

MPEG 동영상의 실시간 워터마킹 기법

김 경순, 임 재혁, 원 치선
 동국대학교 전자공학과 영상처리연구실

Real-time Watermarking Algorithm for MPEG Video

Gyong Soon Kim, Jae Hyuck Lim, Chee Sun Won
 Department of Electronics Engineering, Dongguk University
comster.jaehyuck, cswon@dongguk.edu

개 요

본 논문에서는 MPEG 비트스트림 기반의 동영상 저작권 보호를 위한 간단하면서도 효율적인 실시간 워터마킹 알고리즘을 제안한다. 제안된 알고리즘에서는 MPEG 부호화된 동영상에 대해 DCT 계수 영역으로 부분 복호화한 후 양자화 기법을 사용하여 DC 계수에 비가시성을 가지도록 워터마크를 삽입한다. 실험결과 제안된 알고리즘은 기존의 비트스트림 기반 워터마킹 기법보다 계산 복잡도 및 강인성 면에서 향상된 성능을 보여주고 있다.

1. 서 론

최근 들어 상업적 가치가 높은 VOD, DVD 등의 보급 확산으로 동영상에 대한 많은 디지털 워터마킹 기법들이 제안되어지고 있다. 동영상은 많은 데이터량을 가지기 때문에 MPEG과 같은 시간적 공간적인 중복성을 제거한 압축된 형태로 저장 및 전송되어진다. 동영상의 워터마킹 알고리즘은 동영상이 정지영상의 연속되는 신호로 생각되어질 수 있기 때문에 정지영상에서의 워터마킹 알고리즘을 그대로 동영상으로 확대 적용할 수가 있다. 그러나 정지영상의 워터마킹 알고리즘을 확대 적용시 계산 복잡도 문제가 발생되어진다. 대부분의 정지영상 워터마킹 알고리즘은 동영상에 실시간 또는 빠른 시간 내에 워터마크의 삽입 및 검출이 이루어져야 하는 응용에 대해서는 확대 적용이 거의 불가능하다고 할 수 있다. 비트스트림으로 저장되어져 있는 동영상에 대해서는 복호화를 한 후 워터마크를 삽입해야 하기 때문에 계산 복잡도는 더욱 증가하게 된다. 그래서 비트스트림 기반의 동영상에 대해 실시간 응용을 위한 계산 복잡도

가 낮은 워터마킹 알고리즘의 개발이 필요하다.

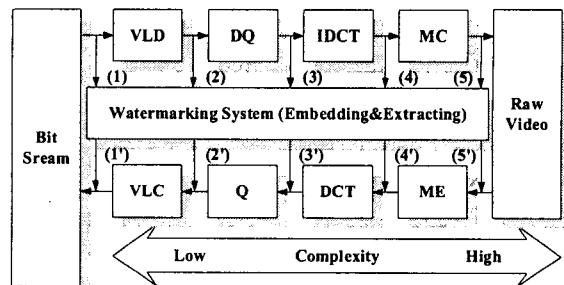


그림 1 워터마킹 영역에 따른 계산 복잡도

그림 1은 비트스트림 기반의 워터마킹 알고리즘의 가능한 워터마킹 영역을 나타내고 있으며 복호화 정도에 따라 워터마킹 알고리즘의 계산 복잡도가 커짐을 나타내고 있다. 가급적이면 시간 소모적인 IDCT와 움직임 보상 같은 공정은 피해서 최소한의 복호화를 한 후 워터마크를 삽입하는 알고리즘을 개발하는 것이 바람직하다. 동영상 비트스트림 워터마킹 알고리즘 개발시 낮은 계산 복잡도 외에 충족해야 될 몇 가지 요구조건들이 있다. 워터마크 삽입 후 비트율이 증가하거나 인간의 시각에 인지되어서는 안되며, 상업적인 가치를 떨어뜨리지 않는 비의도적인 영상처리에 대해 워터마크를 검출할 수 있어야 한다. Hartung[1]은 MPEG 동영상에 대해 상관관계 기반의 워터마킹 알고리즘을 제안했다. 실시간에 근접한 빠른 알고리즘으로 I, P, B 모든 프레임에 워터마크를 삽입하고 있다. 모든 프레임에 워터마크를 삽입하고 있기 때문에 워터마크 삽입에 의한 에러가 그 프레임을 참조하는 프레임으로 전파되는데 이를 막기 위해 에러 보상(Drift Compensation)을 해주고 있다

이는 알고리즘의 계산 복잡도를 증가시키는 요인이 된다. Langelaar[2]는 임의의 DCT 블록들간의 고주파 성분의 에너지 차를 만들어 실시간으로 워터마크를 삽입하는 DEW 알고리즘을 제안했다. 이 알고리즘은 인트라 프레임에만 워터마크를 삽입하고 있기 때문에 에너지 누적이 되지 않는다. 알고리즘이 두 영역간에 일정 에너지 차를 만들기 위해 한 영역의 고주파 성분의 AC 계수를 버리는 기법이기에 때문에 경계선이 있는 영역에서 흐려짐이 발생하는 문제점이 있다.

본 논문에서는 기존의 기법보다 계산 복잡도 및 성능이 개선된 MPEG 비트스트림 기반의 실시간 워터마킹 기법을 제안한다. 제안된 알고리즘은 양자화 기법을 사용하여 DC 계수에 직접 워터마크를 삽입하는 방법으로 실시간 처리가 가능하며 복호화된 공간영역에서도 워터마크를 검출할 수 있다는 특징을 가진다. 본 논문의 구성은 2장에서 제안된 알고리즘에 대해서 서술하고 3장에서 제안된 방식의 성능을 실험을 통해서 DEW 알고리즘[2]과 비교 평가한다. 마지막으로 4장에서 본 논문의 결론을 맺는다.

2. 제안된 알고리즘

본 논문에서는 워터마크를 삽입하기 위해서 DCT DC 계수를 조작한다.

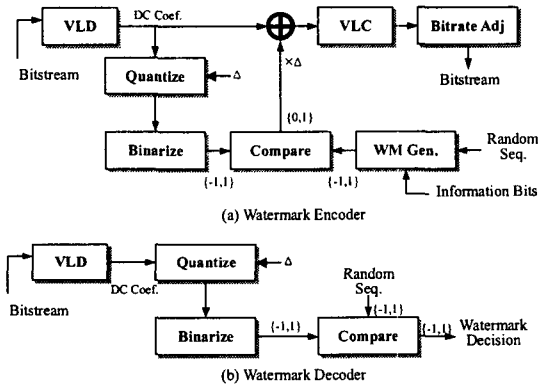


그림 2 워터마킹 알고리즘 블록도

MPEG 인트라 블록에서의 DC 계수는 가변장 부호화 하기에 너무 큰 값이기 때문에 가변장 부호화 테이블을 이용하는 보통의 계수들과는 다른 방법을 이용하여 부호화된다. 절대 크기의 부호화가 아닌 전 블록의 해당되는 값과 비교해서 그 차이값만을 부호화한다. 인트라 블록에서의 DC 계수는 AC 계수와 독립적으로 부호화되

기 때문에 이를 이용하여 워터마크를 삽입할 경우 간단 하면서 효율적인 워터마크를 삽입할 수가 있게 된다. 이는 [3]에서 워터마크의 강인성을 개선하기 위해 DC 계수를 사용하여 비가시적으로 워터마크를 삽입할 수 있음을 잘 보여주고 있다.

그림 2는 제안된 알고리즘의 워터마킹 삽입 과정 및 검출 과정에 대한 블록도이다. 블록도에서 제안된 알고리즘이 매우 간단함을 보여준다. 비트스트림으로부터 복호화된 k번째 DCT 블록의 DC 계수 $F_k(0,0)$ 를 다음 수식에서처럼 $F'_k(0,0)$ 로 변경함으로써 워터마크를 삽입한다.

$$F'_k(0,0) = \begin{cases} F_k(0,0), & \text{if } B(Q_\Delta(F_k(0,0))) = p_i w_j \\ F_k(0,0) + \Delta, & \text{if } B(Q_\Delta(F_k(0,0))) \neq p_i w_j \end{cases} \quad (1)$$

$F'_k(u,v)$: the (u,v) 'th watermarked coefficient of k 'th block
 $F_k(u,v)$: the (u,v) 'th original coefficient of k 'th block
 p_i : the i 'th element of pseudo-random sequence
 w_j : the j 'th element of information sequence

여기에서 $Q_\Delta(F_k(0,0)) = T\left[\frac{F_k(0,0)}{\Delta}\right]$, $T[\]$ 는 소수점 이하 버림 연산이다. 그리고 Δ 는 양자화 스텝이고 $B(\beta)$ 는 다음과 같이 이진화하는 함수이다.

$$B(\beta) = \begin{cases} 1, & \text{if } \beta \bmod 2 = 0 \\ -1, & \text{if } \beta \bmod 2 = 1 \end{cases}$$

식 (1)에서 양자화 스텝 Δ 로 양자화된 DC 계수를 이진화한 후 그 값이 변조된 워터마크 정보와 동일하면 원래 DC 계수를 변경하지 않고 그대로 두고 그렇지 않으면 두 값이 동일해지도록 양자화 간격 Δ 를 더해줌으로써 워터마크가 삽입됨을 보여준다. 워터마크 삽입 후에는 가변장 부호화한 부호 길이가 증가하지 않도록 고주파 성분의 DCT AC 계수를 이용하여 비트율을 조절한다. 비트율 조절은 프레임 단위로 원영상 프레임의 비트율에 비해 증가한 경우 현재 워터마크가 삽입되어 지고 있는 블록의 AC 계수를 구하고 그 AC 계수 중 EOB(End Of Block)전의 최고주파 성분의 AC 계수를 무시함으로써 비트율이 증가하지 않도록 한다.

워터마크 검출 과정은 그림 2의 검출 블록도와 다음 수식에서 보듯이 삽입 과정과 거의 동일하다.

$$w_j = \begin{cases} 1, & \text{if } B(Q_\Delta(F'_k(0,0))) = p_i \\ -1, & \text{if } B(Q_\Delta(F'_k(0,0))) \neq p_i \end{cases} \quad (2)$$

검출 과정에서는 양자화된 DC 계수에 이진화하여 워터마크 삽입시 사용되었던 랜덤 시퀀스와 비교해서

동일하면 워터마크가 '1'로 결정하고 그렇지 않으면 '-1'로 결정한다.

제안된 알고리즘에서 워터마크를 I 프레임의 DCT DC 계수에 삽입하고 있기 때문에 문제점으로 영상에 대한 왜곡 현상으로서 블록킹 현상이 발생할 수 있고 프레임 삭제 등의 의도적인 워터마킹에 대한 공격에 대해 취약성을 보일 수 있다는 것이다. 전자의 경우 작은 값의 양자화 스텝의 사용과 DCT 블록 단위의 워터마킹 기법에서 매크로 블록 단위의 워터마킹 기법으로 확장함으로써 충분히 억제되어질 수 있지만, 후자의 경우는 용적량(Payload)과도 관련이 있기 때문에 반드시 P, B 프레임으로 확장함으로써 해결되어야만 한다. 완전한 에리 보상이 아닌 움직임 벡터를 이용한 참조 블록의 DC 기여도 즉 평균 기여도 등을 계산해서 적절한 화질을 유지하는 범위내에서 보상해주고 워터마크를 삽입한다면 충분히 실시간내에서 가능하다.

3. 실험 결과

본 논문의 실험은 8Mbps로 코딩된 352*240 450F Flowers의 3개의 동영상에 대해 실험을 진행하였고 I-Frame의 휘도 성분에만 워터마크를 삽입하였다. 제안된 알고리즘의 워터마크 삽입 및 검출 소요시간은 표준 복호화기 복호화 시간의 약 1/8정도이며 기존의 실시간 워터마킹 알고리즘의 1.3[1], 1/2[2]에 비해 매우 빠르다는 장점을 가진다. 실험결과에서 제안된 알고리즘이 기존의 압축 동영상의 실시간 워터마킹 기법인 DEW 알고리즘[2]보다 나은 특성을 보여주고 있다. 특히 그림 5에서 볼 수 있듯이 트랜트코딩에서 매우 우수한 특성을 보여주고 있으며 공간영역에서의 잡음 첨가(그림 4) 및 포맷 변환(그림 6)등의 공격에 대해서도 강인성을 보여주고 있다. 복호화된 DC 계수에 워터마크를 삽입할 경우 가장 큰 문제점으로 지적되는 블록킹 현상은 영상의 복잡영역에서는 문제가 되지 않았고 평탄(monotone) 영역에서 발생되어지는 블록킹 현상은 작은 양자화 스텝의 사용과 현재의 블록단위의 워터마크 삽입에서 매크로 블록단위 워터마크 삽입으로 확장했을 경우 원본 영상 없이는 인지되지 않았다(그림 3). 여기서 양자화 스텝은 제안된 워터마킹 기법의 강인성과 비인지성 그리고 용적량간의 최적화(Trade-off)를 위한 인자 역할을 한다.

4. 결론

본 논문에서는 DCT DC 계수에 워터마크를 삽입하는 간단하고 효율적인 MPEG 비트스트림 기반의 동영상 워터마킹 기법을 제안하였다. 워터마크 삽입 및 검출이 최소한의 복호화만을 진행한 후 이루어지기 때문에 기존의 실시간 워터마킹 기법들보다 훨씬 낮은 계산 복잡도를 가진다. 또한 제안된 기법은 재부호화, 트랜스코딩, 포맷변환등과 같은 비의도적인 워터마크 공격에 대해 강인하다는 것을 실험결과에서 보여준다.

5. 참고 문헌

- [1] Frank Hartung and Bernd Girod, *Watermarking of uncompressed and compressed video*, Signal Processing, vol.66(3) p283--301, May 1998.
- [2] G.C. Langelaar, *Real-time watermarking techniques for compressed video data*, Ph.D. dissertation, Delft University of technology, The Netherlands, Jan. 2000.
- [3] Jiwu Huang, Yun Q. Shi, and Yi Shi, *Embedding Image Watermarks in DC Components*, 974 IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No.
- [4] Changryoul Choi, Jechang Jeong, *Watermarking Method in the Spatial Domain Using Block-based Quantization and Jacobian Symbols*, Chunchon Multimedia Conference, Jun. 2001

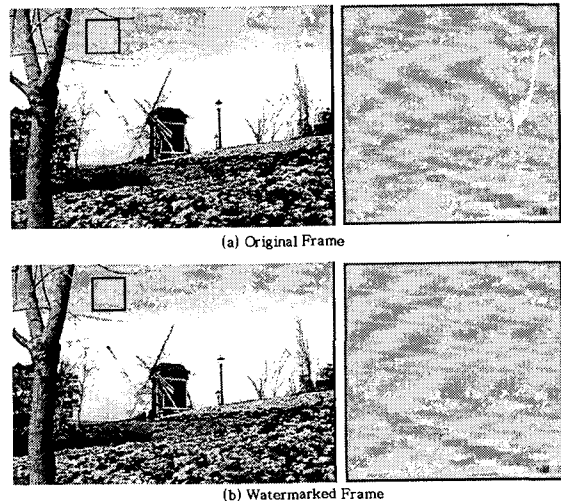


그림 3 워터마크 삽입 전후 영상

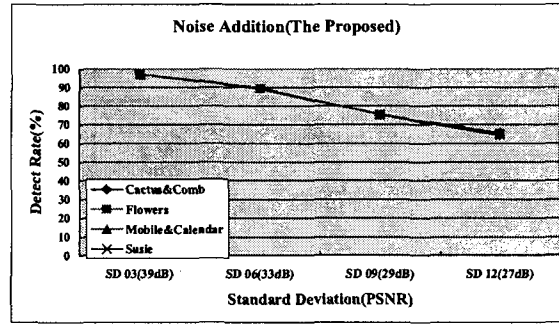
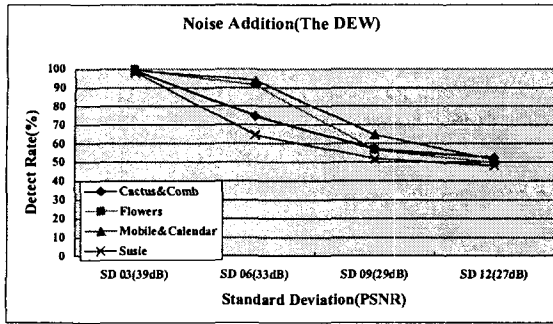


그림 4 실험결과(Noise Addition)

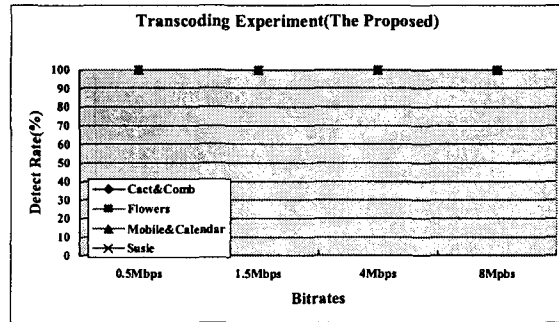
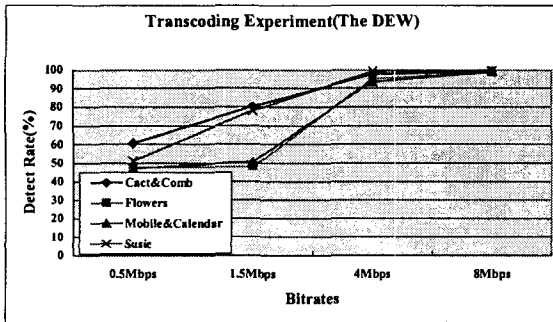


그림 5 실험결과(Transcoding)

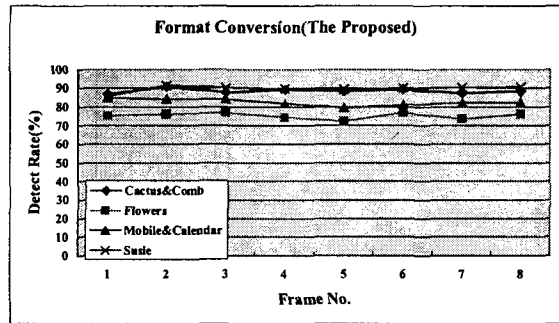
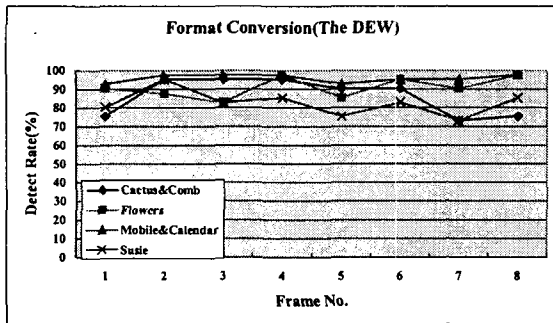


그림 6 실험결과(Format Conversion)