

# MPEG 비디오 스트림에 대한 MC-DCT 영역에서의 자막 처리 방법

## (Caption Processing Method for MPEG Video Stream in MC-DCT Dom)

권 오 형 <sup>†</sup>    낭 종 호 <sup>\*\*</sup>

(Ohyung Kwon) (Jongho Nang)

**요 약** MPEG은 동영상 압축 규약의 하나로서 고화질을 유지하면서 높은 압축율을 제공하기 때문에 그 사용 범위가 점차로 넓어지고 있으며, 또한 이러한 이유로 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 방법 보다 약 4.9 배 빠른 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.

**Abstract** The (cinema) caption processing that adds descriptive texts on the sequence of frames is an important video manipulation function that video editor should support. This paper proposes an efficient MC-DCT compressed domain approach to insert the caption into the MPEG-compressed video stream. It basically adds the DCT blocks of the caption image to the corresponding DCT blocks of the input frames one by one in MC-DCT domain as in [Meng98]. However, the strength of the caption image is adjusted in the DCT domain to prevent the resulting DCT coefficients from exceeding the maximum value that is allowed in MPEG. In order to adjust the strength of caption image adaptively, we should know the exact pixel value of input image that is a difficult task in DCT domain. We propose an approximation scheme for the pixel values in which the DC value of a block is used as the expected pixel value for all pixels in that block. Although this approximation may lead some errors in the caption area, it still provides a relatively high image quality in the non-caption area, while the processing time is about 4.9 times faster than the decode-captioning-reencode approach.

### 1. 서 론

MPEG[2]은 동영상 압축 규약의 하나로서 고화질의 영상을 보장하면서도 높은 압축율을 제공하기 때문에 널리 보급되어 사용되고 있다. MPEG이 처음 등장하였을 당시 특별하게 제작된 하드웨어의 도움 없이 소프트웨어만을 이용하여 MPEG으로 압축된 동영상을 감상할 수는 없었다. 그러나, 컴퓨터 기술의 눈부신 발달은 추가적인 하드웨어의 도움 없이도 소프트웨어만을 이용한 디코딩 및 인코딩을 가능하게 하고 있으며, 데스크 탑 컴퓨터 상

· 본 연구는 한국과학재단 산학협력연구(98-2-11-02-01-2)지원으로 수행되었음.

<sup>†</sup> 비 회 원 : 서강대학교 컴퓨터학과  
ohkwon@mljuno.sogang.ac.kr  
<sup>\*\*</sup> 종신회원 : 서강대학교 컴퓨터학과 교수  
jhnang@ccs.sogang.ac.kr

논문접수 : 2000년 4월 21일  
심사완료 : 2000년 8월 16일

에서의 소프트웨어 편집 시스템을 가능하게 하고 있다.

영상에 대한 여러 편집 기능 중에서 자막 처리는 영상 위에 글자 등을 덧붙임으로서 글자가 더해진 새로운 영상을 얻는 것을 의미한다. 자막은 대체로 불투명하다는 특성이 있기 때문에 자막 처리 과정에서 자막이 첨가될 부분의 픽셀 정보가 자막의 픽셀 정보로 바뀌게 된다. 우리가 쉽게 접할 수 있는 자막은 영화나 각종 TV 프로그램에서 사용되고 있는 자막일 것이다. 자막처리는 기존의 영상에 새로운 정보를 더함으로써 영상에 대한 이해나 재미를 한 층 더 높일 수 있기 때문에 그 이용가치가 높고 비교적 사용 빈도가 높은 편집 기법 중의 하나이다. 본 논문에서는 이와 같이 동영상 편집에 있어서 많이 사용되고 있는 자막 처리를 MPEG 비디오에 대해서 보다 더 효과적으로 하기 위한 방법을 제안하고자 한다.

MPEG 비디오 스트림에 대한 기존의 편집 방법으로서 공간 영역(Spatial Domain)에서의 편집 방법과 이산 코사인 변환(DCT : Discrete Cosine Transform) 영역에서의 편집 방법[3,4,5,6,7,8]을 들 수 있다. 또한 MC-DCT 영역<sup>1)</sup>에서의 워터마킹 기법[1]이 소개되어 있다. 우선 공간 영역에서의 편집은 원본 비디오 스트림보다 다소 화질이 떨어지는 출력물을 제공하며 이것을 수행하기 위한 소요 시간 또한 너무 오래 걸린다는 단점을 안고 있다. 이러한 문제점이 발생하는 원인은 압축된 비디오 스트림을 원시 형태로 완전히 복원한 후에 편집 효과를 가하고 이를 다시 인코딩하는 방법을 사용하기 때문이다. 이 과정에서 역 이산 코사인 변환(IDCT : Inverse DCT), 이산 코사인 변환, 움직임 예측 과정은 특히 오랜 수행 시간을 요구하며, 양자화 과정은 화질 저하의 주원인이 되고 있다. 이산 코사인 변환 영역에서의 편집은 역 이산 코사인 변환, 이산 코사인 변환, 움직임 예측, 양자화 과정을 거치지 않고 수행될 수 있기 때문에 공간 영역에서의 편집과는 달리 최소한의 화질 저하와 빠른 시간 내에 처리 될 수 있다는 장점이 있다. 그러나, 이와 같은 장점들은 I 프레임만으로 구성된 MPEG 스트림에 대해 국한된 것이며 I, P, B 프레임들로 이루어진 일반적인 MPEG 스트림에 대해서는 그 효과가 미약하다. 마지막으로 MC-DCT 영역에서의 워터마킹 기법은 P, B 프레임에 대해서도 역 이산 코사인 변환, 이산 코사인 변환, 움직임 예측의 과정을 생략할 수 있기 때문에 일반적인 MPEG 스트림을 대상으로 적용하였을 경우에도 빠른 수행 속도를 기대할 수 있다. 그러나, 워터마크는 대체로 흐린 이

미지를 원본 이미지에 덧붙이는 과정이기 때문에 강한 이미지를 갖고 있는 자막에는 그대로 사용할 수 없다는 단점이 있다.

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 자막 처리에 있어서 MC-DCT 영역에서의 워터마킹 삽입 기법을 응용하였으며, 자막의 픽셀 값이 강하기 때문에 발생하게 되는 문제점을 알아보고 이것을 해결하기 위한 방법으로 DC 이미지를 참조하여 자막의 세기를 조절하는 방법을 사용하였다. 즉, 자막은 그 특성상 픽셀 값이 강하기 때문에 P, B 프레임에서 참조 프레임에 더해진 자막의 영향을 제거하는 과정에서 심한 화질 저하가 발생하게 되며, 이것을 해결하기 위한 방법으로 MC-DCT 영역에서 쉽게 구할 수 있는 DC 이미지를 이용하여 블록 단위로 자막의 세기를 조절하도록 하였다. MC-DCT 영역에서 자막 처리를 수행함으로써 자막 근처를 제외한 영역에서는 원본과 동일한 영상을 얻을 수 있었으며, 자막 부분에 대해서도 DC 이미지를 이용하여 자막의 세기를 조절함으로써 화질 저하를 최소화할 수 있었다. 즉, P, B 프레임의 자막 처리에서 발생하게 되는 화질 저하의 문제점을 최소화하는 한편 에러를 최소화하는 범위 내에서 가장 강한 자막을 얻을 수 있도록 하였다. 또한 모든 과정이 디코딩과 인코딩의 일부 과정만을 거친 MC-DCT 영역에서 수행되기 때문에 공간 영역에서의 자막 처리 방법보다 약 4.9 배 빠른 시간 내에 자막 처리를 가능하게 하고 있다.

본 논문의 제 2 장에서는 MPEG 비디오 스트림의 구조와 특성, 인코딩 과정과 기존의 편집 과정에서 사용되는 자막 처리 방법에 대해서 살펴보고, 제 3 장에서는 본 논문에서 제안하고자 하는 자막 처리 방법에 대해서 구체적으로 설명할 것이며, 제 4 장에서는 제안한 방법과 기존의 방법들을 실제로 구현해 보고, 예제 스트림을 통한 비교 실험 및 그 결과 분석을 통한 성능 평가를 하도록 하겠다. 마지막으로 제 5 장에서는 결론을 내리고, 추후 연구 방향을 제시하도록 하겠다.

## 2. 연구 배경

본 장에서는 MPEG으로 압축된 동영상의 편집 과정을 이해하기 위한 배경 지식들에 대하여 설명한다. 2.1 절에서는 MPEG 비디오 스트림에 대한 일반적인 내용을 설명하고, 2.2 절에서는 MPEG 비디오 스트림에 대한 자막 처리에 사용되는 기초적인 원리들에 대해서 설명하겠으

1) 움직임 보상 이전의 이산 코사인 변환 영역으로써 움직임 정보와 예측 에러 블록을 가지고 있다.

며, 2.3 절에서는 기존 처리 방법들과 그에 관한 문제점들에 대하여 알아보도록 하겠다.

**2.1 MPEG 비디오 스트림(2.9)**

동영상은 여러 프레임들로 이루어져 있으며 데이터의 중복성 면에서 볼 때 동일한 프레임 내에 존재하는 공간적 중복성과 연속하는 프레임 사이에 존재하는 시간적 중복성을 지니고 있다. MPEG에서는 두 가지 중복성 모두를 제거하기 위한 방법들을 사용하고 있으며 각각의 방법은 다음과 같다.

공간적 중복성을 줄이기 위한 방법으로는 원시 형태의 데이터를 밝기 정보와 색 정보로 나누는 과정, 이산 코사인 변환, 양자화 그리고, 가변 길이 부호화 등의 과정을 거치게 되며 이러한 방법을 인트라 코딩(intra coding)이라고 한다. 이 방법은 I,B,P 모든 종류의 프레임에서 사용된다. 시간적 중복성을 줄이는 방법으로는 P프레임과 B프레임에서 사용되고 있는 인터 코딩(inter coding) 방법이 있다. 이것은 프레임간의 움직임 예측을 통해서 얻은 움직임 정보와 프레임 사이의 차이만을 저장하는 방식을 취하고 있다. 이와 같이 MPEG은 인트라 코딩 방법과 인터 코딩 방법을 함께 사용함으로써 고도의 압축율을 제공하고 있는 것이다.

**2.2 MPEG 비디오 스트림에 대한 자막 처리를 위한 기초 연산들**

이 절에서는 MPEG 비디오 스트림을 MC-DCT 영역에서 편집하기 위해 제안된 기초 이론들에 대해서 알아보기로 한다.

**2.2.1 블록 내 연산 기법(Inner-Block Algebra)**

블록 내 연산 기법[6]은 공간 영역에서 블록들끼리의 덧셈과 이산 코사인 변환 영역에서 블록들끼리의 덧셈이 동일한 작업을 수행한다는 원리에 기초하고 있다. 즉, 이산 코사인 변환 영역에서 얻은 결과를 공간 영역으로 변환했을 경우 공간 영역에서 덧셈을 수행한 것과 동일한 결과를 얻을 수 있다는 것이다. 이 원리를 이용하면 비디오 스트림에 자막을 더하는 작업을 공간 영역에서 수행하는 것과 동일하게 이산 코사인 변환 영역에서 수행할 수 있게 된다.

**2.2.2 MC-DCT 영역에서의 DC 이미지**

MC-DCT 영역에서 DC 이미지를 이용하는 기법[4, 10,11,12]은 이산 코사인 변환의 특성상 DC값은 그 블록의 평균값을 가지기 때문에 DC값을 그 블록의 대표값으로 사용하자는 것이다. 이런 방법으로 각 블록의 DC값만을 취해서 만든 축소된 이미지가 DC 이미지가 되는 것이다. I 프레임의 경우는 단순히 각 블록의 DC값만을 취하면 되지만, P 프레임과 B 프레임의 경우는 움직임 예

측을 기초로 인코딩 되어 있기 때문에 단순히 각 블록의 DC값만을 취하는 방법으로는 정확한 DC이미지를 만들 수 없게 된다. <그림 1>는 P, B 프레임에서 블록의 DC값을 근사하는 방법을 나타내고 있으며, 식 (1)에는 이것을 수식으로 나타내고 있다. 여기서 a, b, c, d 는 각각 A, B, C, D 블록의 DC 값을 의미한다. P, B 프레임의 E 블록이 앵커 프레임의 A,B,C,D 블록에 걸쳐서 참조하는 경우 식 (1)에 나타난 바와 같이 움직임 벡터를 이용하여 면적에 비례하는 가중치 평균을 구하고, 이렇게 근사된 DC값에 예측 에러 블록의 DC값을 더해지게 된다.

$$DC \text{ of block } E = a \frac{(8-x)(8-y)}{8} + b \frac{x(8-y)}{8} + c \frac{(8-x)y}{8} + d \frac{xy}{8} \quad (1)$$

DC이미지는 프레임의 특성을 대부분 가지고 있으면서도 그 크기가 작기 때문에 비디오 데이터를 대상으로 하는 많은 연구에서 사용되어지고 있으며, 본 논문에서는 DC이미지를 이용하여 자막이 들어가는 블록의 평균 밝기 정보를 구하고 이 정보에 기초하여 자막의 세기를 조절하게 된다.

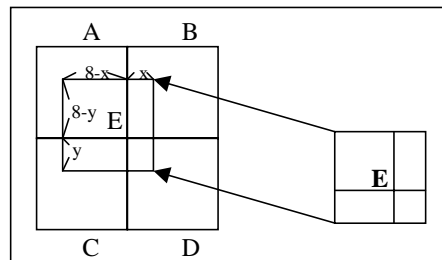


그림 1 면적비례 가중치 평균을 이용한 DC값의 계산

**2.3 기존의 자막 처리 방법**

TV나 영화에서 주로 사용되는 자막은 대체로 불투명한 자막이며 글자의 테두리 부분이 검다는 특징을 가지고 있다. 이와 같은 특징은 화면의 변화에 관계없이 항상 일정한 가시성(Visibility)을 보장하여 사용자가 자막을 읽는 데 있어서 불편을 느끼지 않게 하기 위한 것이다. 자막은 식 (2)와 같은 알파 매스킹(alpha masking)의 원리[3,5]로써 설명되어 질 수 있다.

$$x'_i = x_i(1 - \alpha_i) + y_i\alpha_i \quad (2)$$

여기서  $x_i$ 는 자막이 들어갈 대상 이미지의 픽셀 값을 의미하며,  $y_i$ 는 자막의 픽셀 값을 나타낸다.  $\alpha_i$ 는 자막 픽셀에 부여된 알파 매스킹의 값을 의미하며, 0과 1 사이의 값을 갖는다.  $\alpha_i$ 가 0과 1사이의 값을 갖는 경우

는 투명한 자막이 되지만, 0 또는 1의 값만을 갖는 경우는 불투명 자막이 생성되게 된다. 알파 매스킹에 의한 자막 방법은 공간 영역에서는 낮은 계산 복잡도에서 쉽게 구현될 수 있지만 이산 코사인 변환 영역에서의 구현은 높은 계산 복잡도를 요구하는 DCT Convolution[5]을 이용해야 한다는 단점이 있다. 이러한 문제점으로 인해서 자막 처리는 다음의 식 (3)와 같은 위터마킹 기법[1]을 응용하는 방법을 생각할 수 있다.

$$x'_i = x_i + y_i \alpha_i \quad (3)$$

식 (3)에서  $x_i$ 는 위터마킹이 수행될 이미지의 픽셀 값을 의미하며,  $y_i$ 는 위터마크 이미지의 픽셀 값을 나타낸다.  $\alpha_i$ 는 위터마크 픽셀의 세기를 나타내며, 0과 1 사이의 값을 갖는다. 자막에서 사용되는  $y_i$ 값은 대체로  $y_i$ 의 최대값이 사용되기 때문에 자막에 해당하는  $x_i$ 값이  $y_i$ 값으로 대체되는 효과를 얻게 된다. 즉, 불투명 자막의 효과를 얻게 되는 것이다. 또한, 식 (3)의 방법을 이용하여 이산 코사인 변환영역에서 자막 처리를 수행할 경우,  $\alpha_i$ 에 대한 근사를 거친 후 배열에 대한 스칼라 곱셈과 배열의 덧셈만을 이용하여 자막 처리가 가능하게 되기 때문에 보다 빠른 시간에 자막처리가 가능하다는 잇점이 있다. 그러나, MPEG을 대상으로 MC-DCT 영역에서 수행할 경우 위터마크는 대체로 흐린 이미지를 더하는 작업이기 때문에 그로 인한 화질저하가 심하지 않지만 자막은 강한 이미지를 더하는 작업이므로 이 방법을 그대로 사용할 경우 심한 화질 저하가 발생하게 된다. <그림 2>는 위터마크 기법을 응용한 자막 처리 방법을 나타내었다. 즉 자막 부분은 더해주고 그 테두리 부분은 빼줌으로써 자막 처리가 이뤄지게 된다.

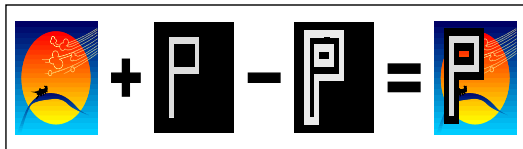


그림 2 위터마크 기법을 응용한 자막 처리 방법

본 절에서는 자막 처리 방법을 공간 영역, 이산 코사인 변환 영역으로 나누어서 그 처리 방법과 문제점을 알아보고 MC-DCT 영역에서의 위터마킹 기법을 소개하기로 한다.

### 2.3.1 공간 영역에서의 자막 처리

이 절에서는 MPEG을 대상으로 공간 영역에서의 자막 처리 방법[13]에 대해서 살펴보고 이 방법의 문제점을 알아보도록 한다. MPEG 비디오 스트림에 대해서 공간 영역에서 자막 처리를 구현 하기 위해서는 완전한 디코딩 과정과 인코딩 과정이 필요하게 된다. 이러한 방법은

단지 디코더와 인코더를 연결해 주고 공간 영역에서 자막 처리를 수행하기 때문에 구현이 용이하고 자막 처리 과정이 간단하다는 장점을 가지고 있다. 그러나, 이와 같은 처리 과정은 수행 시간이 오래 걸리고 화질이 저하된다는 단점을 안고 있다. 이것은 이산 코사인 변환(DCT) 과정과 그 역과정(I-DCT), 그리고 움직임 예측(Motion Estimation) 과정의 높은 계산 복잡도로 인해서 수행 시간이 오래 걸리며, 양자화 과정을 거치는 과정에서 화질이 저하되기 때문이다.

$$x' = \alpha x + (1 - \alpha)y \quad (4)$$

식 (4)은 공간 영역에서의 자막 처리 과정을 나타내고 있다. 식 (4)에서  $x$ ,  $y$ ,  $x'$ 는 각각 원본 프레임, 자막 이미지, 자막 처리된 결과 이미지를 나타내며,  $\alpha$ 는 자막 이미지의 픽셀 단위 세기를 나타낸다.

### 2.3.2 이산 코사인 변환 영역에서의 자막 처리

공간 영역에서의 자막 처리를 수행하게 되면, 앞 절에서 언급했듯이 화질 저하와 수행 시간이 오래 걸린다는 문제점이 발생한다. 이러한 단점을 보완할 수 있는 해결책의 하나로서 이산 코사인 변환 영역에서의 자막 처리 방법[5]을 들 수 있다. 이산 코사인 변환 영역에서의 자막 처리 방법은 식 (5)로 나타내어 질 수 있다.

$$DCT(x') = DCT(\alpha) \otimes DCT(x) + DCT(1 - \alpha) \otimes DCT(y) \quad (5)$$

식 (5)은 공간 영역에서의 자막 처리 과정을 나타내고 있는 식 (4)를 이산 코사인 변환 영역의 식으로 변환한 것이다. 여기서  $DCT(t)$ 는 공간 영역의 이미지  $t$ 에 이산 코사인 변환을 취한 것을 의미하고  $\otimes$ 는 이산 코사인 변환 영역에서의 콘벌루션(Convolution)을 의미한다. 이산 코사인 변환 영역에서의 콘벌루션은 계산 복잡도가 매우 높은 작업으로써 그에 대한 최적화 알고리즘이 이미 발표되었지만 아직까지는 높은 계산 복잡도를 요구하고 있는 작업이다.

이 방법은 입력 MPEG 비디오 스트림에 대한 처리 결과가 모두 I 프레임으로 된 스트림이 된다는 단점이 있지만 공간 영역의 자막 처리보다는 향상된 결과를 얻을 수 있는 방법이다. 특히 입력 스트림이 I 프레임으로만 이루어져 있을 경우에는 이산 코사인 변환 과정과 그 역 과정을 거치지 않은 상태에서 바로 자막 처리를 할 수 있기 때문에 화질과 속도 면에서 모두 만족한 결과를 얻

을 수 있는 방법이다.

2.3.3 MC-DCT 영역에서의 워터마킹

공간 영역과 이산 코사인 변환 영역에서의 처리 과정이 안고 있는 수행 시간과 화질 저하의 문제점을 해결하기 위한 대안으로써 MC-DCT 영역에서의 자막 처리 방법을 생각할 수 있다. 이러한 관점에서 본 논문에서 제안한 방법의 토대가 되는 MC-DCT 영역에서의 워터마킹 기법[1]을 살펴보도록 한다.

(1) I 프레임에서의 워터마킹

I 프레임은 구성하는 모든 매크로블럭들은 인트라 매크로블럭이기 때문에 단순히 이산 코사인 변환 영역에서 워터마크 이미지를 더하는 방법을 사용하게 된다. 즉, 입력 MPEG 비디오 스트림에 대해서 Run-Length 코딩과 허프만 코딩, 그리고 역양자화 및 지그재그 스캔 만을 거친 후 이산 코사인 변환 영역에서 워터마크를 더하는 과정으로 이루어지게 된다. 계산 복잡도가 높은 이산 코사인 변환의 역과정을 거치지 않기 때문에 공간 영역에서의 처리보다 빠르게 수행되어 질 수 있다. 이 때 사용되는 워터마크는 이산 코사인 변환 과정을 거친 이산 코사인 변환 영역의 블록이 된다. 워터마킹 결과가 이산 코사인 변환 영역의 블록이므로 인코딩 과정 또한 계산 복잡도가 높은 이산 코사인 변환 과정을 거치지 않고 단지 지그재그 스캔 및 양자화, 허프만 코딩과 Run-Length 코딩 만을 거침으로써 수행되어질 수 있다.

(2) P, B 프레임에서의 워터마킹

P 프레임과 B 프레임은 움직임 예측을 사용하여 인코딩 되기 때문에 I 프레임과는 달리 인트라 매크로블럭 뿐만 아니라 참조 프레임에 대한 참조 자료, 즉 움직임 정보를 함께 가지고 있는 인터 매크로블럭도 가지고 있다. 입력 프레임에 대해서 부분적인 디코딩, 즉 허프만 코딩과 Run-Length 코딩, 그리고 역양자화 과정 및 지그재그 스캔을 거친 결과로써 움직임 정보와 예측 에러 블록(이산 코사인 변환 영역의 블록)을 얻게 된다. 인트라 매크로블럭들은 독립적으로 인코딩이 되어있기 때문에 I 프레임에서의 매크로블럭들에 대한 방법과 동일한 방법, 즉 단순히 예측 에러 블록에 워터마크 이미지의 이산 코사인 변환 블록을 더하는 과정을 통하여 워터마킹을 수행하게 된다.

<그림 3>은 인터 매크로블럭들에 대한 워터마킹 과정을 나타낸다. 프레임 A가 프레임 B의 참조 프레임일 경우 워터마킹 이후에 프레임 B에 대해서 별도의 조치가 없을 경우는 <그림 3>에서 보듯이 프레임 B가 아닌 프레임 B'을 얻게 된다. 이것은 프레임 A에 워터마크가 더해진 결과로써 프레임 A'이 되었고, 이것을 그대로 참조

하였기 때문에 발생하는 문제점인 것이다. 이 문제점을 해결하기 위해서 프레임 A에 더해진 워터마크 P에 대한 움직임 보상의 결과를 프레임 B'에서 빼줌으로써 프레임 B와 동일한 프레임 B''을 얻을 수 있게 된다. 그 이후에 워터마크를 더해지게 되면 우리가 원하는 대로 워터마크가 더해진 프레임 C를 얻게 되는 것이다. 이 과정을 요약하면 참조 프레임에 더해진 워터마크를 움직임 보상을 통해서 제거한 후에 새롭게 워터마크를 더하는 것이라고 생각할 수 있다. 이 때 주의 할 것은 추가된 워터마크를 제거하기 위한 움직임 보상의 대상은 참조 프레임이 아닌 워터마크 이미지라는 것이다. 즉, 참조 프레임에 더해준 워터마크 이미지를 대상으로 움직임을 하고 그 결과를 빼게 되는 것이다. 이러한 제거 과정을 거친 이후에 워터마크를 더하고 빼는 과정은 인트라 매크로블럭들에 대한 과정과 동일하다. 즉, 워터마크 제거 과정과 워터마크 추가 과정은 모두 DCT 블록들에 대해서 수행된다. 인터 매크로블럭들에 대한 워터마크 처리 과정은 식 (5)로 나타내어 질 수 있다. 참조 프레임의 워터마킹 과정에서 더해진 워터마크는 움직임 보상을 통해서 빼주고 새로운 워터마크를 더해주는 것이다.

$$DCT(P_{new}) = DCT(P_b) - MC(MV(P_b), F_c) + DCT(P_c) \quad (5)$$

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

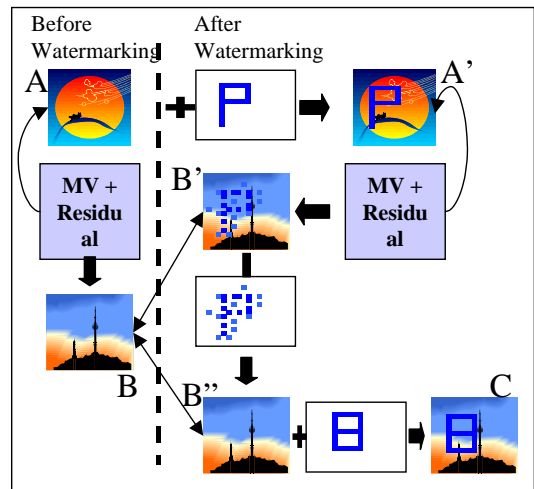


그림 3 인터 매크로블럭에 대한 워터마킹

3. MC-DCT 영역에서의 효과적인 자막 처리 방법

제 2 장에서 살펴본 바와 같이 공간 영역에서의 자막 처리와 이산 코사인 변환 영역에서의 자막 처리는 수행 시간이 오래 걸리고 화질이 저하된다는 단점을 안고 있다. 이것을 해결하기 위한 방법으로써 MC-DCT 영역에서의 자막 처리 방법을 생각할 수 있다.

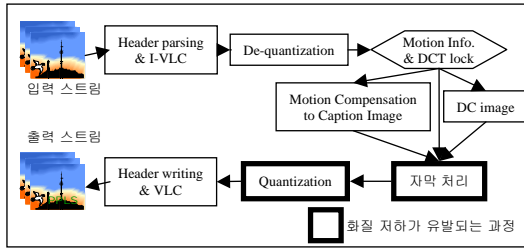


그림 4 MC-DCT 영역에서의 자막 처리 과정

<그림 4>는 본 논문에서 제안한 DC 이미지를 이용한 MC-DCT 영역에서의 자막 처리 과정을 나타낸다. MC-DCT 영역에서의 자막 처리는 입력 MPEG 스트림에 대해서 최소한의 디코딩 과정만을 거친 후 MC-DCT 영역에서 자막 처리를 수행하고 이것에 최소한의 인코딩 과정을 거쳐서 출력 MPEG 스트림을 생성하게 된다. 여기서 주목할 사항은 자막의 특성이다. 자막은 그 목적상 강한 값을 갖게 된다. 그렇기 때문에 MC-DCT 영역에서의 워터마크 방법을 그대로 사용할 경우 약한 자막이 되거나 강한 자막이 되더라도 심한 화질 저하 현상이 발생하게 된다. 이러한 문제점을 해결하기 위해서 MC-DCT 영역에서 쉽게 구할 수 있는 DC 이미지를 이용하여 화면 상태를 고려한 자막의 세기 조절을 하게 된다. 대략적인 과정을 살펴보면 I 프레임의 경우 DC 이미지에 따라서 자막의 세기를 조절하고 그 결과를 이산 코사인 변환 블록에 더해지게 된다. P, B 프레임의 경우는 참조 프레임에 더해진 자막의 영향을 제거하는 과정과 해당 프레임에 자막을 추가하는 과정으로 나누어 생각할 수 있다. 자막을 제거하는 과정에서는 참조 프레임에 사용된 DC 이미지를 사용하여 참조 프레임에 더해진 자막의 영향을 제거하고, 자막을 추가하는 과정에서는 해당 프레임의 DC 이미지에 따라서 자막의 세기를 조절하고 그 결과를 더하게 된다. 이러한 모든 과정은 MC-DCT 영역에서 얻을 수 있는 정보들을 대상으로 수행되며 그 결과에 대해서 부분적인 인코딩 과정을 거쳐서 출력 MPEG 스트림이 나오게 된다.

제 3 장에서는 앞에서 언급한 MC-DCT 영역에서의 자막 처리 과정에서 발생할 수 있는 문제점에 대해서 좀 더 자세히 알아보고 본 논문에서 제시한 해결책인 DC

이미지를 사용하는 방법에 대해서 설명하도록 하겠다.

3.1 MC-DCT 영역의 자막 처리 방법의 문제점

<그림 5>은 MC-DCT 영역에서의 자막 처리의 문제점과 그 원인을 나타낸 것이다. P, B 프레임을 처리하는 데 있어서 가장 중요한 원리는 참조 프레임에 더해진 자막 만큼을 빼주고 해당 자막을 더한다는 것이다. 그러나, <그림 5>에서 볼 수 있듯이 더해지는 자막의 값이 너무 클 경우 자막 처리 이전의 값을 제대로 복원할 수 없다는 문제점이 발생하게 된다. 이런 MC-DCT 영역에서의 자막 처리의 문제점을 공간 영역의 동작을 이용하여 설명하면 다음과 같다.

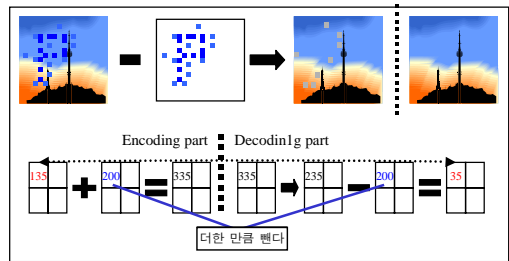


그림 5 MC-DCT 영역에서의 자막 처리의 문제점

<그림 5>에서와 같이 참조 프레임의 해당 픽셀 값이 135 이고, 이곳에 더해진 자막의 값이 200 일 경우 결과는 335가 된다. MPEG에서는 공간 영역의 Luminance의 범위를 16~235로 제한하여 사용하고 있기 때문에 이것은 Luminance의 제한 범위인 235를 벗어나는 값이 된다. P, B 프레임에서 참조 프레임에 더해진 자막의 영향을 제거하는 과정에서는 참조 프레임에 더해진 200을 빼게 된다. 문제가 발생하는 것은 이 픽셀을 참조하는 프레임의 디코딩 과정이다. 이 픽셀을 참조하는 프레임의 디코딩 과정에서 335를 235로 바꾸어서 참조 프레임 버퍼에 저장하게 되고, 이것으로부터 참조 프레임에 더해진 자막의 값, 즉 200을 빼게 되는 것이다. 그 결과 35 라는 값을 얻게 되며 이것은 우리가 기대하고 있는 값 135 보다 100 이 적은 값이 되게 된다. 이러한 오차의 영향으로 참조 프레임에 추가된 자막의 영향이 완전히 제거되지 못하고 남게 되는 것이다. 이러한 문제점은 워터마크의 삽입에 있어서는 워터마크의 세기가 그리 크지 않으므로 큰 문제가 되지 않지만 자막은 그 픽셀 값들이 대체로 크며 그로 인한 오차의 영향이 심각하게 된다.

3.2 DC 이미지를 이용한 해결 방법

3.1절에서 언급한 문제점을 해결하기 위해서는 더해지는 자막의 세기를 입력 프레임의 해당 픽셀 값에 따라 조정함으로써 해결할 수 있다. 즉, <그림 6>의 예를 보

면 자막을 더한 결과가 Luminance의 제한 범위인 235를 넘지 않게 하기 위해서 자막의 세기를 200에서 100으로 조절하는 것이다.

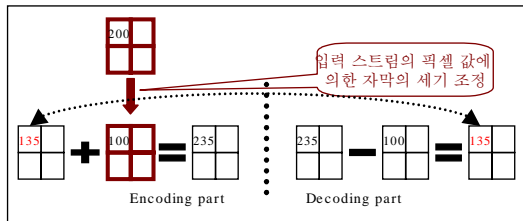


그림 6 MC-DCT영역에서의 자막 처리의 문제점에 대한 해결 방법

이러한 과정을 수식으로 일반화하면 식 (6)와 같다.

$$x'_i = x_i + y_i * \left(1 - \frac{x_i}{235}\right) \quad (6)$$

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  $x'$ ,  $x$ ,  $y$ 블록들의  $i$  번째 픽셀 값이다. <그림 7>는 식 (6)에 나타난 수식에서  $y_i$ 를 상수로 생각했을 경우의 그래프를 나타낸다. 그래프에서 볼 수 있듯이 식 (6)은 자막 처리 결과 픽셀의 Luminance 값을 MPEG에서의 제한 범위인 235 이내로 제한하는 기능과 함께  $x_i$ 의 값이 커질수록 자막으로 더해지는 값이 줄어드는 특성을 보인다.

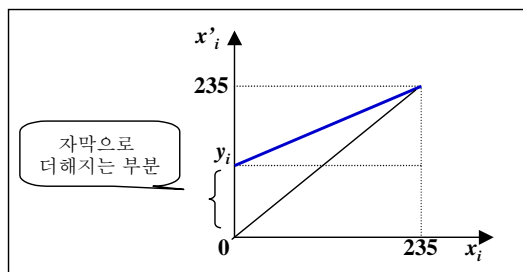


그림 7 식 (6)의 그래프

여기서 중요한 것은 각 픽셀의 값을 알아야 하는 것인데, MC-DCT 영역에서는 이런 값을 공간 영역까지 디코딩하기 전에는 알 수가 없다. 본 논문에서는 이런 문제를 해결하기 위해서 DC 이미지를 이용해서 픽셀 값을 블록 단위로 근사하는 방법을 사용하였다. 즉, 이러한 이유로 해서 식 (6)에서의  $y_i$ 값을 조정하기 위해서 사용된  $x_i$

를  $x$  블록의 평균  $e(x)$ 으로 근사하여 식 (7)을 얻게 된다.

$$x'_i = x_i + y_i * \left(1 - \frac{e(x)}{235}\right) \quad (7)$$

식 (6)을 식 (7)로 근사함으로써 발생하게 되는 오차는  $x_i < e(x)$ 인 경우와  $x_i > e(x)$ 인 경우로 나누어 생각할 수 있다.  $x_i < e(x)$ 인 경우는  $y_i * \frac{x_i - e(x)}{235}$  만큼 덜 더해지지만  $x'_i$ 의 값이 235 이하가 된다는 점에서 인터 매크로 블록의 처리에 문제점이 되지 않는다. 그러나,  $x_i > e(x)$ 인 경우는  $y_i * \frac{x_i - e(x)}{235}$  만큼 더 더해지는 결과가 발생하게 되며, 이것은  $x'_i$ 의 값이 235 이상이 될 가능성의 원인이 되며 인터 매크로블록에서 완벽하게 복원되지 못하고 화질 저하를 유발하는 원인이 된다. 식 (7)을 블록 단위의 식으로 변환하면 식 (8)와 같이 될 수 있다.

$$x' = x + y * \left(1 - \frac{e(x)}{235}\right) \quad (8)$$

이렇게 얻어진 식 (8)은 다음의 DCT 변환의 특성[7]을 이용하여 DCT 영역의 수식으로 변환될 수 있다. 다음 수식에서  $a$ ,  $b$ 는 공간 영역의 블록이며,  $A$ ,  $B$ 는 DCT 변환 이후의 블록이다.  $k$ 는 상수를 나타낸다.

$$\begin{aligned} a + b &\leftrightarrow A + B \\ ka &\leftrightarrow kA \\ DC(A) &= e(a) * 8 \end{aligned}$$

식 (9)은 DCT 영역에서 쉽게 구할 수 있는 입력 프레임의 블록들의 DC 값을 이용하여 자막의 세기를 조절하는 수식을 보여주고 있다.

$$X' = X + Y * \left(1 - \frac{DC(X)}{1880}\right) \quad (9)$$

식 (9)에서 사용되고 있는 DC 값은 2.2.2 절에서 설명한 방법을 통하여 I, P, B 프레임 모두에서 간단한 계산을 통하여 구할 수 있다. 이렇게 구해진 DC 이미지에 따라서 자막의 세기를 조절하게 되며, P, B 프레임에서 참조 프레임의 자막을 제거하는 과정에서는 참조 프레임의 DC 이미지를 이용하여 제거할 자막의 세기를 조절하게 된다.

3.3 전체 알고리즘

<그림 8>과 <그림 9>에는 이상에서 설명한 DC 이미지를 이용한 MC-DCT 영역에서의 자막 처리 방법을 인트라 프레임에 대한 처리 방법과 인터 프레임에 대한 처리 방법으로 나누어서 나타내었다.

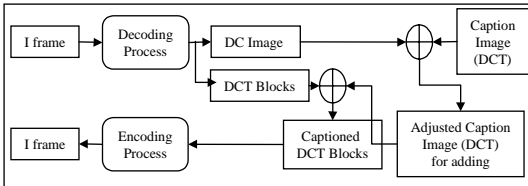


그림 8 DC 이미지를 이용한 I 프레임에 대한 자막 처리

I 프레임은 다른 프레임을 참조하지 않고 독립적으로 인코딩이 되어 있다. 즉, 모든 매크로블록이 인트라 코딩 방법으로 인코딩 되어 있기 때문에 단순히 이산 코사인 변환 블록에 자막을 더해주기만 하면 된다. 이 때 이 프레임을 참조할 프레임들을 위해서 더해지는 자막의 세기를 DC 이미지를 참조하여 조절하는 기능만 추가하면 된다. <그림 8>에서 보면 이산 코사인 변환 상태의 자막 이미지에 DC 이미지를 참조하여 실제로 더해질 자막의 세기를 조절하고 있다. 또한 DC 이미지는 이 프레임을 참조하는 P 프레임이나 B 프레임에서 참조 프레임에 더해진 자막의 영향을 제거하는 과정에서도 사용되게 된다. DC 이미지에 따라서 그 세기가 조절된 이산 코사인 변환 영역의 자막 이미지는 입력 프레임에 부분적인 디코딩만을 거쳐서 얻어진 이산 코사인 변환 블록들에 더해져서 출력 프레임을 구성하게 된다.

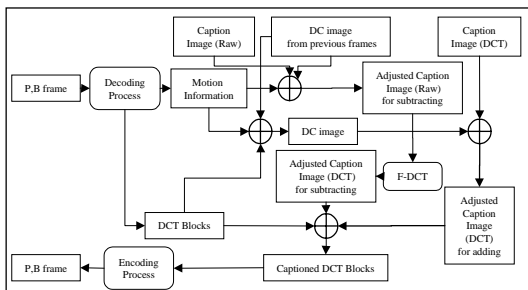


그림 9 DC 이미지를 이용한 P, B 프레임의 자막처리

인터 프레임, 즉 P, B 프레임에 대한 자막 처리는 참조 프레임에 더해진 자막을 제거하는 과정과 새로이 자막을 더하는 과정으로 나누어서 생각할 수 있다. 이 과정들을 <그림 9>을 참조하여 보다 자세히 살펴보기로 한다. 우

선 입력 프레임으로부터 부분적인 디코딩 과정을 거쳐서 움직임 정보와 예측 에러 블록 정보(DCT 블록)를 얻을 수 있다. 참조 프레임에 추가된 자막의 영향을 제거하기 위해서 움직임 정보와 자막 이미지 그리고 참조 프레임에 사용되었던 DC 이미지를 이용하여 자막 제거용 이미지를 만든다. 이렇게 만들어진 자막 제거용 이미지를 입력 프레임의 예측 에러 블록에서 빼 줌으로써 참조 프레임의 자막의 영향을 제거할 수 있게 된다. 입력 프레임에 더해질 자막을 생성하는 것은 인트라 프레임에서와 마찬가지로 DC 이미지와 자막 이미지를 이용하여 만들어진다. 참조 프레임에 처리된 자막의 영향을 제거하기 위해 제조된 자막과 더해질 자막, 그리고 입력 프레임으로부터 얻은 예측 에러 블록들을 더하여 자막 처리가 된 블록을 얻게 된다.

인코딩 과정은 자막 제거 및 추가 과정을 거쳐서 얻어진 블록들과 입력 프레임에 대한 디코딩으로 얻은 움직임 정보를 이용하여 수행되게 된다. 자막 처리 과정을 거치면서 입력 프레임의 매크로블록이 가지고 있던 타입이 변하게 되는 데 이것은 존재하지 않던 예측 에러 블록이 발생하기도 하고 기존의 예측 에러 블록이 모두 0으로 되어서 사라지기도 하기 때문이다. 이러한 예측 에러 블록의 변화에 따라서 매크로블록 타입을 적절히 변환시켜 주게 된다.

4. 성능 평가 및 분석

본 논문에서 제안한 MC-DCT 영역에서의 자막 처리 방법을 실제 구현을 통하여 자막 처리로 인한 화질의 변화와 그 수행 시간을 측정하여 각각의 성능을 평가 비교하였다. 또한 본 논문에서 제안한 방법과 비교 대상이 되었던 방법들, 즉 공간 영역에서의 자막 처리 방법과 I 프레임으로 변환하여 자막 처리하는 방법을 함께 구현하여 그 성능을 비교 분석함으로써 제안한 방법의 우수성을 보이도록 하겠다<sup>2)</sup>. 그리고, DC 이미지를 사용하지 않은 상태의 MC-DCT 영역에서의 자막 처리 결과를 함께 보임으로써 DC 이미지의 필요성을 보이도록 하겠다. 언급한 3 가지 자막 처리기는 모두 MPEG 1 비디오 스트림을 입력으로 받아서 자막을 처리한 후 다시 MPEG 1 비디오 스트림을 출력하도록 구현하였다. 이 때 자막 이미지는 PPM 형태로 제작된 동일한 이미지를 사용하였다. 화질의 비교는 각 방법들에 의해서 얻어진 MPEG 비디오 스트림을 동일한 디코더[14]를 이용하여 PPM 형태의 파

2) 단, 모든 자막 처리 방법은 워터마크 기법을 응용한 방법을 사용하였다.



일들로 만들고 이것에 대해서 수행하였다. 화질을 비교하기 위한 기준으로 사용된 이미지는 자막 처리 이전의 MPEG 비디오 스트림을 PPM 형태로 출력하여 공간영역에서 자막 처리를 수행한 것을 사용하였다. 화질의 비교를 위하여 사용된 수식은 식 (10)에 나타나 있는 PSNR(Peak Signal-To-Noise Ratio) 값을 이용하였다. 식 (10)에서 M, N은 비교하는 그림의 가로 및 세로의 길이이고, X 및 Y는 비교하는 두 그림의 좌표 i, j에서의 화소값이다.

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - Y_{ij})^2} \text{ dB} \quad (10)$$

성능 평가를 위해서 I 프레임 1 개, P 프레임 4 개, B 프레임 10 개로 구성된 GOP를 갖는 MPEG 스트림을 입력으로 사용하였고, 100 프레임에 대상으로 하여 자막 처리를 수행하였다.

4.1 화질 비교

본 절에서는 화면의 각 부분별 화질 변화를 각 방법별로 비교하고, 그 원인을 분석하기로 한다. 우선 화면의 부분별 비교는 자막 픽셀에 대한 화질 비교, 화질 저하 예상 구역(자막 픽셀 제외)에 대한 화질 비교, 화질 저하 예상 구역을 제외한 부분에 대한 화질 비교, 화면 전체에 대한 화질 비교로 나눠서 알아보겠다. <그림 10>은 화면의 각 부분에 대한 예를 나타낸다.

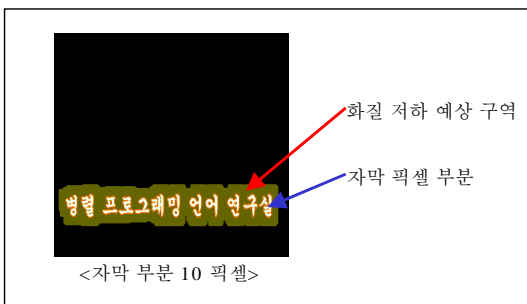


그림 10 화면의 구역 구분

4.1.1 자막 픽셀에 대한 화질 비교

<그림 11>은 실제 자막이 첨가된 픽셀들에 대한 화질 비교 실험 결과를 나타낸 것이다. 실험에 의하면 공간 영역에서의 자막 처리 방식이나 I 프레임으로 변환한 후에 자막 처리를 한 방법이 본 논문에서 제안한 방법보다 우수한 화질을 보장하는 것을 볼 수 있다. DC 이미지를 사용하지 않는 MC-DCT 영역에서는 I 프레임에 제외하

나지 않는 DC 이미지를 사용하지 않는 MC-DCT 영역에서의 자막 처리 결과가 공간 영역의 자막 처리 방법 보다 낮은 PSNR을 보이는 것은 DC 이미지에 기초해서 더해지는 자막의 세기를 줄였기 때문으로 볼 수 있다. 이것은 인터 프레임들에서 참조 프레임에 처리된 자막을 효과적으로 제거하기 위해서 DC 이미지를 사용했기 때문에 나타나는 현상으로 I, P, B 프레임 모두에서 비슷한 PSNR 값을 보여주고 있다.

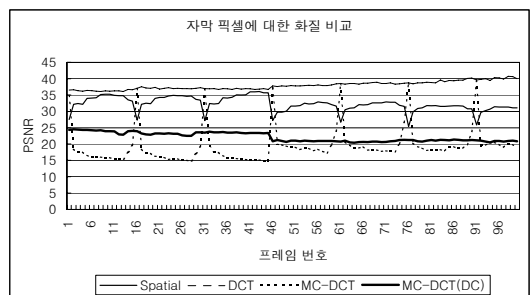


그림 11 자막 픽셀에 대한 화질 비교

4.1.2 화질 저하 예상 구역(자막 픽셀 제외)에 대한 화질 비교

<그림 12>은 자막이 추가됨으로써 화질의 저하를 보일 것으로 예상되는 구역에 대한 화질 비교를 나타낸 것이다. DC 이미지를 사용하지 않는 MC-DCT 영역의 방법을 제외하고는 대체로 비슷한 PSNR을 나타내고 있다. 특히 DC 이미지를 사용하는 MC-DCT 영역의 방법의 결과를 보면 I 프레임에서는 비교적 높은 PSNR을 보여주고 있으며, P, B 프레임에서는 약간 낮은 PSNR을 보여주는 것을 알 수 있다. 이것은 P, B 프레임에서 참조 프레임에 추가된 자막을 제거하는 과정에서 약간의 오차가 발생하고 있음을 보여주고 있는 것이다. 이것은 3.1 절에서 설명했던 평균보다 큰 값을 갖는 픽셀에 대해서 발생하는 오차의 영향이 참조 프레임의 자막 부분에 대한 움직임 보상 영역에 걸쳐서 나타나기 때문으로 P, B 프레임의 PSNR이 I 프레임의 PSNR 보다 낮게 나타나는 원인이 된다. DC 이미지를 사용하지 않는 MC-DCT 영역에서의 방법을 보면 P, B 프레임에서 매우 낮은 PSNR을 보여주고 있다. 이것은 DC 이미지를 사용함으로써 화질 저하 예상 구역에 대한 처리가 잘 되었음을 알 수 있게 해준다.

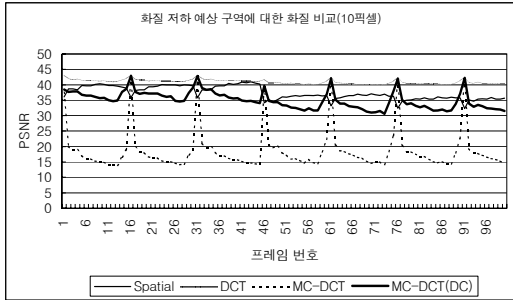


그림 12 화질 저하 예상 구역에 대한 화질 비교(10픽셀)

4.1.3 화질 저하 예상 구역을 제외한 부분에 대한 화질 비교

<그림 13>은 화질 저하 예상 구역을 제외한 나머지 부분에 대한 화질 비교 결과를 나타낸다. MC-DCT 영역에서의 방법이 월등히 높은 PSNR을 나타내는 것을 볼 수 있다. 이것은 공간 영역을 거치지 않고 압축된 상태에서 자막을 처리하고 다시 압축하는 방법을 사용했기 때문에 화질 저하 구간을 최소화한 결과로 볼 수 있는 것이다. 실제로 단순히 MC-DCT 영역까지 디코딩한 후 이것을 다시 인코딩하는 과정은 화질의 손실을 발생시키지 않고 진행될 수 있다. 그렇기 때문에 자막 처리에 따른 영향을 받지 않은 부분은 자막 처리 이전과 동일하게 되며 그로 인해서 높은 PSNR을 나타내게 되는 것이다. P, B 프레임에 있어서 I 프레임보다 낮은 PSNR을 보이는 이유는 P, B 프레임에서 참조 프레임에 추가된 자막의 영향을 제거하는 과정에서 일부 제거되지 못한 부분이 있기 때문이다. 이 때 P, B 프레임에서 참조 프레임의 자막 부분을 참조하는 매크로블럭이 많을수록 PSNR 값이 낮을 확률이 높아지게 된다. <그림 13>에서 초반부의 PSNR 값이 후반부 보다 높은 이유는 초반부는 움직임 벡터의 영향권이 비교적 좁은 프레임들이었으며, 후반부는 움직임 벡터의 영향권이 비교적 넓은 프레임들이었기 때문이다. 또한 하나의 GOP 내에서 PSNR의 변화 추이를 보면 GOP의 후반부로 갈수록 PSNR이 낮아지는 것을 볼 수 있다. 이것은 참조 프레임의 자막을 제거하는 과정에서 남은 오차가 후반부의 프레임으로 갈수록 누적되며 또한 움직임 보상 영역이 넓어지기 때문인 것으로 볼 수 있다.

4.1.4 화면 전체에 대한 화질 비교

<그림 14>는 화면 전체에 대한 화질 비교 결과를 나타낸 것이다. 대체로 공간 영역에서의 자막 처리나 I 프레임으로의 변환을 통한 자막 처리가 MC-DCT 영역에서

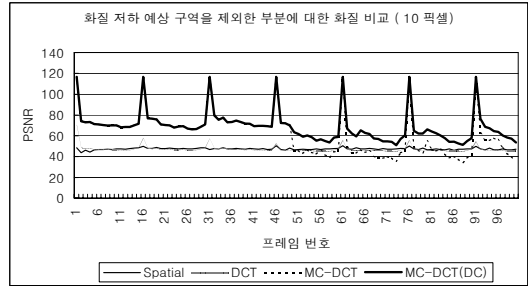


그림 13 화질 저하 예상 구역을 제외한 부분에 대한 화질 비교(10 픽셀)

의 자막 처리보다 높은 PSNR을 보여주고 있다. 특히 DC 이미지를 이용하지 않은 MC-DCT 영역에서의 자막처리는 I 프레임을 제외한 대부분의 구간에 있어서 현저히 낮은 PSNR값을 나타내고 있다. 이것은 참조 프레임에 처리된 자막의 영향을 제대로 제거하지 못했기 때문에 발생하는 것이다. 반대로 I 프레임에 있어서는 제안한 방법보다 우수한 결과를 나타내고 있는데 이것은 제안한 방법과는 달리 자막의 세기를 DC 이미지에 따라 줄이지 않았기 때문이다. DC 이미지를 이용한 MC-DCT 영역에서의 자막처리는 전반적으로 공간 영역에서의 자막 처리보다 낮은 PSNR을 나타내고 있지만 DC 이미지를 이용하지 않은 MC-DCT 영역의 자막 처리보다는 월등히 좋은 성과를 보여주고 있다.

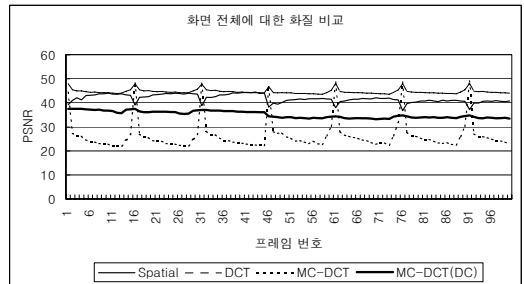


그림 14 화면 전체에 대한 화질 비교

4.1.5 화질 비교에 대한 분석 결과

<그림 15-1>과 <그림 15-2>은 각 방법 별 자막 처리 결과를 시각적으로 나타낸 것이다.

4.1.4에서 살펴본 바와 같이 프레임 전체에 대한 화질 비교에 있어서는 공간 영역에서의 자막 처리 방법과 DCT 영역에서의 자막 처리 방법이 DC 이미지를 이용한 MC-DCT 영역에서의 처리 방법보다 높은 PSNR을 나타



그림 15-1 각 방법 별 결과 (오른쪽 그림 : 자막 처리 이전 프레임과의 차이)



그림 15-2 각 방법 별 결과 (오른쪽 그림 : 자막 처리 이전 프레임과의 차이)

내고 있는 것을 볼 수 있다. 그러나, <그림 15-1>에서 볼 수 있듯이 공간 영역에서의 자막 처리 방법과 DCT 영역에서의 자막 처리 방법은 프레임 전체에 걸쳐서 화질이 저하되었다는 것을 알 수 있다. 이것은 자막 처리를 위해서 디코딩과 인코딩의 모든 과정을 거쳤기 때문에 발생한 것으로서 특히 양자화 과정에서 발생한 오차의 영향으로 프레임 전체에 걸친 화질 저하 현상을 보여주고 있는 것이다. 반면 <그림 15-2>에서 볼 수 있듯이 MC-DCT 영역에서의 자막 처리 방법은 자막 부근에서는 화질 저하가 발생하고 있지만, 화질 저하 예상 구역을 제외한 부분에 대해서는 자막 처리 이전의 스트림과 동일한 값을 갖고 있음을 확인 할 수 있다. 또한 DC 이미지를 사용하지 않은 MC-DCT 영역에서의 자막 처리 방법에서 볼 수 있는 자막 부근의 심한 화질 저하 현상이 DC 이미지를 사용하여 블록 단위로 자막의 세기를 조절

함으로써 급격히 감소했음을 알 수 있다.

#### 4.2 수행 시간 비교

본 절에서는 자막 처리를 위한 4 가지 방법들의 수행 속도를 비교함으로써 제안한 자막 처리 방법이 수행 속도 면에서 우수함을 보이려 한다. <그림 16>에는 평균 자막 처리 시간을 각 방법별로 나타내고 있다.

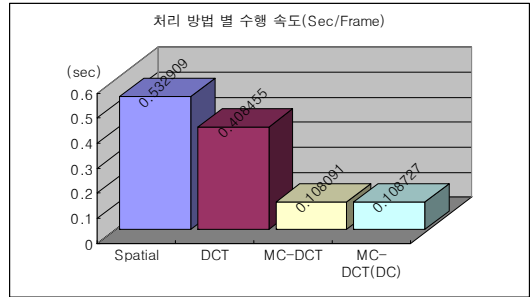


그림 16 자막 처리 방법 별 프레임 당 자막 처리 속도

이산 코사인 변환 영역에서의 처리가 공간 영역에서의 처리보다 1.3 배 정도 빠른 것을 볼 수 있다. 이것은 인코딩 과정에서 P, B 프레임의 움직임 예측 과정이 생략된 결과로 볼 수 있다. MC-DCT 영역에서의 처리는 공간 영역에서의 처리 보다 4.9 배 정도 빠른 것으로 나타났다. 이것은 많은 수행 시간이 요구되는 부분들, 즉 IDCT, DCT, 움직임 예측 등의 과정을 생략함으로써 얻어진 결과이다.

### 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 DC 이미지를 이용한 MC-DCT 영역에서의 자막 처리 방법을 제안하고 그 성능을 기존의 다른 자막 방법과 비교 분석하였다. 제안한 방법은 MC-DCT 영역에서의 워터마킹 기법을 응용하였으며, 자막의 특성으로 인해서 발생하는 문제점을 DC 이미지를 이용하여 해결하고 있다. 자막은 픽셀 값이 강하기 때문에 P, B 프레임에서 참조 프레임에 더해진 자막을 제거하는 과정에서 오차가 발생하게 되며 이것은 심각한 화질 저하의 원인이 된다. 제안한 방법에서는 이러한 문제점을 해결하기 위하여 MC-DCT 영역에서 쉽게 구할 수 있는 DC 이미지

를 이용하여 자막의 세기를 조절하는 방법을 제안하고 있다.

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 않았다. 공간 영역에서의 자막 처리 방법과 비교해 볼 경우 자막과 자막 부분의 화질 비교에 있어서는 본 논문에서 제안한 방법의 성능이 뒤떨어지지만 자막 부분을 제외한 나머지 부분에 대해서는 월등히 높은 성능을 보이고 있다. 제안한 방법에서 자막 부분의 화질이 공간 영역의 방법을 이용한 것 보다 나쁜 이유는 DC 이미지에 따라서 자막의 세기를 조절했기 때문이다. 반면 MC-DCT 영역에서의 처리를 채택함으로써 자막 부근을 제외한 영역에서는 원본과 동일한 화질을 얻을 수 있었다.

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 처리 할 수 있을 것이다.

### 참 고 문 헌

- [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] 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.
- [3] 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.
- [4] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System," Proc. of ACM Multimedia 96 Conference, Nov. 1996.
- [5] 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.
- [6] Bo Shen and Ishwar K. Sethi, "Inner-Block Operations On Compressed Images," Proc. of ACM Multimedia 95, Nov. 1995.
- [7] Neri Merhav and Vasudev Bhaskaran, A Fast Algorithm for DCT-Domain Inverse Motion Compensation, HPL Technical Report #HPL-95-17, Sep. 1995.
- [8] Soam Acharya and Brian Smith, "Compressed Domain Transcoding of MPEG," Proc. of IEEE International Conference on Multimedia Computing and Systems, Jul. 1998.
- [9] 정제창, 그림으로 보는 최신 MPEG, 교보문고, Dec. 1995.
- [10] Jianhao Meng, Yujen Juan and Shih-Fu Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," Proc. of IS&T/SPIE Symposium on Digital Video Compression: Algorithms and Technologies, Vol. 2419, Feb. 1995.
- [11] Boon-Lock Yeo and Bede Liu, "A Unified Approach to Temporal Segmentation of Motion JPEG and MPEG Compressed Video," Proc of International Conf. on Multimedia Computing and Systems, May 1995.
- [12] Vrkrant Kobra and David Doerman, "Compressed Domain Video Indexing Techniques Using DCT and Motion Vector Information in MPEG Video," Proc. of SPIE Conf. on Storage and Retrieval for Image and Video Databases V, Vol. 3022, Feb.1997.
- [13] A. R. Smith and J. F. Blinn, "Blue Screen Matting," Proc. of SIGGRAPH'96, Aug. 1996.
- [14] MPEG Software Simulation Group, MPEG-2 Video Codec, Available from <http://www.mpeg.org/MPEG/MSSG>.



권 오 형

1998년 2월 서강대학교 전자계산학과 졸업(학사). 2000년 2월 서강대학교 대학원 컴퓨터학과 졸업(석사). 현재 (주)한화/정보통신 연구원. 관심분야는 멀티미디어 시스템, 무선 인터넷(WAP) 등.

남 종 호

정보과학회논문지 : 소프트웨어 및 응용 제 27 권 제 8 호 참조