

MPEG-2 비트 스트림에 대한 고속 워터마크 삽입방법

(A fast watermark embedding method for MPEG-2 bit stream)

김성일, 서정일, 김구영, 원치선

서울 중구 필동 3-26 동국대학교 전자공학과

우편번호 : 100-715

전화 : 260-3337

fax : 260-3337

e-mail : cswon@cakra.dongguk.ac.kr

Abstract

In this paper, we propose a new watermarking algorithm for copyright protection of video data. The proposed algorithm inserts a watermark directly on the MPEG-2 bitstream. Since more and more video data are stored and transmitted in a compressed form, it is desirable to insert a watermark on the compressed bit stream to avoid the expensive full-decoding and re-encoding process. Embedding a watermark in the compressed domain, we can also avoid the effect of the compression error which may erase the watermark.

1. 서론

영상과 비디오의 디지털 전송 및 저장이 보편화되면서 원본과 똑같은 복사 영상이 매우 쉽게 만들어지고, 간단한 방법으로 내용의 일부를 변경시킬 수 있으며, 네트워크(network)상으로 쉽게 배포될 수 있게 되었다. 이와 같이 디지털 비디오의 불법적인 복사가 행하여질 경우 저작권을 증명할 근거가 필요하게 되었다[1]. 그러한 이유로 비디오 데이터의 저작권을 보호하기 위해 디지털 문양을 삽입하는 방법이 제안되고 있다[2][3].

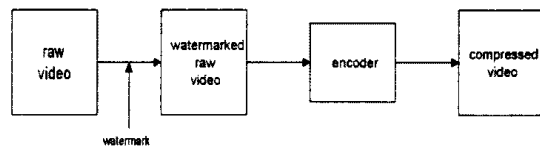
본 논문에서는 <그림 1>과 같이 압축되지 않은 원영상에 워터마크(watermark)를 삽입하는 방법이나[2], <그림 2>와 같이 압축하는 과정에서 워터마크를 삽입하는 기존의 방법[3][4]과는 달리 <그림 3>에서처럼 이미 MPEG-2로 압축되어 있는 비트열에 직접 워터마크를 삽입하는 방법을 제안한다.

제안한 알고리즘은 이미 압축하여 저장된 비디오 데이터의 비트열에 직접 워터마크를 삽입하는 방법이다. 이 방법은 이미 압축되어 있는 데이터의 경우 완전히 디코딩한 후 원영상에 워터마크를 삽입하고 다시 압축하는 기존의 방법보다 매우 효율적이며, 그 결과는 원영상에 워터마크를 삽입한 결과와 같은 효과를 낼 수 있고, 전체 동영상에서 특정한 장면만을 발췌하거나 장면의 순서를 재배열하는 동영상에 대한 공격에 강할 수 있다.

삽입된 워터마크는 비트열에서 워터마크를 삽입하지 않고 복원한 영상과 워터마크가 삽입된 비트열을 복원한 영상 사이의 평균값을 비교함으로써 간단히 검출 될 수 있다.

본 논문의 구성은 다음과 같다. 제2장에서 기존 알고리즘의 장점과 단점, 제3장에서는 제안 알고리즘의 구조, 제4장에서는 실험결과 그리고 제5장에서 결론으로 끝을 맺는다.

2. 기존의 워터마크 구조



<그림 1> 원영상에 워터마크를 삽입하는 구조

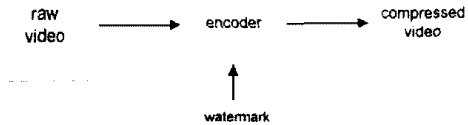
<그림 1>은 원비디오에 워터마크를 삽입하여 워터마크가 삽입된 비디오를 생성한 다음 이를 인코더에 입력하여 압축된 비트열을 만들어 내는 구조이다.

이미 비디오 데이터가 압축상태로 저장되어 있을 경우 <그림 1>의 방법으로 워터마크를 삽입하려면 먼저 압축상태의 비트열을 완전히 디코딩한 다음 <그림 1>과 같은 과정을 거쳐 워터마크가 삽입된 비트열을 얻을 수 있다. 이러한 방법은 원비디오 데이터인 경우에는 간단한 방법으로 워터마크를 삽입할 수 있는 장점

이 있지만, 워터마크를 삽입했을 때 워터마크를 삽입하지 않은 원비디오를 압축했을 때 보다 비트열이 증가할 수 있고, 압축과정에서 워터마크에 손실을 가져올 수 있는 단점이 있다.

<그림 2>는 압축하는 과정에서 DCT과정 후 DCT 계수에 워터마크를 삽입하거나, 양자화 과정후 양자화 계수에 워터마크를 삽입하는 구조이다.

<그림 1>과 마찬가지로 비디오 데이터가 이미 압축되어 있는 경우 완전히 디코딩을 하고



<그림 2> 압축과정중에 워터마크를 삽입하는 구조

다시 압축하는 과정에서 워터마크를 삽입해야 하는 방법이다. 이 방법은 <그림 1>의 방법보다 속도 면에서 좀더 빠를 수 있는 장점이 있지만, 압축과정중에 워터마크가 손실될 우려가 있고, 워터마크를 삽입하기 위해서 표준 인코더에 수정을 가해야 된다는 큰 단점을 가지고 있다.



<그림 3> 비트열에 워터마크를 삽입하는 구조

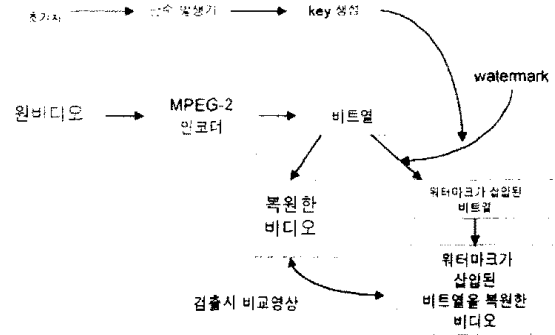
<그림 3>의 구조는 이미 압축이 끝난 비트열의 상태에서 워터마크를 삽입하는 구조이다 [5][6]. [5]에서는 비트열을 역 VLC와 역 양자화 과정을 수행한 후에 모든 I, P, B 그림의 DCT계수에 워터마크를 삽입하였는데 이 때문에 이전 영상에 삽입한 워터마크가 계속적으로 다음영상으로 전파되어 이를 보상하기 위해 워터마크 신호이외에 보상신호를 넣어야 하는 단점이 있다.

[6]에서는 두 가지 방법을 제안했는데 첫째는 비트열에서 직접 조건에 맞는 run-length 코딩에서 level의 값을 바꾸는 방법, 둘째는 비트열을 역 VLC와 역 양자화과정을 수행한 후 인간 시각에 크게 민감하지 않은 고주파부분을 삭제하여 오히려 원영상의 비트열 크기보다 워터마크가 삽입된 영상의 비트열 크기를 줄이는 방법에 대하여 제안하였다.

3. 제안하는 워터마크 구조

본 논문에서는 <그림 3>과 같은 방법을 사용하였는데 [5][6]과는 달리 MPEG-2로 압축된 비트열에 역 VLC와 역 양자화과정을 거치지

않고 비트열에 직접 워터마크를 삽입하는 구조를 제안한다.

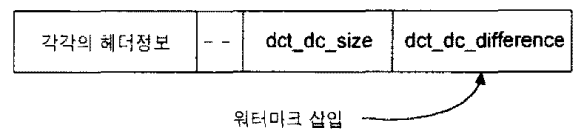


<그림 4> 제안한 워터마크 알고리즘의 전체적인 구조

<그림 4>는 제안한 워터마크 방법의 구조이다.

먼저, 난수 발생기에 초기치를 넣어 key를 생성한다. 이 key를 이용하여 표준 MPEG-2 인코더로 압축된 비트열을 파싱(parsing)하여 각각의 GOP내 I-picture의 블록에서 dct_dc_size를 읽어 낸다. 조건에 맞는 dct_dc_size값을 찾게되면 key를 참조하여 dct_dc_difference값에 직접 워터마크를 삽입한다. 블록내의 데이터들은 앞 블록과의 차분 값이기 때문에 삽입된 워터마크는 계속해서 다음 블록으로 전파된다. 따라서 원하는 블록에만 워터마크를 삽입하기 위해서는 원하는 블록의 다음 블록에서 삽입된 워터마크 만큼을 빼주어야 한다.

이 워터마크 방법은 dct_dc_difference값에 변화를 주기 때문에 복원한 영상에서는 블록내 평균값에 영향을 끼치게 된다. 이러한 이유로 워터마크를 검출할 때는 워터마크가 삽입된 비트열을 복원한 영상과 워터마크를 넣지 않은 비트열을 복원한 영상에 대하여 평균값을 비교함으로써 워터마크를 검출할 수 있다.



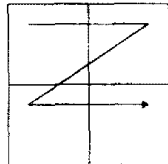
<그림 5> MPEG-2 비트열내의 정보

<그림 5>는 MPEG-2 비트열의 일부분을 나타낸다. 비트열은 여러 가지 헤더 정보들과 데이터들로 구성되는데 본 논문에서 제안하는 워터마크 삽입방법은 헤더정보들은 변화를 주지 않고 각각의 GOP내 I-picture에만 워터마크를 삽입하면 그 뒤의 B, P-picture에는 움직임 추정을 통하여 워터마크가 삽입되는 구조이다.

I-picture는 모두 인트라코딩(intra coding)으로 되어있기 때문에 스킵(skip) 블록이 없고 각각의 블록마다 dct_dc_size와 dct_dc_difference값이 포함되어 있다. 여기서 각각의 블록에 대한 dct_dc_size의 길이정보를 이용하여 워터마크가 삽입될 영역 즉, 단순영역, 예지영역, 질감

영역을 선택하고 dct_dc_difference값에 워터마크를 삽입하여 블록간의 평균 밝기의 세기 차를 변형하여 워터마크 효과를 얻는 방법이다.

dct_dc_size와 dct_dc_difference는 바로 앞 블록(슬라이스(slice)의 맨 처음 매크로 블록의 최초 블록은 정해진 값128)과의 차분 정보를 나타내는데 앞 블록과 밝기 차이가 클수록 즉, 에지영역이나 질감영역일수록 dct_dc_size값이 커지게 되고 앞 블록과 밝기 차이가 작을수록 즉, 단순영역 일수록 dct_dc_size값이 작아지게 된다.



<그림 6> 매크로 블록내의 코딩순서

128	130	140
	120	150

<그림 7> 원영상내 매크로 블록의 DC값

2	10
-20	30

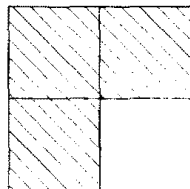
<그림 8> <그림 7>이 코딩되는 DC값

<그림 7>은 슬라이스의 맨 처음 매크로 블록이고, 블록 안의 숫자는 그 블록의 DC값이다. 이때 128은 이전블록을 나타내는 참조값이고 130은 지금 코딩되어야 할 값이다. 차분이 코딩되어지기 때문에 코딩되는 값은 현재 데이터에서 참조값과의 차분을 나타내는 <그림 8>과 같다.

<그림 8>은 <그림 6>과 같은 순서로 바로 앞 블록과의 차분을 나타내기 때문에 어느 한 블록의 값을 바꾸면 그 값이 계속 전파된다. 그렇기 때문에 예를 들어 <그림 9>와 같이 +1을 삽입하고, 그 후에 -1을 하면 <그림 10>과 같은 패턴으로 워터마크가 삽입되게 된다.

2+1	10
-20	30-1

<그림 9> 워터마크를 삽입한 매크로 블록



<그림 10> 워터마크가 삽입된 패턴

위의 과정을 통하여 워터마크가 삽입되면 이전 블록과의 차분 값인 dct_dc_difference에 워터마크가 삽입되기 때문에 복원한 영상에서는 dct_dc_difference에 양의 수치를 더한 만큼 8*8블록의 전체 평균값이 증가하게 되고, 음의

수치를 더한 만큼 전체 평균값이 감소하게 된다.

이 워터마크 방법은 이미 비디오 데이터가 압축상태로 저장되어 있을 경우 디코딩과정(역 VLC, 역 양자화)이 필요 없이 직접 비트열에 워터마크를 삽입하기 때문에 매우 효율적이다. 또한, 단순한 비트 계산만을 이용하기 때문에 매우 빠르게 워터마크가 삽입된 비트열을 생성할 수 있다. 그리고 이미 압축이 끝난 비트열에 워터마크를 삽입하기 때문에 인코더의 수정이 불필요하며, <그림 1>과 <그림 2>의 방법에서 나타났던 표준 인코더에 의한 압축과정에서 워터마크가 손실될 우려가 없는 장점이 있다.

3.1 워터마크 삽입 방법

- ① 난수 발생기에 초기치를 주어 key를 생성한 다음 이 초기 key값을 저장해 놓는다.
- ② 단순영역, 에지영역, 질감영역에 대해서 어느 영역에 워터마크를 삽입할 것인지 dct_dc_size에 조건을 주어 결정한다.
- ③ 비트열을 파싱하여 각각의 GOP내의 I-picture 블록 데이터중에서 dct_dc_size값을 읽어 ②에서 정한 dct_dc_size 조건에 맞는 값을 찾는다.
- ④ ③의 조건에 맞는 dct_dc_size일 경우 dct_dc_difference값에 원하는 만큼의 워터마크를 삽입했을 때 오버플로우(overflow)가 일어나는지 점검한다.
- ⑤ ④의 과정을 거친 후 생성된 key값을 읽어 조건에 맞는 값일 경우 이 블록에 워터마크를 삽입한다. 그 결과는 이 블록부터 계속해서 워터마크가 삽입되게 된다. 따라서 그 뒤 블록에서 삽입한 만큼의 워터마크를 빼주어야 한다.
- ⑥ 원하는 블록만큼 워터마크를 삽입했다면 그 뒤 블록에 대해서 조건에 맞는 dct_dc_size를 읽은 다음 dct_dc_difference에서 삽입한 워터마크 만큼을 빼 값이 언더플로우(underflow)가 일어나지 않을 경우 삽입한 워터마크 값을 빼면 그 사이의 블록에 원하는 만큼의 워터마크가 삽입된다.
- ⑦ 각각의 GOP내에서 I-picture의 블록 데이터가 끝날 때까지 ③에서 ⑥까지를 계속 반복한다.

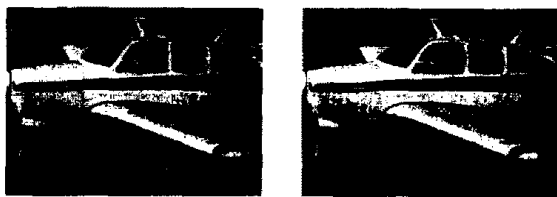
3.2 워터마크 검출 방법

제안한 워터마크 방법은 영상의 DC값에 영향을 주기 때문에 워터마크의 검출은 <그림 4>에서와 같이 워터마크를 넣지 않은 비트열을 다시 복원한 영상과 워터마크를 넣은 비트열을 복원한 영상의 8*8블록당 평균값을 비교함으로써 간단히 검출할 수 있다.

- ① 원 비디오 데이터의 비트열에 같은 초기 key를 가지고 워터마크를 삽입하여 워터마크가 삽입된 비트열을 만들어 낸다.

- ② 만들어진 비트열과 원 비디오 데이터의 비트열을 복원한 다음 평균값을 비교하여 패턴 정보를 얻어낸다.
- ③ 이 패턴 정보를 가지고 원영상과 상대방 영상의 전체 비디오 데이터를 매크로 블록 단위로 비교한다. 즉, 원영상의 첫 번째 블록의 평균값과 상대방영상의 첫 번째 블록 값을 비교하여 그 차이가 패턴정보와 같게 되면 공격이 없었던 영상으로 판단하고 <그림 6>의 순서에 따라 다음 블록을 비교한다. 만약 차이가 있으면 공격이 있었다고 판단하고 그 차이 값을 저장해 두었다가 그 다음 블록들을 비교할 때 차이 값을 오프셋으로 더하여 준다.

4. 실험 결과



시간 t에서의 영상 (I-Picture) 시간 t+1에서의 영상 (B-Picture)
 <그림 11> 원영상을 압축, 복원한 영상



시간 t에서 워터마크를 삽입한 패턴 시간 t+1에서 워터마크를 삽입한 패턴
 <그림 12> 워터마크의 패턴 정보



시간 t에서 워터마크가 삽입된 영상 시간 t+1에서 워터마크가 삽입된 영상
 <그림 13> 워터마크를 삽입한 영상

<그림 11>은 압축되어 있는 비트열을 그대로 복원한 영상이고, <그림 12>는 워터마크를 삽입한 패턴 정보를 나타낸 그림으로 흰색으로 표시한 블록은 워터마크가 삽입된 블록이고 검은색으로 표시한 그림은 워터마크가 삽입되지 않은 블록이다. <그림 13>는 <그림 11>에 <그림 12>과 같은 패턴으로 워터마크를 삽입한 후 복원한 영상이다. 위의 예는 dct_dc_size 가 3이상일 때 워터마크를 +1 삽입하고

dct_dc_size 가 2이상일 때 워터마크를 -1 삽입한 경우이다.

5. 결론

본 논문은 MPEG-2로 압축되어 있는 비디오의 비트열에 직접 워터마크를 삽입하는 구조를 제안하였다. 이 방법은 방송국처럼 데이터를 미리 저장해 놓은 곳에서 데이터에 워터마크를 삽입할 경우 디코딩 없이 바로 비트열에 워터마크를 삽입할 수 있기 때문에 기존의 방법보다 매우 효율적이라고 할 수 있다. 그리고 데이터가 이미 압축이 되어 있기 때문에 인코더의 수정이 불필요하고, 표준 인코더에 의한 압축과정에서 워터마크가 손실될 우려가 없다. 공격에 있어서 전체 영상에 대한 모든 picture의 패턴 정보를 검색할 수 있기 때문에 특정 picture의 삭제, 대치, 발췌나 picture의 순서를 바꾸는 등 영상적인 편집 공격에 강할 수 있고, key를 사용하여 같은 영상에도 다른 워터마크를 삽입하여 공격에 대비하였다.

또, 통신상에서 워터마크의 에러를 방지하기 위해 수신단에서 워터마크를 삽입한다면 본 논문에서 제안하는 방법이 필수적이라고 생각된다.

<참고 문헌>

- [1] B. M. Macq, J.-J. Quisquater. "Cryptology for Digital TV Broadcasting" Proceedings of the IEEE, VOL. 83, NO. 6, JUNE 1995
- [2] Caronni G. "Assuring Ownership Rights for Digital Images", Proceedings of Reliable IT Systems, VIS '95, Vieweg Publishing Company, Germany, 1995.
- [3] I.J.K.O. Ruanaidh, W.J. Dowling, F.M. Boland "Watermarking digital images for copyright protection", IEE Proc.-Vis. Image Signal Process., Vol. 143, No.4, August 1996.
- [4] I.J. Cox, J. Kilian, T. Leighton, T. Shamoon. "Secure Spread Spectrum Watermarking for Multimedia", NEC Reserch Institute, Technical Report 95-10.
- [5] F. Hartung and B. Girod, "watermarking of MPEG-2 Encoded Video Without Decoding and Re-encoding", Proceedings Multimedia Computing and Networking 1997 (MMCN97), San Jose, CA, February 1997.
- [6] G.C.Langelaar, R.L. Lagendijk, J.Biemon. "Real-time Labeling Methods for MPEG Compressed Video" 18th Symposium on Information Theory in the Benelux, 15-16 May 1997, Veldhoven, The Netherlands.