

論文2001-38SP-2-3

DV에서 MPEG-2로의 주파수 영역 변환 부호화 (A Frequency Domain DV-to-MPEG-2 Transcoding)

金度年*, 尹汎植**, 崔潤植***

(Donyeon Kim, Bumsik Yoon, and Yoonsik Choe)

요약

디지털 캠코더에서 이용하는 영상 압축 방식인 DV 부호화 방식은 DCT와 가변장 부호화 방식을 이용한 다. DV 방식은 하드웨어 복잡도가 낮은 반면 압축된 비트율이 약 26Mb/s로 높은편이다. 따라서 스튜디오에서 낮은 복잡도로 영상을 부호화 한 후 VOD 시스템에 사용하기 위하여 MPEG-2로 변환부호화 할 필요가 있다. 이때의 두 압축방식이 DCT를 이용하므로, DCT 영역에서 변환부호화 하면 중간 과정을 줄일 수 있어서 계산상의 복잡도를 줄일 수 있다. DV 방식에서 MPEG-2 인트라로 변환부호화 시에, DV 방식의 4:1:1 색차 포맷을 MPEG-2의 4:2:2 영상 포맷으로 변환할 때 와 2-4-8 DCT 모드에서 8-8 DCT 모드로 변환 시 변환 영역에 있는 데이터에 미리 계산된 행렬을 곱함으로써 병렬처리가 가능하게 하였다. MPEG-2 유통 시에 서브 블록의 분산을 완전히 DCT 영역에서 계산하였다. 실험을 통하여 제안한 방식들을 검증하였다. MPEG-2 인터 프레임 부호화로 변환 부호화 할 때 DCT 계수를 이용하여 계층적으로 움직임을 추정하였다. 먼저 4개의 서브 블록에 있는 4개의 DC 값으로 하나의 매크로 블록에 대한 움직임을 추정한 다음 각 서브 블록의 저주파수에 해당하는 2×2 에 IDCT를 취하여 16 포인트로 구성된 매크로 블록을 만든 후 이에 대한 움직임을 추정하며, 다섯 번째 단계에서 서브 화소에 대한 움직임을 추정함으로써 움직임 추정을 마친다. 탐색영역을 겹치는 방식이 겹치지 않는 방식보다 좋은 PSNR값을 보여 주었다.

Abstract

Digital Video (DV) coding standards for digital video cassette recorder are based mainly on DCT and variable length coding. DV has low hardware complexity but high compressed bit rate of about 26 Mb/s. Thus, it is necessary to encode video with low complex video coding at the studios and then transcode compressed video into MPEG-2 for video-on-demand system. Because these coding methods exploit DCT, transcoding in the DCT domain can reduce computational complexity by excluding duplicated procedures. In transcoding DV into MPEG-2 intra coding, multiplying matrix by transformed data is used for 4:1:1-to-4:2:2 chroma format conversion and the conversion from 2-4-8 to 8-8 DCT mode, and therefore enables parallel processing. Variance of sub block for MPEG-2 rate control is computed completely in the DCT domain. These are verified through experiments. We estimate motion hierarchically using DCT coefficients for transcoding into MPEG-2 inter coding. First, we estimate motion of a macro block (MB) only with 4 DC values of 4 sub blocks and then estimate motion with 16-point MB using IDCT of 2×2 low frequencies in each sub block, and finish estimation at a sub pixel as the fifth step. ME with overlapped search range shows better PSNR performance than ME without overlapping.

MPEG-2, 변환부호화(Transcoding), DCT, 움직임추정(Motion Estimation), 유통제어(Rate Control)

* 學生會員

** 正會員, 三星電子 中央研究所

*** 正會員, 延世大學校 大學院 電氣電子工學科

(Corporate R&D Center, Samsung Electronics)

(Dept. of Electrical and Electronic Eng., The Graduate School of Yonsei Univ.)

接受日字:1999年11月18日, 수정완료일:2001年1月19日

I. 서 론

변환부호화는 스튜디오 신호를 높은 비트율로 압축 및 저장 후 낮은 비트율로 변환 및 분배 시에 필요하다. 영상 편집, 디지털 방송, 주문형 비디오 서비스는 영상 데이터의 압축방식으로 MPEG-2를 많이 사용하고 있다. 그 이유는 MPEG-2(Moving Pictures Experts Group-2) 복호화 보드가 널리 보급되고 있고, 압축된 비디오 데이터의 비트율이 1.5Mb/s에서 6Mb/s 사이이므로 비디오를 전송하거나 비디오를 매체에 저장하기에 유리하기 때문이다. 디지털 캠코더에 이용되고 있는 DV(Digital Video) 압축방식^[1]은 비트율이 28Mb/s로 높아 전송하거나 매체에 저장하기에 데이터량이 많다. 그러므로 디지털 캠코더로 촬영함으로써 DV 방식으로 압축된 동영상을 MPEG-2^[2]로 변환한 다음 비디오 서버에 저장할 필요가 있다.

본 논문은 DV에서 MPEG-2로 변환 코딩할 때 DV 데이터를 복호화 한 후 다시 MPEG-2로 부호화하는 과정을 줄여서 소프트웨어로 변환할 경우 고속 워크스테이션을 이용하여 1분 영상을 변환하는 데 3시간에서 5시간 걸리는 것을 줄이는 방법을 찾는 것에 관한 것이다. 또한 하드웨어로 구현 시에 계산량 및 복잡도를 줄이는 방법을 모색한다.

변환부호화를 위하여 여러 가지 방법이 있을 수 있으나, 두 번째 부호화기에서 비트율을 줄이고자 할 때 [3]에서는 고주파 DCT(Discrete Cosine Transform) 계수를 없애고, 저주파 DCT 계수는 그대로 보내는 방법을 제안하였다. 이 때의 재양자화는 DCT 기반 저역 통과 필터링이라고 할 수 있다. 이 때 블록 잡음이 보일 수 있다. 이 같은 방법들은 두 부호화기 사이의 비트율 변화가 아주 작을 때 적절한 방법이다.

MPEG-2 인트라 프레임 변환 부호화 할 때 재양자화를 함으로써 “복호화 하고 부호화”하는 방법에 비하여 처리 과정을 줄이고 계산량을 줄일 수 있다. I 프레임에 대한 MPEG-2 호환 변환부호화에 대해서 지금까지의 방법은 두 번째 양자화기 Q_2 의 양자화 구간을 첫 번째 양자화기 Q_1 의 양자화 구간보다 넓게 하는 방법을 이용하였다. 이 경우는 첫 번째 양자화기와 두 번째 양자화기의 비율이 맞을 때만 적용 가능하다. 그렇지 않을 경우 재양자화 시 다음 두 가지 경우가 발생한다.

첫 번째, Q_1 의 하나의 양자화 구간을 맵핑하였을 경우 Q_2 의 두 구간에 걸치는 경우이다. 두 번째, Q_1 의 하나의 양자화 구간을 맵핑하였을 경우 Q_2 의 한 구간 안으로 들어가는 경우이다. 여기서 첫 번째 경우에 대하여 최적화 할 여지가 있다. 이를 위하여 변환 부호화기의 양자화 구간을 조정하기 위한 두 가지 방법이 있을 수 있다. 먼저, 양자화 구간을 조정하기 위한 지침으로 MSE(Mean Squared Error) 가격 함수를 최소화하는 방법이다. 두 번째 방법은 MAP(Maximum A Posteriori) 가격 함수를 이용하여 첫 번째 방법보다 rate-distortion 성능을 향상시킨 것이다^[4]. MSE 양자화기는 높은 PSNR을 보이며, TM5의 양자화기 보다 1.5dB 이득이 있다. 그러나, MSE 양자화기는 비트율을 증가시키므로 중요한 영상이 있는 블록들에 국부적으로 사용된다^[4].

MPEG-2에서 비트율을 줄여 변환 부호화 시에 [5]에서는 움직임 벡터(MV)를 그대로 이용하고, 주파수 영역에서 움직임 보상(MC: Motion Compensation)을 함으로써 일반적으로 쓰는, 화소 기반 방법보다 계산량을 81% 줄였다고 보고하고 있다.

본 논문의 III장에서 DV 방식에서 MPEG-2 인트라 부호화 방식으로 변환하는 것에 대하여 서술한다. 변환 영역에서 색차 포맷 변환과 DCT 모드 변환을 함으로써 변환기의 복잡도를 줄인다. 또한 변환영역에서 율제어 방식을 제안한다. IV장에서는 DV 방식에서 MPEG-2 인트라 부호화 방식으로 변환하는 것에 대하여 서술한다. DCT 계수를 이용하여 단계적 움직임 추정을 함으로서 탐색 속도를 개선하는 방법에 대하여 서술한다.

II. DV 압축 방식

DV 방식에서 영상 프레임별 편집을 위하여 1개의 영상 프레임을 테이프의 10개 트랙에 저장한다. DV 방식 부호화기의 입력영상은 ITU-R 601 표준의 4:2:2 포맷이다. 압축시 영상의 색차신호 CR, CB를 가로로만 각각 2:1 다운 샘플링하여 4:1:1 영상포맷으로 변환한다. 즉, 그림 2 (b)의 4:2:2 신호에서 휘도 신호에 대하여는 변화가 없으나, 색차신호는 가로로 2 화소에서 한 화소만을 남겨두어 그림 2 (c)와 같은 4:1:1 신호로 만든다. 각 화소는 A/D 변환기에서 0~255 사이 크기로

출력되지만, 이 값에서 128을 빼어서 -128~127 사이의 크기로 바꾼다. 영상신호를 압축하기 위하여 DCT, Run-Length 코딩, 가변장 부호화 과정을 수행한다.

크기가 8×8 블록에 대하여 2차원 DCT를 수행한다. 이 방식을 다음에 설명할 방식과 구분하기 위하여 8-8 DCT라고 한다. 두 필드 사이 차이가 클 경우, 즉 물체의 움직임이 클 경우 두 필드 차이를 4×8 블록 크기로 DCT하며, 이것을 2-4-8 DCT라고 한다.

양자화 스텝을 결정하기 위하여 클래스 번호, DCT 블록 내 지역에 따라 구분되는 지역 번호, 0~15 사이의 값을 가지는 양자화 번호를 이용한다. 압축된 영상신호를 고정된 크기의 비디오 세그먼트에 최적으로 채우는 양자화 번호를 선택한다.

MB(Macro Block)은 4개의 휘도 신호 블록과 각 1개의 Cr, Cb블록으로 구성된다. SB(Super Block)은 27개의 MB으로 구성되며, 화면 가로로 5개, 화면 세로로 10개 배열된다. 한 프레임에서 가로로 5개 SB를 테이프의 1개의 트랙에 저장한다.

상기의 블록에 대한 계층구조와 별도로 세그먼트 블록 구조가 정의되어있다. 1개의 비디오 세그먼트는 5개의 MB으로 구성된다. 비디오 세그먼트의 목적은 크기가 가변적인 압축 데이터를 일정한 속도로 부호화기에서 출력시키기 위하여 가변적인 압축 데이터를 고정된 크기 공간에 최대한 저장하기 위한 것이다. 즉, 한 개의 영상 프레임 내에서 서로 가장 멀리 떨어져있는 5개의

MB을 선택함으로써 세그먼트별 영상 복잡도가 비슷하게 한 후, 5개의 MB을 압축한 데이터를 한 개의 고정된 크기를 갖는 세그먼트에 채워넣는다. MPEG-2에서는 이 기능에 대응되는 것으로 가상 버퍼가 있어서 데이터량이 일시적으로 변동하는 것을 완충한 후 데이터를 일정한 속도로 출력시킨다. MPEG-2 와 DV의 특성 비교는 표 1과 같다.

III. MPEG-2 인트라 부호화

DV 방식으로 압축된 영상신호를 MPEG-2의 인트라 픽처로 변환한다. 이때 DCT 영역 상에서 변환함으로써 두 동영상 압축 방식에서 중복된 과정을 생략하여 구현 시 하드웨어 복잡도를 줄인다.

1. 블록 재구성

두 코덱의 색차신호 포맷이 서로 다르며, 따라서 MB 구성도 그림 1과 같이 서로 다르므로 두 코덱 사이에 MB 데이터를 변환 시켜야 한다. 색차신호 성분을 1:2업 표본화 한 후 재구성한 MB 블록은 그림 1(b)와 같다.

표 1. 압축방식 비교

Table. 1. Compression method comparison.

	DV	MPEG-2
입출력 신호 포맷	ITU-R 601 4:2:2	ITU-R 601 4:2:2/4:2:0
내부 신호 포맷	4:1:1	4:2:2/4:2:0
픽처 타입	프레임 전용	필드 혹은 프레임
DCT 모드	2-4-8 모드 8-8 모드	8-8 모드
움직임 추정/보상	×	○
DCT/Run-Length	○	○
가변장부호화	○	○
버퍼링 방법	세그먼트 블록 정의	율 제어

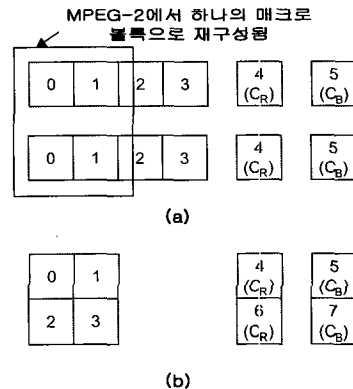


그림 1. 두 압축 방식에서 MB 내의 DCT 블록의 배열
(a) 4:1:1 색차신호 포맷을 이용하는 두 개의 DV MB
(b) 4:2:2 색차신호 포맷을 이용하는 한 개의 MPEG-2 MB

Fig. 1. DCT blocks in the MBs of two compression standards.

(a) DV MB using 4:1:1 chroma format
(b) MPEG-2 MB using 4:2:2 chroma format

2. 4:1:1 색차신호 포맷에서 4:2:2 색차신호 포맷으로 변환

DV 및 MPEG-2에서 쓰는 색차신호 포맷의 종류는

그림 2와 같다. MPEG-2에서 DV 방식에서 쓰는 색차 신호 포맷인 4:1:1을 지원하지 않으므로 4:2:2 혹은 4:2:0로 변환하여야 한다. 본 논문에서는 4:2:2로 변환한다. 색차 성분인 CR, CB로 구성된 서브 블록들 각각에 대하여 공간영역에서 가로방향으로 1:2 업 표본화 하여야 한다. 업 샘플링 방법으로 쌍선형(Bi-linear), 큐빅(Cubic), 스플라인(Spline) 등의 방법^[6]이 있으나, 본 연구에서는 구현이 간단한 쌍선형방식을 선택한다.

이렇게 하는 기본적인 방법은 고속 IDCT 및 DCT 알고리즘을 이용하는 것이다. 즉, 4:1:1 포맷의 DCT 데이터를 역변환 한 후 업 표본화 처리하고, 다시 DCT 처리하는 것이다. 지금까지 알려진 가장 우수한 고속 DCT 알고리즘은 [7]의 방법이다. 이 방식과 달리 본 논문에서는 DCT 영역에서 포맷 변환하기 위하여 필요한 연산을 미리하여 두는 방법을 제안한다.

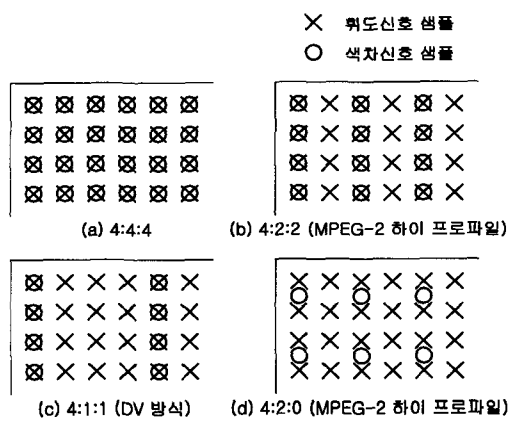


그림 2. 색차 신호 포맷의 종류
Fig. 2. Chroma signal formats.

일차원 신호에 대한 DCT 영역에서 1:2 업 표본화는 식(1)과 같이 계산할 수 있다. 시간영역에서 업 표본화는 입력 데이터로부터 만든 (8×1) 크기의 칼럼 벡터에 (2)와 같은 (16×8) 크기의 행렬 $h_{16 \times 8}$ 을 곱하면 된다. 본 논문에서는 이미 DCT 처리된 입력 데이터로 만든 칼럼 벡터 $X_{8 \times 1}$ 에 대하여 식 (3)의 행렬 $H(1)$, $H(2)$ 를 곱하여, 1:2 업 표본화한 데이터에 대한 DCT 처리한 결과를 얻는다.

$$Z_{16 \times 1} = \begin{bmatrix} S_{8 \times 8} & 0_{8 \times 8} \\ 0_{8 \times 8} & S_{8 \times 8} \end{bmatrix} h_{16 \times 8} S_{8 \times 8}^{-1} X_{8 \times 1} = \begin{bmatrix} H_{8 \times 8}(1) \\ H_{8 \times 8}(2) \end{bmatrix} X_{8 \times 1} \quad (1)$$

(141)

$$h_{16 \times 8} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1/2 & 1/2 & & & \\ 0 & 1 & & & \\ 0 & 1/2 & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 0 \\ \vdots & \vdots & \vdots & 1/2 & 1/2 \\ \vdots & \vdots & \vdots & 0 & 1 \\ 0 & \dots & \dots & 0 & 1 \end{bmatrix} \quad (2)$$

여기서 $X_{8 \times 1}$ 은 DCT 변환된 4:1:1 데이터 이고, $S_{8 \times 8}$ 은 DCT 계수 행렬이고, $S_{8 \times 8}^{-1}$ 은 IDCT 계수 행렬 이고, $Z_{16 \times 1}$ 은 DCT 변환된 4:2:2 데이터이다.

정확히 계산한 행렬 $H_{8 \times 8}(1)$, $H_{8 \times 8}(2)$ 는 식 (3)과 같다. 이 방법은 병렬처리가 가능하여 한 스테이지만에 연산이 끝나는 장점이 있다.

$$H_{8 \times 8}(1) = \begin{bmatrix} 1 & 0.8022 & -0.1633 & -0.3426 & 0 & 0.09 & -0.0676 & -0.1108 \\ 0 & 0.4584 & 0.9435 & 0.4843 & -0.2452 & -0.317 & 0 & 0.1136 \\ 0 & -0.0721 & 0.0884 & 0.6238 & 0.7886 & 0.2779 & -0.2134 & -0.2296 \\ 0 & 0.0382 & 0 & -0.0649 & 0.2079 & 0.6144 & 0.5748 & 0.2158 \\ 0 & -0.0138 & 0.028 & 0.0819 & 0 & 0.0365 & 0.3943 & 0.5225 \\ 0 & 0.0128 & 0 & 0.0109 & 0.1389 & -0.1381 & -0.1715 & -0.3368 \\ 0 & -0.0013 & 0.0366 & 0.062 & -0.056 & -0.1788 & -0.0684 & 0.0485 \\ 0 & 0.0101 & -0.0074 & -0.058 & -0.0488 & 0.0057 & 0 & -0.0316 \end{bmatrix} \quad (3)$$

$$H_{8 \times 8}(2) = \begin{bmatrix} 1 & -0.9756 & 0.1633 & 0.1956 & 0 & -0.1882 & 0.0676 & 0.0763 \\ 0 & 0.3739 & -0.9435 & 0.8722 & -0.2452 & -0.1442 & 0 & 0.17 \\ 0 & 0.0907 & -0.0884 & -0.314 & 0.7886 & -0.7416 & 0.2134 & 0.1358 \\ 0 & 0.0287 & 0 & -0.1628 & 0.2079 & 0.1222 & -0.5748 & 0.5757 \\ 0 & 0.0207 & -0.028 & -0.0163 & 0 & 0.1835 & -0.3943 & 0.3491 \\ 0 & 0.0064 & 0 & -0.0546 & 0.1389 & -0.1908 & 0.1715 & -0.0963 \\ 0 & 0.009 & -0.0366 & 0.0663 & -0.056 & -0.0132 & 0.0884 & -0.0874 \\ 0 & -0.0067 & 0.0074 & 0.0192 & -0.0488 & 0.0401 & 0 & -0.0204 \end{bmatrix}$$

식 (2)의 행렬 $h_{16 \times 8}$ 의 16번째 행은 근사값을 구한다. 네 번째 인터폴레이션한 값을 정확히 구하기 위하여 두 서브 블록에 대한 입력 데이터가 필요하다. 이 경우 제안한 방식을 이용할 경우 계산량이 많이지므로, 행렬 $H_{8 \times 8}(1)$, $H_{8 \times 8}(2)$ 를 Decompose하여 Sparse 행렬로 만드는 연구가 필요하다.

앞에서 색차 포맷 변화 시에 간단한 구조를 만들기 위하여 선형 업 샘플링에 대하여 서술하였다. 다음 장에서 서술할 DV 압축방식에서 MPEG-2 인터 부호화 방식에서는 이미 알려져 있는 FIR 필터를 이용하여 공간영역에서 전처리를 행하였으며, 이때의 필터 계수는 다음과 같다^[8].

- CB, CR의 포맷을 4:1:1에서 4:2:2 으로 바꾸기 위한 과정
 - ① 수평방향 단순 업 샘플링
 - ② 수평방향 필터링
 사용한 필터 계수: {-0.0234375, 0.0, 0.2734375, 0.5,

0.2734375, 0.0, -0.0234375)

3. 2-4-8 DCT에서 8-8 DCT로 변환

MPEG-2에서 2-4-8 DCT 모드를 지원하지 않으므로, 2-4-8 DCT를 8-8 DCT로 변환하여야 한다. 본 논문에서는 Butterfly 구조를 가지는 고속 DCT 알고리즘을 이용하지 않고 앞 절에서와 동일한 방법으로 식 (5)의 $P_{8 \times 8}$ 행렬을 미리 계산하여 곱하는 방식을 이용한다. 식 (5)에서 $T_{8 \times 8}$ 행렬은 2-4-8 DCT에서 8-8 DCT로 변환할 때 순서를 바로잡아 주는 순열 행렬이다. 일차원 2-4-8 DCT 연산을 식 (4)로 나타낼 수 있다. 2-4-8 DCT는 2차원 DCT이지만 행 방향으로는 모드 변환 처리가 필요 없으므로 일차원으로만 모드 변환 처리한다. 식(4)에서 $N_1=8$ 이고, $N_2=4$ 이다.

$$F(u, v) = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} C(u)C(v)\{f(x, 2y) + f(x, 2y+1)\} \cos \frac{(2x+1)u\pi}{2 N_1} \cos \frac{(2y+1)v\pi}{2 N_2}$$

$$F(u, v+4) = \sum_{x=0}^{N_1-1} \sum_{y=0}^{N_2-1} C(u)C(v)\{f(x, 2y) - f(x, 2y+1)\} \cos \frac{(2x+1)u\pi}{2 N_1} \cos \frac{(2y+1)v\pi}{2 N_2}$$

$$C(u)C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases} \quad (4)$$

$$X_{8 \times 1} = S_{8 \times 8} T_{8 \times 8} \frac{1}{2} p_{8 \times 8} \begin{bmatrix} S_{4 \times 4}^{-1} & 0_{4 \times 4} \\ 0_{4 \times 4} & S_{4 \times 4}^{-1} \end{bmatrix} \begin{bmatrix} X_{4 \times 1}(1) \\ X_{4 \times 1}(2) \end{bmatrix} = P_{8 \times 8} \begin{bmatrix} X_{4 \times 1}(1) \\ X_{4 \times 1}(2) \end{bmatrix} \quad (5)$$

$$T_{8 \times 8} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6)$$

$$p_{8 \times 8} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \quad (7)$$

$$P_{8 \times 8} = \begin{bmatrix} 0.7071 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6935 & 0 & 0 & 0 & 0 & 0.1274 & -0.0528 \\ 0 & 0 & 0.6533 & 0 & 0 & 0 & 0.2590 & 0 \\ 0 & 0 & 0 & 0 & 0.5879 & 0.1503 & 0 & 0.3629 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2706 & 0.6533 \\ 0 & 0 & 0 & -0.3928 & 0.2250 & 0 & 0.5432 & 0 \\ 0 & 0 & -0.2706 & 0 & 0 & 0.6036 & 0 & -0.2500 \\ 0 & -0.1379 & 0 & 0 & 0.6407 & 0 & -0.2654 & 0 \end{bmatrix} \quad (8)$$

식 (5)의 $P_{8 \times 8}$ 행렬은 Sparse 행렬이 되어 곱셈량이

일반 고속 DCT 알고리즘과 비슷하지만 Butterfly 구조가 아니어서 병렬처리가 가능하므로 계산속도를 개선할 수 있다. 또한 하드웨어 구현 시 $P_{8 \times 8}$ 행렬을 미리 계산하여 ROM에 저장하여 곱셈을 할 수 있으므로 고속 DCT 알고리즘보다 정확히 계산을 할 수 있다.

고속 DCT 알고리즘 중에서 Arai의 방식^[7]이 가장 계산량이 적은 것으로 알려져있다. 고속 DCT 알고리즘은 FFT(Fast Fourier Transform)을 이용하여 만드는데, Arai등은 8 포인트 DCT 경우 5개의 곱셈만으로 DCT 처리가 가능하다. DCT 영역에서 2-4-8 DCT를 8-8 DCT 모드로 변환 시에 IDCT, DCT 모드 변환, DCT 등의 3가지 처리가 필요하며, 이 경우 제안한 방식은 식 (7)에서 알 수 있듯이 Arai의 방식 보다 2배의 곱셈이 필요하다. 그러나, Arai의 방식^[7]에서는 -1을 곱하는 과정이 Butterfly 구조를 가지고 있어 병렬처리가 불가능하지만 제안한 방식은 병렬처리가 가능하므로 하드웨어 구현시 속도를 개선할 수 있다.

4. m_quant 계산

MPEG-2의 엔코더 검증용 모델에 대하여 기술한 TM 5(Test Model 5)^[9]에서는 가상버퍼의 충만도와 MB별 분산을 곱한 m_quant를 이용하여 MB별 양자화 스텝을 가변시키는 적응양자화 방법을 사용한다. MB에서 하나의 서브 블록 영상신호에 대한 분산은 식 (9)와 같이 계산하며, MB에서 휘도 성분에 대한 프레임 픽처로 구성했을 때의 4개 서브 블록과 필드 픽처로 구성했을 때의 4개 서브 블록에 대한 분산 중에서 가장 적은 값을 택하여 m_quant 계산에 이용한다. DV 방식에서 DCT된 영상신호를 IDCT(Inverse DCT)하지않고 DCT 영역에서 분산을 구하기 위하여 시간영역과 DCT 영역사이의 에너지 정리인 식 (10)을 이용한다^[10].

$$\begin{aligned} \text{var_blk} &= \frac{1}{64} \sum_{n=1}^{64} [x(n) - x]^{-2} \\ &= \frac{1}{64} \sum_{n=1}^{64} x^2(n) - (x)^{-2} \end{aligned} \quad (9)$$

$$\begin{aligned} \sum_{x=0}^N \sum_{y=0}^N |f(x, y)|^2 &= \frac{1}{4N^2} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} |C(u)C(v)F(u, v)|^2 \\ C(u)C(v) &= \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v = 0 \\ 1 & \text{otherwise} \end{cases} \end{aligned} \quad (10)$$

또한 8×8 크기 2차원 DCT 식 (11)에서 주파수가 (0,0)인 점의 값이 시간 영역 데이터의 평균이 됨을 알

수 있다. 그러므로 식 (10)과 식 (11)을 식 (9)에 대입하면 분산을 구하는 식 (12)을 얻을 수 있다.

$$F(u=0, v=0) = \frac{2}{N} \frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)$$

$$\cos \frac{(2x+1)u\pi}{2N} \cos \frac{(2y+1)v\pi}{2N} = \frac{1}{8} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \quad (11)$$

$$\text{var_sblk} = \frac{1}{64} \sum_{u=0}^7 \sum_{v=0}^7 F(u, v)^2 - (8F(0,0))^2 \quad (12)$$

식 (12)를 이용하여 시간영역에서 DCT로 압축된 Motion JPEG 혹은 DV 영상의 m_quant를 별도의 IDCT/DCT 처리를 하지않고 DCT 영역에서 계산한다.

IV. MPEG-2 인터 부호화 방식

DV 동영상 압축 방식을 MPEG-2 인터 코딩으로 변환하는 변환 부호화기를 만들기 위하여 DCT 계수를 이용한 움직임 추정 방법에 대하여 서술한다. DV 동영상 압축 방식 역시 MPEG-2와 유사한 DCT 압축 기술을 이용하므로, 이러한 변환 작업이 DCT 영역에서 이루어진다면, 많은 계산을 요하는 IDCT(Inverse DCT)와 DCT 연산을 줄일 수 있으므로, 실시간 변환부호화가 용이해진다고 할 수 있다. 이러한 노력 중의 하나가, DCT 영역에서 움직임을 추정하는 것이었으나[11], DCT 계수 자체만으로 움직임을 추정한다는 것은 IDCT 후 공간영역에서 전역탐색 BMA (Block Matching Algorithm) 보다 복잡하고 계산 량이 많다.

Koc 등^[12-17]은 Fourier 변환에 있는 시간영역 이동 정리(Shift Property)를 이용하여 시간영역에서 움직임이 있으면, 움직임에 해당하는 만큼 변환영역에서 위상 변화가 있다는 점을 이용한 방식을 발표하였다. 이 경우 변환 영역에서 위상 변화를 구하기 위하여 별도의 DCT, DST(Discrete Sine Transform)를 계산하여야 되며 이에 대한 계산량이 많아져서 구현 시에 하드웨어가 복잡해지는 문제점이 있다. 이 방법들을 대신하여, DCT 계수를 이용한 단계적 움직임 추정 방법을 제안한다.

1. DCT 계수를 이용한 단계적 움직임 추정

DCT 계수를 이용한 단계적 움직임 추정 방법은 DCT 계수의 저주파 영역만을 IDCT하여 작은 영상을 만든다. 그림 3은 8×8의 한 DCT 블록을 나타낸다. 그림 3의 각각의 영역을 차례대로 식 (13)과 같이 IDCT

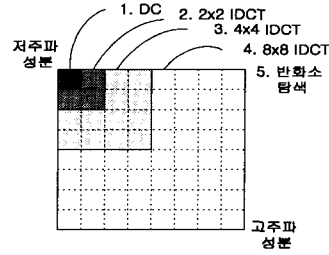


그림 3. DCT 블록에서 IDCT될 영역
Fig. 3. IDCT blocks in the DCT block.

하면, 원 영상의 1/k×1/k 크기의 영상을 얻을 수 있다. 만약 k가 8이면 DC 값만으로 만들어진 영상이 되며, 1이면 본래의 영상을 얻을 수 있다. 이때, 이들 영상은 k가 1일 때를 제외하면, 다운 샘플 및 저역통과 되었다고 할 수 있다.

$$f_k(x, y) = \frac{1}{8} \sum_{u=0}^{N_x-1} \sum_{v=0}^{N_y-1} C(ku)C(kv)F(u, v)$$

$$\cos \frac{(2x+1)ku\pi}{2N_1} \cos \frac{(2y+1)kv\pi}{2N_2}$$

where $k=1, 2, 4, 8,$ and

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } u, v=0 \\ 1 & \text{otherwise} \end{cases} \quad (13)$$

이렇게 얻어진 영상들을 가지고 블록 매칭으로 움직임을 추정하게 된다. 움직임은 4개의 DCT 블록에 해당하는 매크로 블록을 기준으로, 찾고자 하는 영역과의 차이를 비교하여 최소 차이를 가지는 곳으로 움직임 벡터를 찾는다. 우선 1/8×1/8 크기의 영상을 가지고 시작한다. 이 때는 실제 영상의 가로 세로 8 화소 단위의 움직임을 찾는 것과 같으며, 매크로 블록은 총 4개의 화소값을 갖는다. 이 경우 그림 4 (a)와 같다.

제 1단계에서 정해진 탐색 영역까지 움직임을 찾은 후, 제 2단계부터는 바로 전 단계에서 찾은 움직임에서 움직임을 찾게 되는데, 각 단계에서는 그 단계의 최소 화소 한 단위의 크기만큼 움직이며 움직임을 찾는다. 이 경우 그림 4 (b)와 같다. 그런데, 이렇게 한 단위 크기만큼 움직이면, 이론적으로 전체 영상의 모든 영역을 탐색할 수 있는 것이지만, 전 단계의 탐색에서 지역적 최소에 빠질 수 있는 가능성이 있으므로, 그림 4 (b)의 탐색 방법 II 처럼, 탐색 영역을 한 단위 더 넓혀 움직임을 찾게 할 수도 있다.

움직임 벡터를 이용하여 현재 프레임에 대한 예측 프레임을 구성한 후 DCT한다. DCT 영역의 현재 프레

입과 DCT 영역의 예측 프레임의 차를 구한 후 가변장 부호화 한다. 움직임 보상 시에는 그림 5와 같이 DCT의 선형 정리를 이용하여 예측된 프레임과 앞에서 설명한 차 프레임을 더하여 DCT 영역의 이전 프레임을 구성한다.

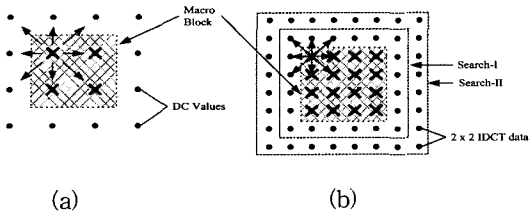


그림 4. 제안된 탐색 방법
(a) 제 1 단계 (b) 제 2 단계

Fig. 4. Proposed search methods.
(a) The first step (b) The second step

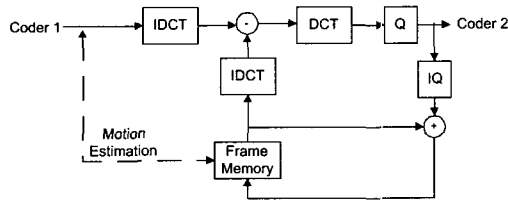


그림 5. 움직임 보상 블록도
Fig. 5. Block diagram on motion compensation.

V. 실험 결과

1. MPEG-2 인트라 변환 부호화 실험

비트율이 26Mb/s인 DV 시퀀스를 10Mb/s인 MPEG-2 하이 프로파일 인트라 압축 방식으로 변환하였다. 휘도 신호와 색차 신호에 대한 PSNR(Peak Signal-to-Noise Ratio) 비교는 표 2에 나타내었다. 변환부호화한 색차 영상인 그림 6 (c)에서 블록 효과가 나타나는 것은 선형 인터폴레이션에 이용한 행렬이(식 (2)에서 $h_{16 \times 8}$) 근사화된 것이었기 때문이다. 표 2에서 휘도신호의 PSNR에서 큰 차이가 나는 것은 두 경우 모두 비슷한 압축 방식을 씀에도 불구하고, DV 부호화 방식으로만 하는 경우는 26Mb/s로 압축하고 변환부호화 경우 MPEG-2 인트라 부호화 방식으로 10Mb/s로 압축하였기 때문이다. 이럴 경우에도 표 1에서 알 수 있듯이 색차신호의 PSNR에는 큰 영향을 주지 않는다. 색차 신호의 PSNR을 개선하기 위하여 업 샘플링 하기

전에 저역통과 필터를 이용하여야 하며, 포맷 변환 시에 선형 인터폴레이션이 아닌 고차원 수식을 이용한 인터폴레이션 방식이 필요하다.

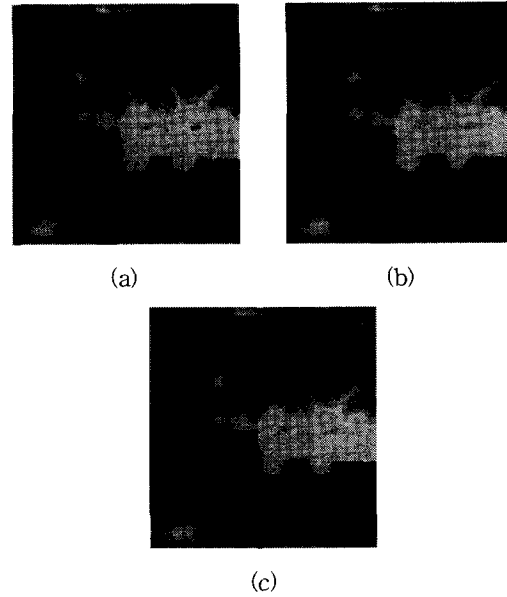


그림 6. 원영상과 복원된 CB 영상
(a) 원영상 (b) DV 코딩 (c) 트랜스코딩
Fig. 6. Original and restored CB Images.
(a) Original image
(b) Restored image from DV coding
(c) Restored image from transcoding

식 (12)에서 알 수 있듯이 두 영역에서 m_quant 를 계산하였을 때 동일한 값을 얻을 수 있다. 시간영역에서 m_quant 를 계산한 값과 DCT 영역에서 제안한 방법으로 계산한 값을 비교하였을 때 결과가 동일함을 확인하였다.

표 2. 비디오 코딩방식에 따른 PSNR 비교
Table. 2. PSNR comparison for two video coders.

	Y (dB)	C _B (dB)	C _R (dB)
DV 코딩	30.62	32.24	33.0
변환코딩	25.73	31.13	31.35

2. MPEG-2 인터 변환 부호화 실험

MPEG-2에서 프레임 픽처 부호화 시 프레임 예측과 필드 예측을 동시에 이용할 수 있다. 먼저 프레임 영상

에 대한 성능을 알아보기 위하여 720×480 크기의 "Mobile & Calendar" 프레임 픽처로 실험 하였고, 시퀀스는 M=3, N=12의 MPEG-2로 변환부호화 하였다. 그리고 전송 비트율은 5Mb/s를 기준으로 하였고, 색차신호 형식은 4:2:2로, P 픽처의 경우 순방향 탐색 영역은 ±15로 B 픽처의 경우 순방향 및 역방향 탐색 영역은 ±7로 하여 비교하였으며, 시퀀스별 25장의 프레임을 사용하여 실험하였다.

PSNR 측정 결과를 그림 7에 나타내었다. 여기서 4단계 탐색이라는 것은 제 1단계 탐색을 8화소 간격으로 탐색하지 않고, 4화소 간격으로 탐색 한 방법인데, 이것은 처음 움직임을 추정할 때 지역적 최소가 되는 것을 줄여보자 힘이었는데, 5단계 탐색과 별 차이가 없는 것으로 드러났다. 그러나, 탐색-I과 탐색-II는 다소 차이를 보였다. 전역탐색 방법과 비교하여 탐색-I은 평균 1.16dB의 차를 보였으며, 탐색-II는 평균 0.31dB의 차를 나타냈다.

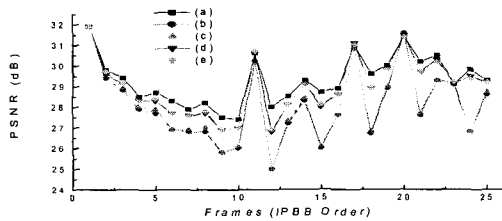


그림 7. 프레임 픽처를 프레임 예측 부호화 시 움직임 추정 방법에 대한 PSNR 비교
 (a) 전역탐색, progressive_sequence=1
 (b) 탐색-I, 5 단계 탐색
 (c) 탐색-I, 4 단계 탐색
 (d) 탐색-II, 5 단계 탐색
 (e) 탐색-II, 4 단계 탐색

Fig. 7. PSNR comparison on ME methods with frame prediction and frame picture.
 (a) Full search with progressive_sequence=1
 (b) Method-I with 5-step search
 (c) Method-I with 4-step search
 (d) Method-II with 5-step search
 (e) Method-II with 4-step search

PC에서 소프트웨어로 변환부호화 하였을 때 움직임 추정의 수행 시간을 측정해 본 결과는 표 3과 같다. 전역탐색 방법은, 탐색 영역이 커질수록 수행 시간 증가율이 큰데 반하여, DCT 계수를 이용한 단계적 움직임 추정 방법은 탐색 영역이 커지더라도 수행 시간 증가

표 3. 제안된 방법들의 수행시간 비교

Table 3. PC run time of proposed methods.

탐색 방법	5 단계 탐색		4 단계 탐색		전역 탐색
	방법-I	방법-II	방법-I	방법-II	
P 2 Frames	3.8 초	5.1 초	4.3 초	5.8 초	34.4 초
B 3 Frames	6.4 초	8.9 초	7.6 초	8.9 초	21.1 초

율이 적다. 표 2에서 제안한 방식은 탐색 시간이 탐색 영역 크기에 영향을 적게 받지만 시간영역 전역탐색의 경우는 탐색 시간이 탐색영역 크기에 크게 영향을 받아 탐색영역이 큰 P 픽처의 탐색 시간이 B 픽처의 탐색 시간보다 크다. 탐색-I과 탐색-II는 탐색 시간이 비슷한 데 반하여, 움직임 추정 효과는 탐색-II가 더 나으므로, 탐색-II의 효율성이 더 좋을 것으로 생각된다.

만약, 영상이 상위 필드와 하위 필드가 합쳐진 것이고, 두 필드의 움직임 차이가 크다면, 프레임에 대해 처리된 DCT 계수를 이용한 단계적 움직임 추정 방법은 정확한 움직임을 추정하지 못한다. "Flower Garden" 영상은 필드 픽처이다. 그림 8은 이러한 영상을 프레임 예측으로 부호화 했을 때 PSNR을 보여준다.

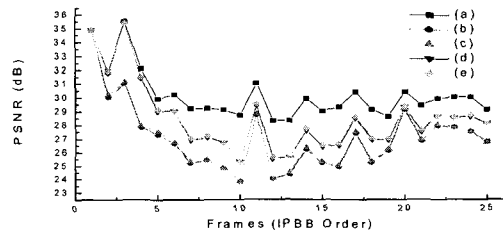


그림 8. 필드 픽처를 프레임 예측 부호화 시 움직임 추정 방법에 대한 PSNR 비교
 (a) 전역탐색, progressive_sequence=0
 (b) 탐색-I, 5 단계 탐색
 (c) 탐색-I, 4 단계 탐색
 (d) 탐색-II, 5 단계 탐색
 (e) 탐색-II, 4 단계 탐색

Fig. 8. PSNR comparison on ME methods with frame prediction and field picture.
 (a) Full search with progressive_sequence=0
 (b) Method-I with 5-step search
 (c) Method-I with 4-step search
 (d) Method-II with 5-step search
 (e) Method-II with 4-step search

프레임 픽처는 프레임 예측 모드를 선택해야 하며, 이 경우 프레임 예측 모드는 픽처의 특성에 따라 적절적으로 프레임 예측 또는 필드 예측을 하여 압축효율을 높인다. 필드 픽처에 대해서는 필드 예측 모드를 선택해야 하며, 이 경우 필드 예측 모드는 필드 픽처를 필드 예측으로만 코딩한다. 그러나, DV 방식에서는 입력 영상을 항상 프레임으로 DCT하기 때문에 제안한 방식으로 변환부호화 하는 경우 프레임 예측만이 가능하다. 그러므로 필드 픽처인 경우 그림 8과 같이 다소 성능이 떨어진다.

필드 픽처들로 구성된 "Susie" 시퀀스에 대하여 같은 실험을 반복하였다. 탐색-II와 5 단계 탐색을 했을 때 그림 9와 같이 평균 1.10dB 차이가 나며, 이것은 같은 조건에서 "Flower Garden"의 평균 1.62dB와 비슷한 결과를 보여준다. 이번에는 "Susie" 시퀀스에 대해 전역 탐색 및 프레임 예측 보호화를 한 것과 제안한 방법과 비교하면 평균 0.57dB 차이가 나며, 이것은 "Mobile & Calendar" 시퀀스의 평균 0.31dB와 비슷한 결과이다. 여기서 다소 차이가 나는 것은 "Susie" 영상의 45번째 프레임부터 잘라서 실험에 사용한 부분이 "Mobile & Calendar" 시퀀스보다 그림의 변화가 심하기 때문이다.

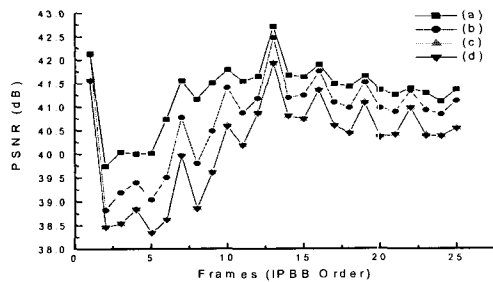


그림 9. 필드 픽처를 프레임 예측 부호화 시 움직임 추정 방법에 대한 PSNR 비교

- (a) 전역탐색, progressive sequence=0
- (b) 전역탐색, progressive sequence=1
- (c) 탐색-II, 5 단계 탐색 (d) 탐색-II, 4 단계 탐색

Fig. 9. PSNR comparison on ME methods with frame prediction and field picture.

- (a) Full search with progressive sequence=0
- (b) Full search with progressive sequence=1
- (c) Method-II, 5-step search (d) Method-II, 5-step search

VI. 결 론

DV 방식 동영상 부호화기에서 MPEG-2로 변환하는 방법에 대하여 서술하였다. 첫 번째 부호화기에서 DCT로 압축된 영상을 DCT 영역에서 MPEG-2의 인트라 프레임으로 변환 및 압축하였다. DV 방식의 4:1:1 색차 포맷을 MPEG-2의 4:2:2 영상 포맷으로 변환할 때와 2-4-8 DCT 모드에서 8-8 DCT 모드로 변환 시 고속 DCT 알고리즘의 Butterfly 구조를 사용하지 않고 병렬 처리가 가능하고 보다 정확한 계산을 수행하는 방법을 제안하였다. DCT 영역에서 DCT의 에너지 정리와 DCT의 DC 계수를 이용하여, 시간영역에서 계산한 값과 동일한 m_quant를 계산하는 방법을 제안하였다. 시간영역에서 m_quant를 계산한 값과 DCT 영역에서 제안한 방법으로 계산한 값을 실험을 통하여 비교하였을 때 결과가 동일함을 확인하였다. 본 논문에서 MAE 혹은 MAP를 이용한 재 양자화 방법[4]을 사용하지 않았지만 MPEG-2 인트라 프레임으로 변환부호화 할 때 이 방법을 쓰면 화질 개선이 예상된다.

DCT 압축방식을 이용하는 DV 방식을 MPEG-2로 변환부호화 할 때 하드웨어 혹은 소프트웨어 복잡도를 줄이기 위하여 DCT 계수를 이용한 고속 움직임 추정 방법들을 제안하였다. DCT 계수를 이용한 단계적 탐색 방법에서 5단계 탐색과 4단계 탐색의 성능 차이는 거의 없었다. 필드 픽처에 제안한 방법을 적용했을 때 프레임 픽처 경우보다 다소 성능이 떨어졌다. 이것은 DV 방식에서 항상 프레임으로 DCT 하기 때문이며 따라서 필드 정보를 얻을 수 없기 때문이다. 두 종류의 픽처에 대해서 중첩된 탐색 영역을 이용하는 탐색 방법-II가 탐색 방법-I 보다 성능이 우수하였다. 그러므로, MPEG-2 인트라 변환 부호화 시 DCT 계수를 이용한 고속 5단계 탐색 및 탐색 방법-II를 이용하면 시간영역의 전역탐색의 PSNR 성능에 근접하면서 시간영역의 전역탐색보다 계산속도가 빠른 변환부호화기를 구현할 수 있다.

참 고 문 헌

[1] Specifications of Consumer-Use Digital VCRs Using 8mm Magnetic Tape, HD Digital Conference, Dec. 1994.

- [2] ITU-T Recommendation H262/ISO-IEC 13838-2, MPEG-2, 1996.
- [3] A. Eleftheriadis and D. Anastassiou, "Constrained and General Dynamic Rate Shaping of Compressed Digital Video", in Proc. 2nd IEEE Int. Conf. Image Processing, pp. 1-4, Arlington, VA, Oct. 1995.
- [4] Oliver Werner, "Requantization for Transcoding of MPEG-2 Intraframes," IEEE Trans. on Image Processing, vol. 8, pp. 179-191, Feb. 1999.
- [5] P. A. A. Assuncao and M. Ghanbari, "A Frequency-Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams," IEEE Trans. on Circuits and Systems for Video Technology, vol. 8, pp. 953-967, Dec. 1998.
- [6] A. K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, pp. 253-255, 1989.
- [7] Y. Arai, T. Agui, and M. Nakajima, "A Fast DCT-SQ Scheme for Images," Trans. of the IEICE, E 71(11), pp. 1095-1097, Nov. 1988.
- [8] MPEG-2 Software Simulation Group, "MPEG-2 Video Source Coding Software".
- [9] Test Model Editing Committee, Test Model 5, ISO/IEC JTC1/SC29/WG11/N0400, April 1993.
- [10] Jae S. Lim, Two-dimensional Signal and Image Processing, Prentice Hall, p. 158, 1990.
- [11] P. Yip and K. R. Rao, "On the Shift Property of DCT's and DST's", IEEE Trans. on Acoust., Speech, Signal Processing, vol. ASSP-35, no. 3, pp. 404-406, Mar. 1987.
- [12] U. V. Koc and K. J. R. Liu, "Discrete-Cosine/Sine Transform based motion estimation", Proc. of IEEE Inter. Conf. on Image Processing, vol. 3, pp. 771-775, Austin, Texas, Nov. 1994.
- [13] U. V. Koc and K. J. Liu, "Adaptive Overlapping Approach for DCT-based Motion Estimation", Proc. of IEEE Inter. Conf. on Image Processing, vol. I, pp. 223-226, Washington DC, Oct. 1995.
- [14] U. V. Koc and K. J. Liu, "Subpixel Motion Estimation in DCT Domain", Proc. of SPIE Visual Comm. and Image Processing, pp. 332-343, Orlando, FL, Mar. 1996.
- [15] U. V. Koc and K. J. Liu, "DCT-based Subpixel Motion Estimation", Proc. of IEEE Int. Conf. Acoustics, Speech, Signal Processing, vol. IV, pp. 1931-1934, Atlanta, GA, 1996.
- [16] U. V. Koc and K. J. Liu, "DCT-based Subpixel Motion Estimation and Fully DCT-based Video Coder", Proc. of IEEE Inter. Conf. on Image Processing, vol. III, pp. 598-601, Santa Barbara, CA, Oct. 1997.
- [17] U. V. Koc and K. J. Liu, "DCT-based Motion Estimation", IEEE Trans. on Image Processing, vol. 7, no. 7, pp. 948-965, July 1998.

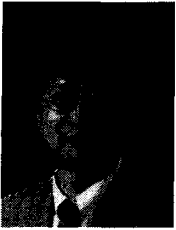
저 자 소 개



金 度 年(學生會員)
 1985년 경북대학교 전자공학과 학사. 1988년 연세대학교 전자공학과 석사. 1989년 8월 ~ 1998년 1월 한국전자통신연구원에서 "디지털 TV" 개발. 1998년 1월 ~ 현재, 연세대학교 전기전자공학과 박사과정



尹 汎 植(正會員)
 1997년 연세대학교 전기공학과 학사. 2000년 연세대학교 전기및컴퓨터공학과 석사. 2000년 1월 ~ 현재, 삼성전자 중앙연구소에서 "Digital Media System" 개발.



崔潤植(正會員)

1979년 2월 연세대학교 전기공학과
 학사. 1984년 5월 미국 Case
 Western Reserve Univ., Dept. of
 System Eng. 석사. 1987년 5월 미국
 Pennsylvania 주립대 Dept. of E. E.
 석사. 1990년 12월 미국 Purdue 대
 학 전기공학부 박사. 1990년 11월~ 1993년 2월 (주)현
 대전자 산업전자연구소 책임연구원. 1993년 3월~ 현재,
 연세대학교 공과대학 전기전자공학과 부교수