콘텐츠를 보호하기 위해 워터마크(watermark)라고 하는 특정 정보를 콘텐츠에 삽입하여 콘텐츠의 불법 복사, 저작권 보호 등에 이용하게 된다[1].

디지털 워터마킹 기술의 요구조건은 응용에 따라 차이가 있으나 일반적으로 비인지성(imperceptibility), 안전성(security) 및 강인성(robustness)을 만족해야 한다. 비디오 워터마킹은 이러한 요구 조건에 삽입 및 추출 시에 실시간 처리의 조건이 추가되고 다수의 프레임 상에 워터마크가 삽입되기 때문에 프레임 공격이 가능하다. 따라서 비디오 워터마킹 기술은 정지 영상을 대상으로 워터마킹 기술에 실시간성, 프레임 공격 등과 같은 제약이 추가되고 이러한 점을 고려하여 워터마크 삽입/추출 알고리즘을 설계하여야 한다.

본 논문에서는 비디오 워터마킹 기술로써 워터마크 추출

시에 비트 에러를 감소하기 위해 프레임 간의 움직임 예측으로부터 임의의 프레임을 구성하고 이 프레임을 워터마크 추출시에 이용하는 방법을 제안한다. 논문의 구성은 다음과 같다. 먼저 2장에서 간단히 비디오 워터마킹 기술에 관하여 소개하고, 3장에서는 워터마크 삽입 과정과 워터마크 추출 과정에서 어떻게 움직임 예측을 이용하는가에 대하여 기술한다. 4장에서는 다양한 실험 결과로부터 제안 방식의 효율성을 확인하고, 마지막 결론으로써 향후 연구 과제를 5장에 기술한다.

2. 비디오 워터마킹 기술

비디오 워터마킹 기술은 비인지성, 안전성, 강인성의 일반적인 요구 사항이외에도 복잡도(complexity), 압축 도메인에서의 처리(compressed domain processing), 고정 비트율(constant bit-rate) 등의 요구사항이 필요하다[2]. 특히 실시간 처리 또는 하드웨어 설계의 용이성과 관련하여 복잡도는 가능한 낮아야 한다. 그러나, 복잡도의 요구사항은 비디오 워터마킹 기술의 응용에 따라 달라질 수 있다. 예를 들어 방송이나 모니터링의 경우에는 워터마크 삽입 및 추출이 가능한 빨라야 하지만, DVD에 이용되는 워터마킹 기술에서는 워터마크 추출이 빨라야 한다.

비디오 워터마킹 기술은 크게 그림1과 같은 3가지 형태로 분류될 수 있다[3].

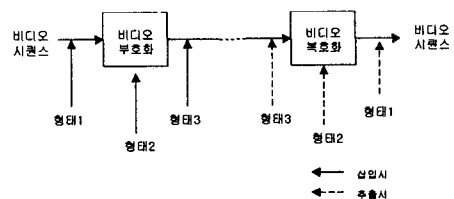


그림1. 비디오 워터마킹 분류

첫 번째 방식은 원 신호 레벨에서 워터마크를 삽입하고 추출하는 형태[4,5]이며, 두 번째 형태는 압축된 비트 스트림의 부분적인 복호를 수행하고 워터마크를 삽입한 후 재부호화하는 방식이다[2,6]. 마지막 형태는 압축 비트 스트림 자체에 워터마크를 삽입하는 것이다[6]. 첫 번째 방식은 압축된 비디오 데이터에 워터마크를 삽입하기 위해서는 전체 비디오 데이터를 복호해야 하는 문제가 있으며, 두 번째 방법은 특정한 압축 형태에 의존하는 단점이 있으며, 세 번째 방법은 워터마크의 제거가 쉽다는 문제가 있다. 이와 같은 방식은 다양한 비디오 압축 포맷을 대상으로 하는 경우에는 첫 번째 방식이 적합하지만, 제한적인 환경에서 이용하는 경우에는 두 번째 방식이 유리할 수 있다. 또한 세 번째 방식은 비디오 데이터에 대한 라벨링이 요구될 때 이용가능하기 때문에, 응용에 따라 적합한 방식을 선택할 수 있다.

본 논문에서는 첫번째 방식으로써 비디오 데이터의 푸도 성분(Y 성분)에 DCT를 적용하고 이진 비트로 구성된 삽입 정보를 워터마크로써 양극성(bipolar) 랜덤 계열을 이용하여 DCT 계수를 변경하여 삽입한다. 정확한 워터마크 추출을 위한 가장 효과적인 방법은 워터마크가 삽입된 영상에서 원 영상을 감산하는 것이지만 워터마크 추출 시 원 영상을 이용하는 것은 효율성이 떨어지며 응용 분야도 한정된다. 따라서, 원 영상을 예측하여 추출 시에 이용하는 것이 효과적이다. 이 점을 고려하여 비디오 데이터는 프레임 간의 상관성이 높아 프레임간의 예측을 이용하여, 추출 성능을 향상시키는 방법을 다음 장에서 기술한다.

3. 워터마크 삽입 및 추출

먼저, 일반적인 워터마크 삽입 및 추출 방식에 대하여 아래에 기술한다.

3.1. 기본 워터마크 삽입 및 추출

기본적인 삽입 알고리즘은 Hernandez 등에 의해 제안된 방법을 이용하여 다음과 같이 삽입한다[7]. 워터마크를 삽입하기 먼저 $M \times N$ 크기의 t 번째 프레임을 8×8 블록으로 분할하고 DCT를 수행하여 각 블록 내의 동일한 주파수 위치 (u, v) , $0 \leq u, v < 8$ 에 있는 DCT 계수 $f(u, v)$ 를

$$X_{t,i}(m, n) = f_{m,n}(u, v) \quad (1)$$

와 같이 그룹화한다. 이때, $i = u/8 + (v \bmod 8)$, $0 \leq i < 64$ 와 (m, n) , $0 \leq m < M/8$, $0 \leq n < N/8$ 은 그룹과 블록의 위치를 의미한다. 이와 같이 그룹을 구성한 후 1비트를 그룹에 삽입하게 된다. 따라서, 1프레임에 최대 워터마크 비트 수는 64비트가 된다. 64비트 워터마크 $b = (b_0, \dots, b_{63})$ 의 1비트 $b_i = \{-1, 1\}$ 는 랜덤 계열 $S = \{s(k) | k = (m, n)\}$

$s(k) = \{-1, 1\}$ 에 의해 $W_{t,i} = b_i \cdot S = (b_i s(0), b_i s(1), \dots)$ 을 구성하여

$$Y_{t,i}(m, n) = X_{t,i}(m, n) + \alpha w_{t,i}(m, n) \quad (2)$$

와 같이 그룹에 가산한다. 이때, α 는 워터마크 강도를 나타내고 랜덤계열의 크기는 $|S| = D = \lfloor M/8 \rfloor \times \lfloor N/8 \rfloor$ 이다.

변경된 각 그룹의 DCT 계수들은 원래의 블록의 위치로 재구성하고 각 블록에 IDCT를 수행하면 워터마크가 삽입된 프레임울 얻을 수 있다.

워터마크의 추출은 대역확산 기법에 의해 랜덤 계열 S 와 워터마크가 삽입된 각 프레임간의 상관값을 이용한다. 워터마크 삽입 시와 마찬가지로 프레임울 블록으로 분할하고 64개의 그룹을 구성한다. 워터마크 삽입 비디오에 대한 공격을 고려하지 않는 경우, 워터마크 비트 b'_i 는 $Y_{t,i}$ 과 랜덤 계열 S 를

$$r_i = \langle Y_{t,i}, S \rangle \quad (3)$$

와 같이 계산한 결과로부터

$$b'_i = \begin{cases} 1, & \text{if } r_i \geq 0 \\ -1, & \text{otherwise} \end{cases} \quad (4)$$

와 같이 결정된다.

3.2. 워터마크 추출을 위한 움직임 예측의 이용

예측 프레임을 구성하여 워터마크 추출 시에 이용하기 위해 블록 매칭 기법으로 움직임 예측을 수행한다. t 번째 프레임에 대하여 블록으로 분할하여 움직임 벡터 $(0, 0)$ 을 기준으로 이전 프레임과 이후 프레임 중에서 가장 유사한 블록을 선택하여 예측 프레임 t' 프레임을 구성하고, t 번째 프레임과 예측 프레임의 DCT 계수들을 각각 $Y_{t,i}$ 과 $\bar{Y}_{t',i}$ 으로 그룹화한다. 이때, $e_{t,i} = Y_{t,i} - \bar{Y}_{t',i}$ 라고 할 때 워터마크 비트 b'_i 의 추출은

$$r_2 = \langle e_{t,i}, S \rangle \quad (5)$$

를 계산하여 식(4)와 유사한 형태로 결정된다.

여기서, 워터마크는 동일한 위치에 삽입되었기 때문에 움직임 벡터가 $t+1$ 번째 프레임에 대해 $(0, 0)$ 인 경우에

$$\bar{Y}_{t',i} = Y_{t+1,i} = X_{t+1,i} + W_{t+1,i} \text{ 이므로}$$

$$\begin{aligned} r_2 &= \langle e_{t,i}, S \rangle \\ &= \sum_{k=(m,n)} (X_{t,i}(k) - X_{t+1,i}(k))s(k) + b_i \alpha \sum_{k=(m,n)} (s(k)^2 - s(k)) \end{aligned} \quad (6)$$

와 같이 워터마크가 제거되어 버리게 되어 워터마크 추출 에러가 발생된다. 따라서, 움직임 예측을 이용하기 위해서는

i 번째 프레임에 각 그룹에 삽입되는 워터마크 b_i 는 $i+1$ 번째에서는 $\bar{b}_i = -b_i$ 로 반전을 수행한다. 이와 같이 워터마크 비트의 반전을 통하여

$$r_2 = \sum_{k=(m,n)} (X_{i,i}(k) - X_{i+1,i}(k))s(k) + b_i \sum_{k=(m,n)} s(k)^2 - \bar{b}_i \sum_{k=(m,n)} s(k)^2$$

$$= \sum_{k=(m,n)} (X_{i,i}(k) - X_{i+1,i}(k))s(k) + 2b_i \alpha \sum_{k=(m,n)} s(k)^2 \quad (7)$$

와 같이 워터마크의 제거를 방지할 수 있다. 추출 시에 위상이 서로 다른 2개의 계열을 원래의 워터마크로 복구하기 위해서는 판단비트를 DC 계수의 그룹에 삽입하여 비트의 값이 "1"인 경우에는 원래의 계열이고 "0"인 경우에는 반전된 계열로 판단하여 추출된 워터마크 비트를 원래의 워터마크로 복구할 수 있다.

지금까지 언급한 움직임 추정 방법은 그 다음 프레임으로부터의 현재 프레임 추정 (backward ME)을 중심으로 기술하였으나, 이전 프레임으로부터의 현재 프레임 추정 (forward ME)도 함께 포함하여 양방향 추정 (bi-directional ME)를 수행하여 추정치의 오차가 적은 추정치를 사용함으로써 성능을 더욱 개선할 수 있다.

3.3. 비트 에러 확률

D 가 그룹의 크기, μ_x 와 σ_x^2 는 $X_{i,i}$ 의 평균과 분산, $E_w = \alpha D$ 일 때, r_1 의 평균 및 분산이 $\mu_1 = E_w$, $\sigma_1^2 = E_w(\mu_x^2 + \sigma_x^2)$ 일 때 예측 프레임을 사용하지 않는 경우의 1비트 에러 확률은

$$P_1 = \Pr(r_1 < 0 | b_1 = 1) = Q\left(\frac{\mu_1}{\sigma_1}\right) \quad (8)$$

이 된다. 이와 비교하여 예측 프레임을 이용하는 경우에 그룹 $X_{i+1,i}$ 와 $X_{i,i}$ 의 공분산 (covariance)이 $r(n) = \sigma^2 \rho^{|n|}$, $|\rho| \leq 1$ [8], $X_{i+1,i} - X_{i,i} = \varepsilon_{i,i}$ 이라고 하면, r_2 의 평균과 분산은, (0,0)의 움직임 벡터의 경우, $\mu_2 = 2E_w$, $\sigma_2^2 = E_w(1 - \rho^2)(\mu_x^2 + \sigma_x^2)$, (0,0)의 움직임 벡터가 아닌 경우, $\mu_2 = E_w$, $\sigma_2^2 = E_w(1 - \rho^2)(\mu_x^2 + \sigma_x^2)$ 이 되어 비트 에러 확률은

$$P_2 = \Pr(r_2 < 0 | b_2 = 1) = \begin{cases} Q\left(\frac{2\mu_2}{\sigma_2}\right), & mv = (0,0) \\ Q\left(\frac{\mu_2}{\sigma_2}\right), & mv \neq (0,0) \end{cases} \quad (9)$$

따라서, $P_1 > P_2$ 로 예측 프레임을 이용하지 않는 경우의 비트 에러 확률이 이론적으로 높음을 알 수 있다. 식(8)과 식(9)에 나타난 Q 함수의 내부의 값이 다음과 같은 관계가 있기 때문이다.

$$E_w / \sqrt{(1 - \rho^2)(\mu_x^2 + \sigma_x^2)} > E_w / \sqrt{(\mu_x^2 + \sigma_x^2)} \quad (10)$$

여기서 ρ 는 현재 블록과 움직임 보상 블록의 DCT 상관 관계를 나타내므로, 움직임이 크지 않은 영상의 경우 큰 값을 가질 가능성이 크다.

4. 실험 및 결과

제안 방식의 실험을 위한 비디오 데이터로서 RGB(8bits/pixel)의 30프레임의 Grandmom(360x240), Tennis(352x240), Flower(352x240)을 이용하였다. 워터마크를 삽입하기 위해 RGB를 YUV로 변환시키고 휘도 성분인 Y 성분에 워터마크를 삽입한다. 그림2는 원 비디오 데이터와 워터마크 강도에 대해 30개의 프레임에 대해 워터마크의 삽입된 데이터와 원 데이터의 평균 PSNR로 양호한 화질을 보이고 있다.

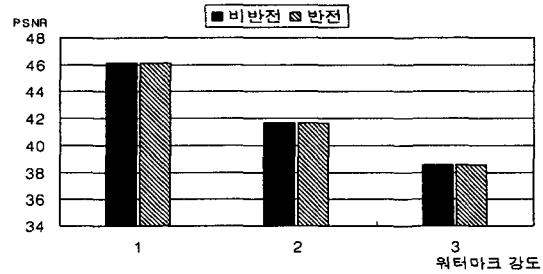


그림2. 워터마크 삽입 비디오의 PSNR

그림3은 워터마크된 비디오에 공격이 가해지지 않은 경우, 움직임 예측을 이용한 경우와 그 반대의 경우의 비트 에러율을 나타낸 그래프이다. Grandmom과 Tennis와 같이 움직임 적은 비디오는 워터마크 비트 에러율의 차이가 그다지 크지 않으나, 움직임 예측을 이용하는 경우에 $\alpha=1$ 로 설정하여도 비트에러가 발생되지 않았다. 또한 Flower와 같이 움직임이 많을수록 움직임 예측을 이용한 경우의 비트 에러율이 감소함을 실험적으로 확인할 수 있다.

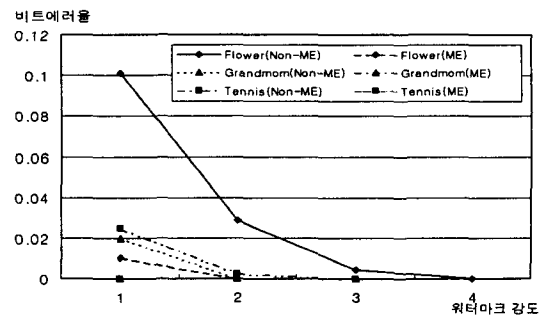


그림3. 워터마크 비트 에러율

4.1. MPEG-2 압축

그림4는 워터마크를 삽입한 후에 다양한 압축율로 MPEG-2 압축 후의 워터마크 추출 결과를 그래프로 나타낸 것이다. 압축율이 높을 수록 워터마크 추출의 비트 에러가 발생이 되지만, Flower를 강도=3과 강도=4로 각각 워터마크한 후 압축을 수행했을 경우에 강도가 높을수록 비트에러율이 감소됨을 알 수 있다.

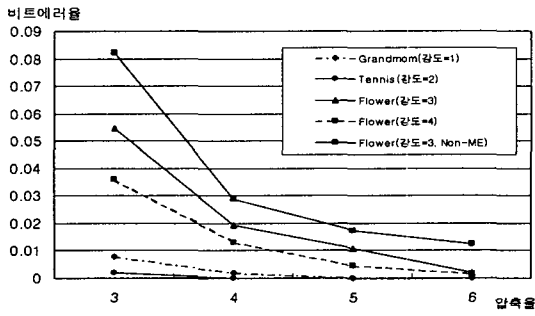


그림4. MPEG-2 압축 후의 비트 에러율

또한 움직임 예측을 사용하지 않을 경우(Flower, 강도=4, Non-ME)에 MPEG-2 압축 후의 비트 에러율이 높음을 알 수 있다.

4.2. 프레임 공격

프레임 공격이란 비디오 데이터의 프레임들을 이용하여 워터마크를 제거하거나 손상시키는 것으로 두 프레임간의 평균을 이용하는 프레임 평균(frame averaging), 프레임 제거(frame dropping)와 프레임 위치 변경(frame reordering) 등을 들 수 있다. 프레임 평균에 대한 실험으로 프레임간의 픽셀 값의 차가 임계값 이하일 때, 픽셀 값의 평균을 수행하는 경우 그림5와 같은 비트 에러가 발생하였다.

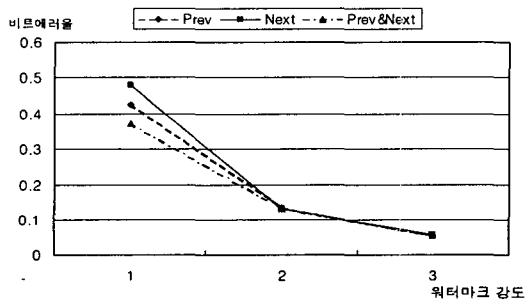


그림5. 프레임 공격후의 비트 에러율

각각 1) 이전 프레임과 현재 프레임(Prev), 2) 현재 프레임과 이후 프레임(Next), 3) 현재 프레임과 이전, 이후 프레임(Prev&Next)을 이용하여 프레임 평균 공격을 취한 결과로 비트 에러율이 많이 발생이 되지만 워터마크 강도를 증가시키면 비트 에러율이 감소하므로 적절한 워터마크

강도의 설정이 필요함을 알 수 있다.

5. 결론

비디오 워터마킹 기술은 정지 영상과 달리 실시간 처리가 가능하여야 하며, 워터마크 삽입 및 검출에 있어서 프레임 간의 정보를 이용할 수 있다는 특징이 있다. 따라서, 본 논문에서는 인접 프레임 간의 움직임 예측을 이용하여 워터마크 추출 시에 발생하는 워터마크 비트 에러를 감소시키는 방법을 제안하였다. 비디오의 특징에 따라 움직임이 많은 비디오에 대해서는 움직임 예측 방식을 이용하는 경우 워터마크 비트 에러가 감소됨을 알 수 있으며 MPEG-2 압축에 대한 강인성을 확인하였다. 그러나, 제안 방식은 프레임 공격에 대한 비교적 취약한 결과를 보여주었다. 이것은 프레임 공격에 의해 예측 프레임에 영향을 주기 때문이다. 따라서, 예측 프레임에 대한 공격의 영향을 워터마크 추출 시에 제거할 수 있는 방안의 연구가 필요하다.

[감사의 글]

본 논문은 정보통신부 지원 “디지털 콘텐츠 관리 기술 개발” 과제의 수행 결과의 일부로서 관계자분에게 감사의 글을 전합니다.

[참고문헌]

- [1] Katzenbeisser and Petitcolas etditors, Information Hiding: Techniques for steganography and digital watermarking, pp.95-215, ARTECH House, 2000
- [2] F. Hartung, B. Girod, “Watermarking of uncompressed and compressed video,” Signal Processing 66, pp.283-301, 1998
- [3] Tae-Yun Chung et. al., “Digital Watermarking for Copyright Protection of MPEG2 Compressed Video,” IEEE Trans. on Consumer Electronics, Vol. 44, No. 3, 1998
- [4] M. D. Swanson, B. Zhu, and A. H. Tewfik, “Multiresolution Scene-Based Video Watermarking Using Perceptual Models,” IEEE Journal on selected areas in communications, Vol. 16, No. 4, 1998
- [5] W. Zhu, Z. Xiong, and Y-Q. Zhang, “Multiresolution Watermarking for Images and Video,” IEEE Trans. on Circuits and Systems for Video Technology, Vol. 9, No.4, 1999
- [6] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, “Real-time Labeling of MPEG-2 Compressed Video,” Journal of Visual Communication and Image Representation, Vol. 9, No. 4, pp.256-270, 1998
- [7] J. R. Hernandez, M. Amado, and F. Perez-Gonzalez, “DCT-Domain Watermarking Techniques for Still Images: Detector Performance Analysis and a New Structure,” IEEE. Trans. on Image Processing, Vol. 9, No. 1, 2000
- [8] A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, pp. 190-193, 1989