

MPEG 비디오 스트림에 대한 MC-DCT 영역에서의 자막 처리 방법

권오형^o 남종호
서강대학교 컴퓨터학과
sgblue@chollian.net, jhnang@ccs.sogang.ac.kr

Caption Processing for MPEG Video Stream in MC-DCT Domain

O Hyung Kwon^o Jong Ho Nang
Dept. of Computer Science & Engineering, Sogang University

요 약

MPEG은 동영상 압축 규약의 하나로서 고품질을 유지하면서 높은 압축율을 제공하기 때문에 그 사용 범위가 점차로 넓어지고 있으며, 또한 이러한 이유로 인해서 MPEG 데이터에 대한 편집 시스템의 요구도 점차 증가되고 있다. 그러나 MPEG 스트림에 대한 편집은 대부분 공간 영역에서 수행되기 때문에 수행 시간이 오래 걸리고 화질이 저하된다는 단점을 안고 있다. 본 논문에서는 DC 이미지를 이용한 MC-DCT 영역에서의 자막 처리 방법을 제안함으로써 기존의 자막 처리 방법이 안고 있는 단점을 극복하도록 하였다. 제안한 방법에서는 MC-DCT 영역에서의 워터마크 삽입 방법[1]을 응용하여 빠른 처리가 가능하도록 하였으며, MC-DCT 영역에서 쉽게 구할 수 있는 DC 이미지를 이용하여 화면 상태에 따라 자막의 세기를 조절함으로써 P, B 프레임의 처리 과정에서 발생하게 되는 화질 저하를 최소화하도록 하였다. 이러한 MC-DCT 영역에서의 자막 방법을 사용함으로써 기존의 자막 처리 방법 보다 약 4.9 배 빠른 수행 속도를 얻을 수 있었으며, 화질 보존의 측면에서도 기존의 자막 처리 방법이 화면 전체에 대해서 화질 저하가 발생하는 반면 제안한 방법은 자막이 들어가는 부분에 대해서만 화질 저하가 발생하였다. 이러한 방법을 MPEG 스트림에 대해서 사용하게 된다면, 소프트웨어만을 이용한 저가의 데스크 탑 환경에서도 효율적인 자막 처리가 가능하게 될 것이다.

1. 서론

MPEG[8]은 동영상 압축 규약의 하나로서 고품질의 영상을 보장하면서도 높은 압축율을 제공하기 때문에 널리 보급되어 사용되고 있으며 이에 대한 편집 요구 또한 급격히 증가하게 되었다. MPEG 비디오 스트림에 대한 기존의 편집 방법으로는 공간 영역(Spatial Domain)에서의 편집 방법과 이산 코사인(DCT) 영역에서의 편집 방법[2,3,4,5,6,7]을 들 수 있다. 공간 영역에서의 편집 방법은 압축된 비디오 스트림을 원시 형태로 완전히 복원한 후에 편집 효과를 가하고 이를 다시 인코딩하는 방법을 사용하며, 이 과정에서 IDCT, DCT, 역양자화, 양자화, 움직임 예측 등의 작업이 필요하고 이로 인해서 수행 시간이 오래 걸리고 화질이 저하된다는 단점을 안고 있다. DCT 영역에서의 편집은 IDCT, DCT, 움직임 예측, 양자화 과정을 거치지 않기 때문에 빠른 시간에 최소한의 화질 저하와 빠른 시간 내에 처리될 수 있다는 장점이 있다. 그러나, 이 같은 장점들은 1 프로임으로 구성된 MPEG 스트림에 대해서는 효과가 크지만 P, B 프레임을 포함하고 있는 일반적인 MPEG 스트림에 대해서는 그 효과가 미약하다.

본 논문에서는 편집에 있어서 많이 사용되고 있는 자막 효과를 MPEG 비디오 스트림을 대상으로 MC-DCT 영역에서의 자막 처리 방법을 제안함으로써 빠른 시간 내에 고품질의 출력물을 얻을 수 있게 하였다. 제안한 방법에서는 MC-DCT 영역에서의 워터마크 기법을 응용하였으며, 이 방법이 자막 처리에 사용되었을 경우에 발생하게 되는 화질 저하의 원인에 대해서 알아보고 그 해결책으로써 DC 이미지를 참조하여 자막 픽셀 값을 조절하는 방법을 사용하였다. 그 결과 화질 저하를 최소화하면서도 공간 영역의 처리 방법보다 약 4.9배 빠른 수행 시간을 얻을 수 있었다.

2. 연구 배경

본 장에서는 공간 영역에서의 자막 처리 방법과 DCT 영역에서의 자막 처리 방법과 그 문제점, 그리고 MC-DCT 영역에서의 워터마크 기법이 자막 처리에 사용될 경우 발생하게 될 문제점들에 대해서 살펴보도록 하겠다.

2.1 기존의 자막 처리 방법

공간 영역에서 자막 처리를 구현하기 위해서는 완전한 디코딩 과정과 인코딩 과정이 필요하게 된다. 이것은 DCT, IDCT, 그리고, 움직임 예측 과정의 높은 계산 복잡도로 인해

서 수행 시간이 오래 걸리며, 양자화 과정을 거치는 과정에서 화질이 저하된다는 단점을 안고 있다. 자막 처리 과정은 다음의 식 (1)에서 보는 것과 같이 나타낼 수 있다.

$$x'_i = x_i(1-\alpha_i) + y_i\alpha_i \quad (1)$$

여기서 x_i 는 자막이 들어갈 대상 이미지의 픽셀 값을 의미하며, y_i 는 자막의 픽셀 값을 나타낸다. α_i 는 자막 픽셀에 부여된 알파 매스킹의 값을 의미하며, 0과 1 사이의 값을 갖는다.

반면 DCT 영역에서의 자막 처리는 식 (1)을 DCT 영역의 식으로 변환한 식 (2)와 같이 나타내어 질 수 있다. 이 때 x, y, x' 는 각각 입력 블록, 자막 블록, 자막 처리된 결과 블록을 나타내며, α 는 자막 블록의 픽셀 세기를 나타내는 블록을 의미한다. 그리고, $DCT(t)$ 는 블록 t 에 DCT 연산을 수행한 결과이며, \otimes 는 콘볼루션을 의미한다.

$$DCT(x') = DCT(\alpha) \otimes DCT(x) + DCT(1-\alpha) \otimes DCT(y) \quad (2)$$

이 방법은 I 프레임에 대해서는 IDCT, DCT, 움직임 예측 등의 과정을 거치지 않고 수행될 수 있기 때문에 빠른 수행이 가능하지만 P, B 프레임에 대해서는 큰 효과를 볼 수 없는 방법이다. 또한 콘볼루션의 부담도 함께 안고 있다.

2.2 MC-DCT 영역에서의 워터마킹 기법

이 절에서는 본 논문의 토대가 되는 MC-DCT 영역에서의 워터마킹 기법에 대해서 살펴보도록 하겠다. I 프레임에 대한 처리는 I 프레임의 구성하는 모든 매크로블록이 인트라 매크로블록이기 때문에 단순히 DCT 영역에서 워터마크 이미지의 세기를 조절하여 더하는 방법을 사용하게 된다. 인트라 매크로블록에 대한 처리는 식 (3)에서 보는 것과 같이 참조 프레임의 워터마킹 과정에서 더해진 워터마크의 영향을 제거하고 새로운 워터마크를 더해주는 과정으로 나타낼 수 있다.

$$DCT(P_{new}) = DCT(P_b) - MC(MV(P_b), F_c) + DCT(F_c) \quad (3)$$

여기서 P_b 는 입력 블록이고, F_c 는 워터마크 프레임을 나타내고, $MV(P_b)$ 는 P_b 의 움직임 벡터를 나타내며, $MC()$ 는 움직임 보상 과정을 나타낸다.

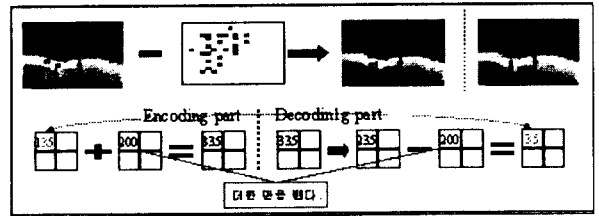
3. MC-DCT 영역에서의 효과적인 자막 처리 방법

본 장에서는 MC-DCT 영역에서의 워터마킹 기법을 자막 처리에 응용하였을 경우 발생하게 되는 문제점에 대해서 알아보고 이의 원인과 그 해결 방법에 대해서 살펴보도록 하겠다.

3.1 MC-DCT 영역에서의 워터마킹 기법의 문제점

<그림 1>은 MC-DCT 영역에서의 워터마킹 기법의 문제점과 그 원인을 나타낸 것이다. P, B 프레임을 처리하는 데 있어서 가장 중요한 원리는 참조 프레임에 더해진 워터마크만큼 빼주고 해당 워터마크를 더한다

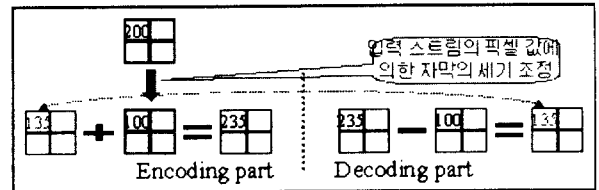
는 것이다. 그러나, <그림 1>에서 볼 수 있듯이 더해지는 워터마크의 값이 너무 클 경우 워터마킹 처리 이전의 값을 제대로 복원할 수 없다는 문제점이 발생하게 된다. 이것을 공간 영역의 동작을 이용하여 <그림 1>을 토대로 설명하도록 하겠다. 입력 픽셀 135에 워터마크에 해당하는 200이 더해져서 335가 인코딩된다. 이후 디코딩 과정에서 335는 235로 조절되고 여기에서 200을 빼줌으로 135가 아닌 35를 얻게 된다. 이것은 P, B 프레임에 발생하게 되는 화질 저하의 원인이 되며, 특히 자막과 같이 더해지는 이미지의 픽셀 값이 큰 경우에는 화질 저하가 더욱 심각하게 된다.



<그림 1> 화질 저하의 원인

3.2 DC 이미지를 이용한 해결 방법

<그림 2>는 3.1 절에서 설명한 화질 저하의 원인을 예방하기 위해서 입력 픽셀의 값에 따라서 워터마크 픽셀의 값을 조절하고 있다. 즉, 입력 픽셀의 값 135에 따라서 워터마크 픽셀의 값 200을 100으로 조절함으로써 235를 넘지 않도록하여 인코딩하고 있다.



<그림 2> 자막 픽셀을 이용한 화질 저하 해결 방법

식 (4)는 위의 과정을 수식화한 것이다.

$$x'_i = x_i + y_i * \left(1 - \frac{x_i}{235}\right) \quad (4)$$

이 식에서 x', x, y 는 각각 자막 처리된 출력 프레임의 블록, 입력 프레임의 블록, 자막 이미지 블록이며, x'_i, x_i, y_i 는 각각 c 블록들의 i 번째 픽셀 값이다.

$$X' = X + Y * \left(1 - \frac{DC(X)}{1880}\right) \quad (5)$$

식 (5)는 식 (4)를 DCT 영역으로 옮긴 것이며, 식 (4)에서 픽셀 단위로 처리하던 것을 블록 단위로 처리하도록 하였고, 이때 자막의 세기를 조절하기 위해서 사용된 x_i 를 블록의 평균에 해당하는 DC 값으로 근사한 것이다.

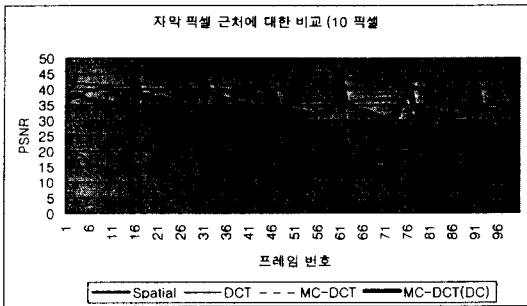
이러한 방법을 사용함으로써 자막 근처의 화질 저하 현상을 상당부분 제거할 수 있었지만 DC로 근사하는 과정에서 발생한 오차의 영향으로 약간의 화질 저하는 남아있다.

4. 성능 평가 및 분석

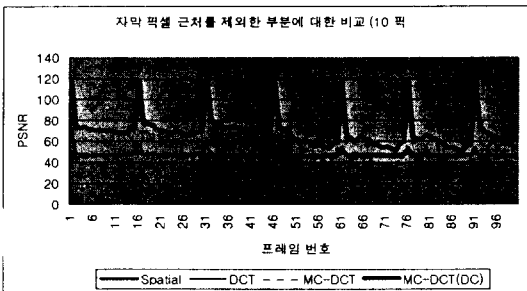
이 장에서는 기존의 자막 처리 방법과 제안한 방법을 실제로 구현하고 실험한 결과를 토대로 화질과 수행 속도 면에서 성능을 비교 분석하도록 하겠다.

4.1 화질 비교

<그림 3>은 자막 근처에 대한 화질 비교를 나타내고 있으며, <그림 4>는 이 영역을 제외한 영역에 대한 화질 비교를 나타내고 있다. <그림 3>에서는 제안한 방법(MC-DCT(DC))이 공간 영역의 방법에 비해서 다소 낮은 PSNR을 갖는다는 것을 볼 수 있다. 이것은 자막의 세기를 DC에 따라서 조절함으로써 자막 픽셀 자체의 값이 작아졌으며, DC로 근사하는 과정에서 발생한 오차가 남아있기 때문이다. 그러나, DC를 사용하지 않는 MC-DCT 영역의 방법과 비교해 볼 경우 월등히 높은 PSNR을 보이는 것을 알 수 있다. <그림 4>에서는 제안한 방법이 월등히 높은 PSNR을 보여주는 것을 볼 수 있다. 이것은 MC-DCT 영역에서의 방법이 자막의 근처를 제외하고는 높은 화질 유지 특성이 있다는 것을 알 수 있게 해준다.



<그림 3> 자막 픽셀 근처에 대한 화질 비교



<그림 4> 자막 픽셀 근처를 제외한 부분에 대한 화질 비교

4.2 수행 속도 비교

공간 영역의 방법은 0.53(sec/frame), DCT 영역의 방법은 0.41(sec/frame), MC-DCT 영역의 방법은 0.11(sec/frame), 제안한 방법은 0.11(sec/frame)의 결과로써 제안한 방법이 공간 영역의 방법보다 약 4.9배 빠른 것으로 나타났다. 이것은 제안한 방법이 자막 처리에 있어서 IDCT, DCT, 움직임 예측 등 계산 복잡도가 높은 과정을 거치지 않기 때문이다.

5. 결론 및 추후 연구 방향

MPEG은 고화질의 영상을 보장하면서도 높은 압축율을 제공하기 때문에 널리 보급되어 사용되고 있으며, 이에 대한 편집 욕구도 함께 증가하고 있는 실정이다. 그러나, 기존의 편집 방법은 수행 시간이 오래 걸리고 처리 후 화질이 저하된다는 단점을 안고 있다.

본 논문에서는 기존 편집 시스템의 단점을 극복하기 위해서 MC-DCT 영역에서의 워터마킹 기법을 응용한 효과적인 자막 방법을 제안하였다. MC-DCT 영역에서의 워터마킹 기법을 자막 처리에 응용할 경우에 발생하게 되는 문제점의 원인을 살펴보고 그에 대한 해결책으로써 DC 이미지를 이용하여 자막의 세기를 블록 단위로 조절하는 방법을 제안하였다. 본 논문에서 제안한 자막 방법을 사용함으로써 공간 영역의 자막 처리 방법보다 약 4.9배 빠른 수행 시간을 얻을 수 있었다. 또한 공간 영역의 자막 처리 방법이 화면 전체에 걸친 화질 저하 현상을 보이는 반면 본 논문에서 제안한 방법은 자막 부근에 대해서만 화질 저하 현상이 발생되고 그 이외의 부분에 대해서는 화질 저하 현상이 발생되지 않는 결과를 얻었다.

본 논문에서는 P, B 프레임의 처리 과정 중 참조 프레임에 추가된 자막의 영향을 제거하는 과정에서 공간 영역의 자막 이미지를 대상으로 움직임 보상을 수행하고 이것을 다시 이산 코사인 변환하는 방법을 사용하였다. 이 과정을 이산 코사인 변환 영역에서의 움직임 보상법으로 대체할 경우 보다 빠른 시간에 처리할 수 있을 것으로 생각되어진다.

6. 참고 문헌

- [1] Jianhao Meng and Shih-Fu Chang, "Embedding Visible Video Watermarks in the Compressed Domain," *Proc. of ICIP International Conference on Image Processing*, Oct. 1998.
- [2] Shih-Fu Chang and David G. Messerschmitt, "Manipulation and Compositing of MC-DCT Compressed Video," *IEEE Journal on Selected Areas in Communications*, Vol. 13, No. 1, Jan. 1995.
- [3] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System," *Proc. of ACM Multimedia 96 Conference*, Nov. 1996.
- [4] Bo Shen, Ishwar K. Sethi and Vasudev Bhaskaran, "DCT Convolution and Its Application in Compressed Video Editing," *Proc. of SPIE 3024 in Visual Communications and Image Processing*, Feb. 1997.
- [5] Bo Shen and Ishwar K. Sethi, "Inner-Block Operations On Compressed Images," *Proc. of ACM Multimedia 95*, Nov. 1995.
- [6] Neri Merhav and Vasudev Bhaskaran, *A Fast Algorithm for DCT-Domain Inverse Motion Compensation*, HPL Technical Report #HPL-95-17, Sep. 1995.
- [7] Soam Acharya and Brian Smith, "Compressed Domain Transcoding of MPEG," *Proc. of IEEE International Conference on Multimedia Computing and Systems*, Jul. 1998.
- [8] ISO/IEC/JTC1/SC29/WG11, *Coding of Moving Pictures and Associated Audio for digital storage media at up to about 1.5 Mbit/s*, ISO/IEC International Standard 11172-2, Aug. 1993.