

論文 2002-39SP-6-11

## 고속 압축변환기를 위한 MPEG-2 복호기

## (MPEG-2 Decoder with Down-sampling for Fast Transcoding)

吳承均\*, 朴玄旭\*\*

(Seung-Kyun Oh and HyunWook Park)

## 요 약

고속으로 압축변환을 하기 위해 DCT(Discrete Cosine Transform) 영역에서 영상의 해상도를 줄이는 MPEG-2 복호기를 제안한다. 제안하는 복호기는 움직임 보상을 하기 전에 DCT 영역에서 영상의 해상도를 줄이며 해상도가 줄어든 영상에 대해서 DCT 영역에서 움직임 보상을 수행한다. 본래 해상도의 영상 ( $N \times M$ )에서는  $\frac{1}{2}$  화소 정밀도의 움직임 보상을 해야 하지만 해상도가 반으로 줄어든 영상 ( $N/2 \times M/2$ )에 대해서는  $\frac{1}{4}$  화소 정밀도의 움직임 보상을 수행해야 한다. 하지만 해상도가 줄어든 영상에 대해  $\frac{1}{4}$  화소 정밀도의 움직임 보상을 수행하면 영상이 흐릿해지는 현상이 발생할 수 있다. 본 논문에서는 해상도를 줄인 영상에 대해 계산시간 뿐만 아니라 DCT 영역에서의 움직임 보상 시에 발생 할 수 있는 흐려짐도 줄일 수 있는 방법을 제안한다. 실험 결과는 제안한 방법이 높은 해상도의 영상을 낮은 해상도의 영상으로 효율적으로 복호할 수 있는 알고리즘이라는 것을 보여준다.

## Abstract

We propose an MPEG-2 decoder with down-sampling in the Discrete Cosine Transform (DCT) domain for fast transcoder. It resizes images in the DCT domain before taking motion compensation (MC). After resizing the images, the motion compensation in the DCT domain (MC-DCT) is performed to the down-sampled image. If the MC was performed with a half-pixel precision in a full resolution image ( $N \times M$ ), the MC in a two-fold down-sampled image ( $N/2 \times M/2$ ) should be done with a quarter-pixel precision. However, image blurring happens for MC-DCT with a quarter-pixel precision in the down-sampled image. We propose a high precision method that can reduce not only computation time but also blurring of MC-DCT at down-sampled image. The experimental results show that the proposed algorithm can efficiently decode a high resolution bitstream to a low resolution bitstream of MPEG-2.

**Keywords** : 변환부호화 (Transcoding), 해상도 변환 (Image resizing), 움직임 보상 (Motion compensation), 이산여현변환 (Discrete Cosine Transform), 흐려짐 (Blurring).

\* 學生會員, \*\* 正會員, 韓國科學技術院 電子電算學科  
(Department of Electrical Engineering and Computer Science, Division of Electrical Engineering, Korea Advanced Institute of Science and Technology)

※ 이 연구는 정보통신부 정보통신 분야 인력 양성사업의 지원으로 수행되었습니다.

接受日字: 2002年7月15日, 수정완료일: 2002年10月16日

## I. 서 론

디지털 기술의 눈부신 발전과 함께 네트워크에서 점점 더 많은 디지털 데이터들이 사용되고 있다. 디지털 TV, 무선 이동 통신, 주문형 비디오 서비스 등에서 디지털 비디오가 사용된다. 하지만 이러한 많은 디지털

비디오의 사용은 네트워크의 부담을 가중시키기 때문에 데이터양을 줄 일 필요가 있다. 압축변환은 하나의 영상 비트열을 다른 영상 압축표준이나 다른 해상도를 갖도록 변환하는 것을 말한다. 본 논문에서 주목하는 것은 데이터의 양을 줄이기 위해 해상도 변경을 할 수 있는 압축변환기이다. 압축변환기는 압축변환을 수행하는 영역에 따라서 크게 두 가지로 나눌 수 있다. 하나는 공간 영역에서 압축변환을 수행하는 것이고<그림 1> 다른 하나는 DCT 영역에서 압축변환을 수행하는 것이다.

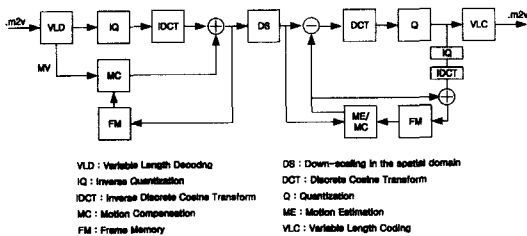


그림 1. 공간 영역에서의 압축변환기  
Fig. 1. Conventional spatial domain transcoder.

공간영역에서의 압축변환기는 공간영역에서 DCT, 역 DCT, 움직임 추정을 수행하여야 한다. 이러한 부분은 계산량을 많이 요구하기 때문에 고속의 압축변환을 위해서는 고속의 알고리즘이 요구된다. 본 논문에서는 효율적인 계산과 계산 시간을 줄이기 위해서 DCT 영역에서의 압축변환기에 주목하기로 한다.

DCT 영역에서 해상도를 줄일 수 있는 압축변환기는 DCT 영역에서 영상의 해상도를 변경할 수 있고 움직임 보상을 할 수 있는 MPEG-2 복호기가 필요하다. 그동안 영상의 해상도를 변경 할 수 있는 MPEG-2 복호기에 대한 연구가 진행되었다.<sup>[1,2]</sup> 진행되어 온 연구에서는 DCT 영역에서 영상의 해상도를 변경하고 공간영역에서  $\frac{1}{4}$  화소 정밀도의 움직임 보상을 수행하는 방법이다.

하지만 완전한 DCT 영역에서의 압축변환을 위해서는 움직임 보상 또한 DCT 영역에서 이루어져야 한다. DCT 영역에서 영상의 해상도를 변경할 수 있는 압축변환기에 대한 연구도 진행되어 왔다.<sup>[3-6]</sup> 이들 연구에서는 움직임 보상을 수행한 후에 영상의 해상도를 변경하는 방법인데 반해, 본 논문에서 제안하는 방법은 움직임 보상을 수행하기 전에 영상의 해상도를 변경하

는 방법이다.

본 논문에서는 고속의 MPEG-2 압축변환기를 위해 DCT 영역에서 영상의 해상도를 변경할 수 있는 MPEG-2 복호기를 제안한다. II장에서는 제안하는 MPEG-2 복호기의 구조에 대해서 설명한다. III장에서는 DCT 영역에서  $\frac{1}{4}$  화소 정밀도를 가지는 움직임 보상에 대해서 살펴본다. IV장에서는 대칭 중첩합 (Symetric convolution)을 사용하여 DCT 영역에서 해상도를 변경하는 알고리즘을 설명하고 V장에서는 실험 결과에 대해 설명한다. 마지막으로 VI장에서는 본 논문의 결론을 맺는다.

## II. 제안하는 MPEG-2 복호기

제안하는 고속의 MPEG-2 압축변환기를 위한 MPEG-2 복호기에서는 해상도를 줄이는 것과  $\frac{1}{4}$  화소 정밀도를 가진 움직임 보상이 모두 DCT 영역에서 수행된다. 기존 대부분의 압축변환기에서는 본래 해상도에서 움직임 보상을 한 후 해상도를 축소하였다. 본 논문에서는 압축변환기의 속도를 향상시키기 위해서 본래 해상도에 대해서 영상의 해상도를 변경한 후 변경된 해상도의 영상 데이터에 대해서 움직임 보상을 수행하는 방법을 제안한다. 기존의 방법과 제안하는 방법의 차이는 <그림 2>에 나타나 있다.

영상 시퀀스의 첫 부분인 I 영상은 움직임 보상이 필요 없기 때문에 부 표본화만 수행한다. 이렇게 해상도

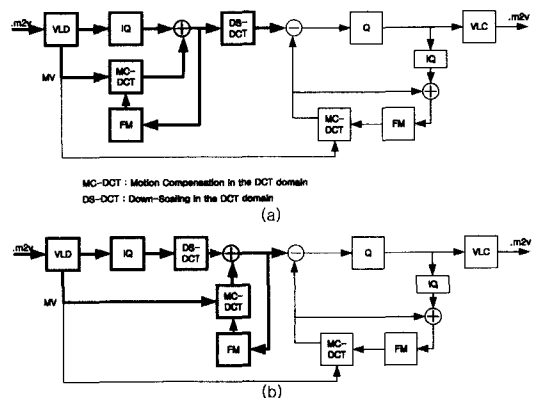


그림 2. (a) 기존의 DCT 영역에서의 압축변환기  
(b) 제안하는 DCT 영역에서의 압축변환기  
Fig. 2. (a) Conventional DCT domain transcoder  
(b) Proposed DCT domain transcoder

가 줄어든 I 영상은 이어서 나오는 P 영상의 참조 영상으로 사용된다. 첫 번째 P 영상 복호 시에 생성된 오차 영상을 부 표본화 하고 참조 영상을 이용하여 구한 움직임 보상된 영상과 합하여 첫 번째 P 영상을 복원해 낼 수 있다. 뒤에 이어서 나오는 영상들도 마찬가지로 처리 과정을 통하여 복원이 가능하다. 이러한 처리가 모두 DCT 영역에서 일어나며 움직임 보상을 수행 할 시에 본래의 움직임 보상과는 다르게 화소의 정밀도가 높아진 상태에서 움직임 보상을 수행 하여야 한다. 본문에서 제안하는 MPEG-2 복호기는 움직임 보상 전에 영상의 해상도를 줄이고 줄어든 영상에 대해서 움직임 보상을 수행한다. 그리고 DCT 영역에서의 움직임 보상을 위해  $\frac{1}{4}$  화소 정밀도를 사용한다. DCT 영역에서의  $\frac{1}{4}$  화소 정밀도를 가지는 움직임 보상 방법에 대해서는 다음 장에서 자세히 설명하기로 한다.

### III. DCT 영역에서 $\frac{1}{4}$ 화소 정밀도를 갖는 움직임 보상

본 논문에서 제안하는 MPEG-2 복호기에서는  $\frac{1}{2}$  화소 정밀도가 아닌  $\frac{1}{4}$  화소 정밀도를 갖는 움직임 보상 방법이다. 그 이유는 본래의 해상도를 갖는 영상에서의 움직임 추정 방법은  $\frac{1}{2}$  화소의 정밀도를 가지고 움직임 추정과 보상을 수행한다. 하지만 제안하는 방법에서는 움직임 보상을 수행하기 전에 영상의 해상도를 반으로 줄인 후 움직임 보상을 수행하게 된다. 이렇게 되면  $\frac{1}{2}$  화소 단위의 움직임 벡터는  $\frac{1}{4}$  화소 단위의 움직임 벡터가 된다. 따라서 움직임 보상도  $\frac{1}{4}$  화소의 정밀도로 수행되어야 한다.

제안하는 방법은 DCT 영역에서의 처리를 하기 때문에 움직임 보상 역시 DCT 영역에서 수행된다. DCT 영역에서의 움직임 보상 방법은 Chang과 Messerschmitt에 의해서 연구되었다.<sup>[3]</sup> 그리고 공유하는 블록의 정보를 이용하여 고속의 움직임 보상 방법에 관한 연구도 진행되었다.<sup>[7]</sup>

움직임 보상을 행렬 형태의 표현식으로 보면 블록 내에 있는 필요한 부분을 새로운 블록에 이동시키는 것이다. 만약 현재 움직임 보상을 해야 하는 블록 P의

움직임 벡터가 모두 0이 아니라 해보자. 그러면 <그림 3>과 같이 참조 영상 내에 있는 네 개의 블록에 걸쳐 있는 부분이 필요하게 된다.

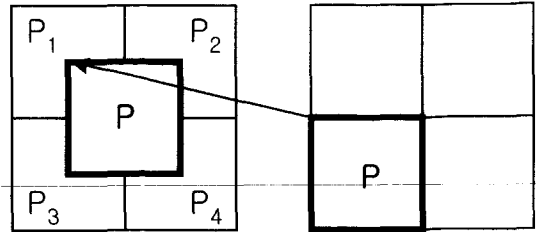


그림 3. 움직임 보상의 예  
Fig. 3. An example of motion compensation.

$$P = \sum_{i=1}^4 S_{i1} \times P_i \times S_{i2} \tag{1}$$

식 (1)에서 행렬  $S_{i1}$ 과  $S_{i2}$ 는 행렬  $P_i$ 를 적절하게 이동시킨 후 합하면 블록 P를 구할 수 있다. 움직임 벡터가 정수 화소 단위라면  $S_{i1}$ 과  $S_{i2}$ 는 0과 1로만 이루어지나  $\frac{1}{2}$  화소 단위라면 0, 0.5, 1로 이루어진다. 0.5의 값은 움직임 벡터가  $\frac{1}{2}$  화소일 경우 두 화소의 선형 보간으로 구하기 위한 값이다.

공간영역이 아닌 DCT 영역에서는 움직임 보상에 의한 결과 값 역시 DCT 영역에 있다. DCT 영역에서의 움직임 보상을 행렬 형태로 표현하면 다음과 같다.

$$DCT(P) = \sum_{i=1}^4 DCT(S_{i1}) \times DCT(P_i) \times DCT(S_{i2}) \tag{2}$$

여기서  $S_{ij}$ 는 다음과 같은 이동 행렬이다.

$$L_n = \begin{pmatrix} 0 & 0 \\ I_n & 0 \end{pmatrix} \text{ or } R_n = \begin{pmatrix} 0 & I_n \\ 0 & 0 \end{pmatrix} \tag{3}$$

각  $I_n$ 은 크기가  $n$ 인 항등 행렬이다. DCT 영역에서의  $\frac{1}{2}$  화소 정밀도의 움직임 벡터를 고려한 움직임 보상 시에도  $L_n$ 과  $R_n$ 의 수정을 통해 구할 수 있다.<sup>[8]</sup>

$\frac{1}{4}$  화소 정밀도를 갖는 움직임 보상도  $L_n$ 과  $R_n$ 의 수정을 통해서 구현된다. 해상도를 반으로 줄인 영상에서 움직임 벡터가  $(v_x, v_y)$ 이고 다음과 같은 관계가 성립한다고 가정하자.

$$v_x - |v_x| = a, \quad v_y - |v_y| = b \tag{3}$$

여기서 연산자 |저는  $x$ 보다 작거나 같은 수 중에서 가장 큰 정수를 반환한다. 만약  $a$ 값이 0.5이면 움직임 보상은  $\frac{1}{2}$  화소 정밀도를 가지고 수행하여야 하고 0.25나 0.75일 경우에는  $\frac{1}{4}$  화소 정밀도를 가지고서 움직임 보상을 하여야 한다. 그림 4는  $\frac{1}{2}$  화소 정밀도를 가지는 경우와  $\frac{1}{4}$  화소 정밀도를 가지는 경우에 대해서 설명하고 있다.

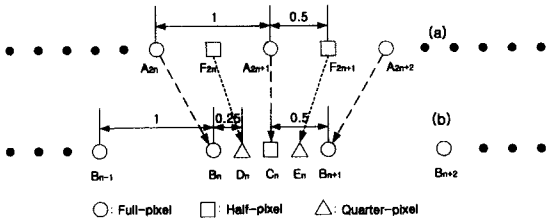


그림 4. (a) 본래 해상도 영상에서의 화소  
(b)  $\frac{1}{2}$ 로 해상도를 줄인 영상에서의 화소

Fig. 4. (a) Pixels in a full-resolution image  
(b) Pixels in a two-fold down-sampled image

화소  $C_n$ 은 <그림 4(a)>에서는 정수 단위의 화소이지만 <그림 4(b)>에서는  $\frac{1}{2}$  화소 단위의 화소이다. 즉, <그림 4(a)>에서는 정수 단위의 화소로서 움직임 보상에 있어 보간이 필요 없지만 해상도가 줄어든 <그림 4(b)>에서는 움직임 보상에 보간을 필요로 하는 화소로 변경되었다. 해상도가 반으로 줄어든 영상에서  $C_n$ 을 구하는 간단한 방법은 화소  $B_n$ 과  $B_{n-1}$ 을 선형 보간하는 것이다. 하지만  $C_n$ 을 구하기 위해서 선형 보간을 이용하게 되면 심각한 흐려짐 현상을 일으키게 된다. 움직임 보상을 통해서 재구성된 영상은 다음 영상에서 참조 영상으로 사용되기 때문에 흐려짐이 생긴 영상이 움직임 보상으로 다음 영상에 영향을 미치게 되고 다음 영상에서도 화소 보간 시에 흐려짐 현상을 일으키기 때문에 심각한 화질 열화가 나타난다.

영상이 부 표본화 될 때, 앨리어싱(Aliasing)을 막기 위해서 저주파 통과 필터를 사용 하게 된다. 앨리어싱을 막기 위한 가장 이상적인 저주파 통과 필터가 싱크(Sinc) 함수이다. 따라서 크기가 반으로 줄어든 영상에서 보간 시에 생기는 흐려짐을 막기 위해서는 선형적

인 보간을 사용하는 대신에 싱크 함수를 사용하는 보간을 이용해야 한다. 즉,  $C_n$ 을 구하기 위해서 식 (4)를 사용하는 대신 식 (5)를 사용하여 구한다.

$$C_n = 0.5B_n + 0.5B_{n+1} \tag{4}$$

$$C_n = \dots - 0.21B_{n-1} + 0.64B_n + 0.64B_{n+1} - 0.21B_{n+2} \dots \tag{5}$$

$\frac{1}{4}$  단위의 화소인  $D_n$ 의 경우는 싱크 함수를 사용하는 보간이 아닌 식 (4)와 같은 선형 보간을 수행한다. 본래의 해상도를 갖는 경우인 <그림 4(a)>를 살펴보자. 그림 4(b)에서의 화소  $D_n$ 은 <그림 4(a)>에서는  $\frac{1}{2}$  단위의 화소인  $F_{2n}$ 에 해당한다. 이 화소는 본래 해상도 영상에서  $\frac{1}{2}$  화소의 움직임 벡터가 존재 할 때 화소  $A_{2n}$ 과  $A_{2n+1}$ 을 이용하여 선형 보간에 의해 구해지는 화소이다. 그러므로 해상도가 반으로 줄어든 <그림 4(b)>에서의 화소  $D_n$ 은 본래의 해상도 영상에서의 화소  $A_{2n}$ 과  $A_{2n+1}$ 에 해당하는  $B_n$ 과  $C_n$ 을 선형 보간하여 구할 수 있다.

$$D_n = 0.5B_n + 0.5C_n \tag{6}$$

DCT 영역에서  $\frac{1}{4}$  화소 정밀도의 움직임 보상을 수행하기 위해서는 움직임 보상 시의 이동 행렬인  $L_n$ 과  $R_n$ 은 적절하게 수정해 주어야 한다. 행렬  $L_n$ 과  $R_n$ 의 행렬 원소들도 식 (4)에서와 같이 선형 보간 시에 사용된 0.5의 값이 식 (5)에서와 같은 싱크 함수값으로 변경되어야 한다. 영상 표준에서는 대부분의 연산을  $8 \times 8$  블록 단위로 처리한다. 그렇기 때문에 싱크 함수를 이용해서 보간을 해야 할 경우에 보간을 위해 무한한 화소를 사용 할 수가 없고  $8 \times 8$  블록 내에 있는 화소로 제한이 된다. 이동 행렬의 원소들 중에서 경계에 있는 원소들은 본래의 값이 아닌 경계 조건을 만족시키는 값으로 변경한다. <그림 4(a)>처럼 화소  $C_n$ 을 찾기 위해서 주변의 화소들을 이용하여 보간하는 경우  $C_n$ 을 중심으로 오른쪽에 있는 화소들을 보간 할 때의 계수들의 합을  $a$ 라고 하면  $C_n$ 을 중심으로 왼쪽에 있는 화소들을 보간 할 때의 계수들의 합은  $(1-a)$ 가 된다. 마찬가지로 수직 방향에 대해서도 위 방향의 계수들의 합이  $b$ 라고 하면 아래 방향으로의 계수들의 합은

$(1-b)$ 가 된다.

예를 들어, 한 화소가 해상도가 반으로 줄어든 영상에서 가로 방향으로  $\frac{1}{2}$  화소 단위의 움직임 보상에 의해 싱크 함수를 이용한 보간을 해야 한다고 가정하자. 이 때,  $a$ 값은 0.5가 되고 그 화소의 좌측과 우측의 계수값의 합 모두 0.5가 되어야 한다. 식 (5)에서  $C_n$ 의 값을 구하기 위해서  $C_n$ 을 중심으로 좌측으로  $B_{n-2}$ 부터  $B_n$ 까지 세 화소만 이용할 수 있다고 하면 화소  $B_{n-2}$ 에 곱해지는 계수값은  $0.07(=0.5-(0.64-0.21))$ 이 되어야 한다.

식 (2)에 의해서 구현되는 DCT 영역에서의 움직임 보상은 DCT 계수가 2 바이트의 정수이므로 정수 연산에 의해서 구현된다. 움직임 보상 연산의 대부분이 덧셈과 반복적인  $8 \times 8$  행렬 간의 곱셈이기 때문에 연산의 최적화를 위해 인텔의 MMX 기술을 사용하여 DCT 영역에서의 움직임 보상을 구현하였다.<sup>[9]</sup> MMX 기술은 클럭의 한 사이클 동안에 네 번의 덧셈이나 두 번의 곱셈을 처리 할 수 있기 때문에 이를 활용하여 움직임 보상 부분의 연산을 최적화 시켜 주었다.

#### IV. 대칭 중첩합을 사용한 DCT 영역에서의 해상도 변경 알고리즘<sup>[10]</sup>

DCT 영역에서의 부 표본화 방법은 일반적으로 DCT 계수들 중에서 고주파 쪽의 계수들을 버리고 저주파 쪽의 계수들만을 취해서 구할 수 있다. 하지만 이러한 방법은 기본 처리 단위가  $8 \times 8$ 인 영상 압축 영역에서는 부가적인 계산을 요구하게 된다. 본 알고리즘은  $8 \times 8$  블록 단위의 DCT 값들을 이용하여 해상도가 변경된 결과 값도  $8 \times 8$  블록 단위로 나타낼 수 있는 방법이다.

Martucci에 의해 유형-2 DCT가 한 영역에서의 곱셈이 다른 영역에서의 중첩합과 같다는 곱셈-중첩합의 성질을 만족한다는 것이 밝혀졌다.<sup>[11]</sup> 공간 영역에서의 영상 신호의 곱을 DCT한 값이 DCT 영역에서의 중첩합과 같다는 식을 다음과 같이 나타낼 수 있다.

$$x(n) \times w(n) = C_{2e}^{-1} C_{2e} x(n) \otimes_s C_{2e} w(n) \quad (7)$$

여기서  $\otimes$ 는 대칭 중첩합을 나타내고  $x(n)$ 과  $w(n)$ 은

공간 영역에서의 시퀀스(Sequence)이다. 위의 대칭 중첩합을 만족시키는 방법은 각 시퀀스를 반 대칭 확장을 한 후 비대칭 회전(Skew circular) 중첩합을 하는 것이다.

그러면 8점 시퀀스  $x_1(n)_8$ 과  $x_2(n)_8$ 의 DCT 값을 이용하여 16점을  $\frac{1}{2}$ 로 줄인 8점 시퀀스  $Z(k)_8$ 을 구하는 방법을 살펴보자.

$x_1(n)_8$ 의 대칭 확장된 신호를  $\tilde{x}_1(n)_{16}$ 이라 하고  $x_2(n)_8$ 의 대칭 확장된 신호를 8점 이동한 신호가  $\tilde{x}_2(n)_{16}$  이고  $\tilde{x}_1(n)_{16}$ 과  $\tilde{x}_2(n)_{16}$ 의 16점 DCT 된 결과를  $\bar{X}_1(n)_{16}$ 과  $\bar{X}_2(n)_{16}$ 이라 하자.

$$\tilde{x}_1(n)_{16} = \begin{cases} x_1(n)_8, & 0 \leq n < 8 \\ x_1(15-n)_8, & 8 \leq n < 16 \end{cases} \quad (8)$$

$$\tilde{x}_2(n)_{16} = \begin{cases} x_2(7-n)_8, & 0 \leq n < 8 \\ x_2(n-8)_8, & 8 \leq n < 16 \end{cases} \quad (9)$$

$$\bar{X}_1(k)_{16} = \begin{cases} X_1(k/2)_8, & k = \text{even}, 0 \leq k < 16 \\ 0, & k = \text{odd}, 0 \leq k < 16 \end{cases} \quad (10)$$

$$\bar{X}_2(k)_{16} = \begin{cases} X_2(k/2)_8 \times (-1)^{k/2}, & \\ k = \text{even}, 0 \leq k < 16 \\ k = \text{odd}, 0 \leq k < 16 \end{cases} \quad (11)$$

그리고 윈도우(Windowing) 신호  $w_1(n)_{16}$ 과  $w_2(n)_{16}$ 의 16점 DCT를  $W_1(k)_{16}$ 과  $W_2(k)_{16}$ 이라 하자.

$$w_1(n)_{16} = \begin{cases} 1, & 0 \leq n < 8 \\ 0, & 8 \leq n < 16 \end{cases} \quad (12)$$

$$w_2(n)_{16} = \begin{cases} 0, & 0 \leq n < 8 \\ 1, & 8 \leq n < 16 \end{cases} \quad (13)$$

공간 영역에서 윈도우를 통한 곱셈 연산은 식 (7)에서 나타낸 것처럼 DCT 영역에서는 대칭 중첩합으로 구할 수 있다. 이러한 관계를  $\bar{X}_1(n)_{16}$ 과  $\bar{X}_2(n)_{16}$ 을 사용하여 나타내면 다음과 같다.

$$X(k)_{16} = \bar{X}_1(k)_{16} \otimes_s W_1(k)_{16} + \bar{X}_2(k)_{16} \otimes_s W_2(k)_{16} \quad (14)$$

구한  $X(k)_{16}$ 을 가지고서  $\frac{1}{2}$ 로 부 표본화 하는 방법은 식 (15)같이  $X(k)_{16}$  값 중에서 저주파의 8 점만 선택하는 것이다.

$$Z(k)_8 = X(k)_{16}, \quad 0 \leq k < 8 \quad (15)$$

V. 실험 및 결과

본 논문에서 제안한 알고리즘을 평가하기 위해 여섯 종류의 CCIR-601 비디오 시퀀스를 사용했다. 영상의 크기는 720×480이고 각 시퀀스 당 100장의 영상을 사용했다. 사용한 시퀀스는 "Cheerleaders(Cheer)", "Car", "Flower Garden (FG)", "Football(Foot)", "Mobile and Calendar (Mob)", "Susie(Sus)" 이다. 테스트 영상으로 MPEG-2 인코딩 시에는 목표 비트율을 4Mbps로 하였고 모든 프레임은 프레임 영상으로 하였다. 그리고 I 영상은 매 10 프레임마다 삽입을 하였고 B 영상은 사용하지 않았다.

<표 1>에서는 각 100 프레임씩의 시퀀스를 세 가지의 보간 방법을 사용했을 때의 평균 PSNR을 나타낸다. FULL이 의미하는 것은 본래의 해상도에서 움직임 보상을 먼저 수행한 후 완전히 복원된 영상에 대해서 대칭 중첩합을 사용한 해상도 변경 알고리즘을 사용하였을 때이다. SINC가 의미하는 것은 먼저 제안한 해상도 변경 알고리즘을 사용하여 해상도를 줄인 후 8×8로 제한된 싱크 함수를 이용하여 움직임 보상을 했을 때의 결과이다. 마지막으로 LINEAR가 나타내는 것은 SINC와 같이 먼저 해상도를 변경하는 방법은 동일하나 해상도를 변경한 후에 움직임 보상을 행할 시에 보통의 선형 보간을 사용했을 때의 결과이다. 1/2 화소 단위일 때는 0.5를 사용하여 보간을 하고 1/4 화소 단위일 때는 0.75와 0.25를 사용하여 보간을 하였다. 표 1에서도 알 수 있듯이 SINC 방법이 FULL 보다는 PSNR이 약간 떨어지지만 그냥 LINEAR 방법을 쓰는 것 보다는 PSNR이 좋음을 알 수 있다.

<표 2>에서는 Mob 시퀀스를 사용하여 복호를 했을 때, 각 주요 부분의 계산 시간을 측정된 결과를 보여준다. 실험 환경으로는 CPU는 Intel Pentium III 1GHz이고 메모리는 512Mbytes 그리고 OS는 Windows 98이다. SINC 방법이 FULL 방법보다 줄어든 해상도에서 움직임 보상을 수행하기 때문에 움직임 보상 계산 시간 측면에서 2.63배 빠르다는 것을 보여준다. SINC 방법과 LINEAR 방법의 시간이 동일한 것은 DCT 영역에서는 보간 방법에 상관없이 식 (2)에 의해 움직임 보상을 수행하기 때문이다.

<그림 4>는 Mob 시퀀스에 대한 각 보간 방법의

PSNR 비교 그래프이다. LINEAR 방법이 SINC 방법보다 PSNR이 좋지 않고 급격히 떨어지는 것을 알 수 있다.

<그림 5>는 Mob 시퀀스를 이용하여 각 보간 방법으로 보간을 했을 때의 주관적인 화질 비교를 위한 예이다. 영상의 크기를 반으로 줄였을 때, 시퀀스 중에서 9번째 P 영상이다. FULL과 SINC는 주관적 화질 비교에서 그리 차이가 나지 않으나 LINEAR 방법은 흐려짐이 심하게 생기는 것을 볼 수 있다.

표 1. 테스트 시퀀스의 평균 PSNR  
Table 1. Average PSNR of test sequences.

[단위 : dB]

	Cheer	Car	FG	Foot	Mob	Sus
FULL	31.3	34.4	31.8	34.0	28.9	43.6
SINC	30.8	33.5	30.9	32.6	27.8	41.5
LINEAR	30.3	32.1	29.2	31.2	25.3	40.2

표 2. 해상도를 줄인 Mob 시퀀스의 복호 시간

Table 2. Computation time for decoding with down-scaling of Mob sequence (100 frames)  
[단위 : 시간]

	VLD+IQ	Resizing	MC	Total
FULL	1.76	0.80	3.81	6.37
SINC	1.76	0.80	1.45	4.01
LINEAR	1.76	0.80	1.45	4.01

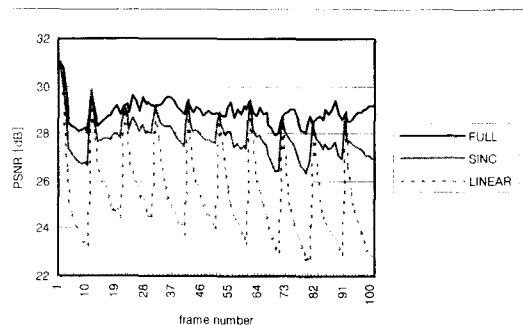
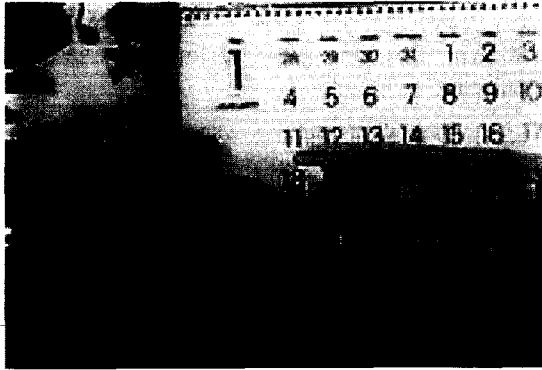
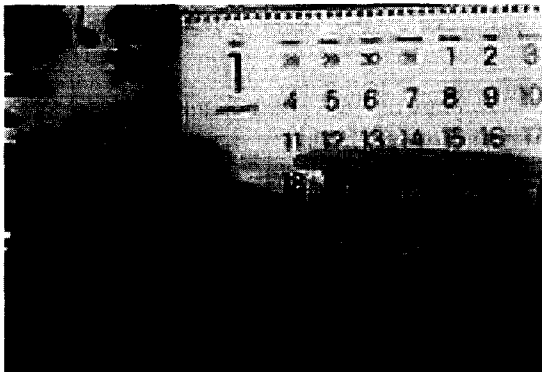


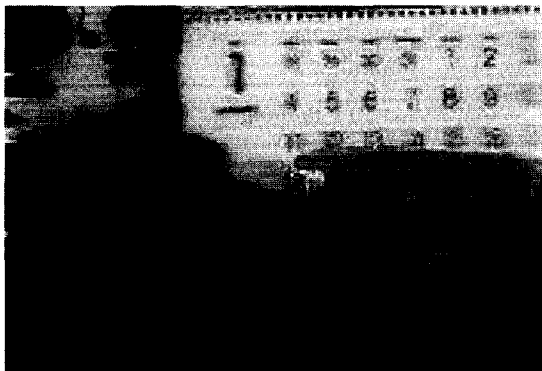
그림 4. Mob 시퀀스의 PSNR 곡선  
Fig. 4. PSNR curve of Mob sequence.



(a) FULL



(b) SINC



(c) LINEAR

그림 5. Mob 시퀀스의 9번째 P 프레임의 각 보간 방법의 영상 비교

Fig. 5. Image comparison for the 9th P-frame of "Mobile and Calendar sequence"

## VI. 결 론

MPEG-2 복호기를 사용하여 MPEG-2 영상을 복호한 후 영상의 해상도를 효율적으로 줄일 수 있는 방법을 제안하였다. 해상도를 줄인 후 일반적인 선형 보간 방법으로 움직임 보상을 수행하면 심한 흐려짐이 생기

게 된다. 이를 해결하기 위해 싱크 함수를 사용하는 보간 방법을 제안하였다. 제안한 방법은 해상도 변경을 수행 한 후에 움직임 보상을 수행하기 때문에 메모리 공간을 작게 차지 할 뿐만 아니라 계산시간 적인 측면에서도 장점을 가지고 해상도를 변경 할 수 있는 고속의 압축변환기를 구성하는데도 많은 도움을 줄 수 있을 것이다. 실험 결과에서도 제안한 방법이 계산 시간이나 화질 면에서도 효율적임을 보여주었다.

## 참 고 문 헌

- [1] A. Vetro and H. Sun, "On the motion compensation within a down conversion decoder," *Journal of Electronic Imaging*, Vol. 7, No. 3, pp. 616~627, Jul. 1998.
- [2] B. C. Song, M. J. Kim, S. K. Jang, and J. B. Ra, "Modified IDCT kernel for down-conversion of MPEG-2 compressed video," *IEEE Transactions on Consumer Electronics*, Vol. 45, No. 2, pp. 340~344, May 1999.
- [3] S. -F. Chang and D. G. Messerschmitt, "A new approach to decoding and compositing motion-compensated DCT based images," in *Proc. ICASSP '93*, Minneapolis, MN, pp. 421~424, Apr. 1993.
- [4] N. Merhav and V. Bhaskaran, "Fast algorithm for DCT-domain image down-sampling and for inverse motion compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 3, pp. 468~476, Jun. 1997.
- [5] W. Zhu, K. Yang, and M. Beackem, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," *Bell Labs technical journal*, Vol. 3, No. 3, pp. 21~29, Jul.-Sep. 1998.
- [6] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," *IEEE Transactions on Multimedia*, Vol. 2, No. 2, pp. 101~110, Jun. 2000.
- [7] J. Song and B. Yeo, "A fast algorithm for DCT-domain inverse motion compensation

based on shared information in a macro block," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 10, pp 767~775, Aug. 2000.

[8] P. Assuncao and M. Ghanbari, "A frequency domain video transcoder for dynamic bit-rate reduction of MPEG-2 bitstreams," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 8, pp. 953~967, Dec. 1998.

[9] D. Bistry, et. Al., The complete guide to MMX

technology, Intel Corporation, McGraw-Hill, Inc., 1997.

[10] H. W. Park, Y. S. Park, and S. K. Oh, "L/M-fold image resizing in block-DCT domain using symmetric convolution," submitted to *IEEE Transactions on Image Processing*, Sep. 2001.

[11] S. A. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Transactions on Signal Processing*, Vol. 42, No. 5, pp. 1038~1051, Dec. 1998.

---

저 자 소 개

---



吳承均(學生會員)

1998년 한국과학기술원 전기 및 전자공학과 졸업(공학사). 2000년 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 2000년~현재 한국과학기술원 전자전산학과 전기 및 전자공학전공 박사과정 재학 중. <주

관심분야 : 동영상 코덱, 트랜스코딩, 고속영상처리 등>



朴玄旭(正會員)

1981년 서울대학교 전기공학과 졸업(공학사). 1983년 한국과학기술원 전기 및 전자공학과 졸업(공학석사). 1988년 한국과학기술원 전기 및 전자공학과 졸업(공학박사). 1989년~1992년 University of Washington

연구원. 1992년~1993년 삼성전자 수석연구원. 1993년~현재 한국과학기술원 전자전산학과 전기 및 전자공학전공 교수. <주관심분야 : 영상처리, 영상압축, 의료영상 시스템, 멀티미디어 시스템>