

일반논문-10-15-3-12

H.264/AVC에서 3차원 변환에 기반을 둔 효율적인 동영상 압축 방법

조재규^{a)‡}, 조혜정^{a)}, 이진호^{b)}, 정세윤^{b)}, 안창범^{a)}, 오승준^{a)}

The Efficient Coding Tools based 3-Dimensional Transform in H.264/AVC

Jaekyu Jo^{a)‡}, Hye-Jeong Cho^{a)}, Jinho Lee^{b)}, Seyoon Jeong^{b)}, Changbeom Ahn^{a)}, and Seoungjun Oh^{a)}

요약

본 논문에서는 H.264/AVC에서 사용하는 정수형 변환을 4x4 블록 단위로 적용하고 인접 네 개의 4x4 블록의 같은 주파수 위치의 계수값을 모아서 추가로 4x1 블록 단위로 1차원 변환하는 3차원 변환(3DT : Dimensional Transform)을 기반으로 부호화 방법을 통합한 환경인 3DTE(3DT Environment)를 제안한다. 제안하는 방법에서 사용하는 변환인 3DT는 H.264/AVC가 가지는 예측오차를 유지하면서 인접블록 간에 남아있을 공간적 중복도를 추가로 줄일 수 있고 이와 같은 특성에 적합한 방법을 적용하였다. 본 논문의 실험에서 제안하는 방법은 JM11.0과 비교 실험했을 때 대표적인 시험 시퀀스에 대하여 High profile 환경에서 평균적으로 약 3.58% 정도의 비트율 절감을 얻었다. 특히 HD급과 같이 해상도가 높은 영상에서 평균 약 5.40% 정도의 높은 비트율 절감을 보였다.

Abstract

In this paper, we propose 3DTE(3-Dimensional Transform Environment) that is based on 3DT(Dimensional Transform) that performs 2-dimensional integer DCT(Discrete Cosine Transform) based on 4x4 block and 1-dimensional integer DCT based on 4x1 block after collecting same frequency coefficients in neighboring 4x4 block and supports its additional coding tools for high performance. The transform of 3DT can keep prediction error by using 4x4 block and reduce spatial redundancy additionally. The proposed 3DTE can provide coding tools to improve the coding efficiency with using 3DT. The performance of 3DTE compared to JM11.0 is average 3.58% and 5.40% bit savings for all test sequences and HD sequences, respectively, with keeping subjective video quality in High profile.

Keywords : 3DTE, 3DT, additional coding tools, prediction error, spatial redundancy

1. 서론

가장 최근에 제정된 H.264/AVC표준은 국제 표준화 기구인 MPEG(Moving Picture Expert Group)과 VCEG (Video Coding Expert Group)으로 구성된 JVT(Joint Video Team)에 의해 공동 개발되었다. H.264/AVC표준은 기존의 표준보다 우수하고 뛰어난 비디오 이미지 압축 성능을 갖는다. 이것은 화면 내 예측 부호화, 1/4단위의 움직임 보상

a) 광운대학교 VIA-멀티미디어 센터

VIA-Multimedia Center at Kwangwoon University

b) 한국전자통신연구원

ETRI

‡ 교신저자 : 조재규(jaeku220@media.kw.ac.kr)

※ 본 연구는 방송통신위원회, 지식경제부 및 한국산업기술평가관리원 IT 원천기술개발사업의 일환으로 수행한 연구[과제관리번호:2008-F-011, 과제명:차세대DTV 핵심기술 개발]와 지식경제부, 한국산업기술진흥원의 전략기술인력양성사업으로 수행된 결과임

· 접수일(2010년4월15일), 수정일(2010년5월5일), 게재확정일(2010년5월27일)

(motion estimation/compensation) 및 다양한 블록크기의 움직임 예측, 4x4 블록단위의 정수형 변환(4x4 integer DCT(Discrete cosine transform)), in-loop deblocking filter, CAVLC (Context-based Adaptive Variable Length Coding) 와 CABAC(Context-based Binary Arithmetic Coding) 등의 기술을 이용한 것에 기인한다^{[1][2]}.

특히 H.264/AVC표준이 기존의 표준보다 뛰어난 성능개선을 보인 요인 중의 하나는 4x4 블록까지 움직임 예측을 한 것이다. 기존의 표준에 사용된 블록크기보다 작은 4x4 블록의 사용은 예측블록크기가 작아짐으로써 예측오차를 줄일 수 있는 효과를 얻을 수 있다. 일반적으로 예측블록의 크기가 작을수록 예측오차를 줄일 수 있고 예측블록의 크기가 클수록 변환했을 때 공간적 중복도를 효율적으로 줄일 수 있다^[3]. H.264/AVC표준은 4x4 블록 단위까지 움직임 예측 및 보상하여 예측오차를 줄임으로써 기존의 비디오 표준안보다 성능을 개선시키지만 4x4 블록단위의 정수형 변환을 사용함으로써 큰 블록의 변환에 비해 공간적 중복도를 줄이는 효과는 감소시킨다.

본 논문에서는 H.264/AVC가 가지는 예측성능을 유지하면서 공간적 중복도를 효율적으로 줄일 수 있는 3차원 변환(3DT : 3-Dimensional Transform)을 기반으로 다양한 방법으로 부호화하는 3DTE(3-Dimensional Transform Environment)를 제안한다. 3DTE에서 사용하는 변환인 3DT는 H.264/AVC에서 사용하고 있는 4x4 블록단위의 2차원 정수형 변환을 수행한 후에 인접한 네 개의 4x4 블록의 같은 주파수 위치의 계수를 모아서 4x1 블록단위의 1차원 정수형 변환을 수행한다. 이와 같이 3DT를 수행할 때 블록모드에 따라 방향성을 고려하여 네 개의 인접 4x4 블록 형성 방법을 변형한다. 그리고 3DT는 정수형 변환에 기초를 두기 때문에 양자화 과정에도 영향을 준다. 따라서 두 번의 정수형 변환을 고려한 양자화기를 설계하고 양자화 후에 엔트로피 부호화를 효율적으로 하기 위해 3DT에 적합한 주사 방식 및 임계값 수정 등 여러 방법을 제안함으로써 부호화의 효율을 높인다.

본 논문의 전체적인 구성은 다음과 같다. 2장에서는 H.264/AVC에서 적용하고 있는 변환과 양자화에 대한 소개를 하고, 3장에서는 제안하고자 하는 3DTE에 사용되는 3DT와 이를 기반으로 효율적으로 부호화할 수 있도록 하

는 여러 가지 부호화 방법에 대해 서술한다. 4장에서는 3DT에 적합한 정수형 양자화기를 설계한 방법에 대해 설명하고 5장에서는 제안하는 방법에 대한 실험결과를 보여 주며 6장에서 결론을 맺는다.

II. H.264/AVC에서의 4x4 블록단위의 2차원 정수형 변환 및 양자화

H.264/AVC는 이전의 표준과는 달리 4x4 블록크기까지 예측함에 따라 4x4 블록단위의 2차원 정수형 변환과 이에 적합하게 설계된 양자화기를 사용한다^[4]. 본 논문에서는 4x4 블록단위의 2차원 정수형 변환을 4x4 HT(H.264 Transform)라고 부르기로 한다. 4x4 HT는 정수 산술 연산을 사용하기 때문에 복호기의 변환 결과와 부호기의 변환기 입력값이 정확히 일치한다. 그리고 4x4 HT의 핵심변환부는 덧셈과 시프트(shift) 연산만으로 구현할 수 있으며, 4x4 HT와 양자화기가 병합된 단순한 형태를 가지고 있다. 4x4 HT는 DCT(Discrete Cosine Transform)에 기반을 둔다.

$$Y = AXA^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} X \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix} \quad (1)$$

식 (1)은 4x4 블록 잔차신호 행렬 X 에 대한 4x4 블록단위의 DCT 연산이다. 식 (1)에서 a, b, c는 각각 1/2, $\sqrt{1/2} \cos(\pi/8)$, $\sqrt{1/2} \cos(3\pi/8)$ 이다. 계산을 간단히 하기 위해 식 (1)에서 스케일 행렬만 빼서 식 (2)와 같이 만들어 준다.

$$Y = CXC^T \otimes E = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} X \begin{bmatrix} 1 & 1 & 1 & d \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \quad (2)$$

식 (2)에서 a, b, d는 각각 1/2, $\sqrt{2/5}$, 1/2이다. 행렬 CXC^T 는 2차원 변환의 코어행렬연산 부분이다. 또한 \otimes

은 행렬의 같은 위치 요소간의 곱셈으로 정의한다. d 는 c/b 의 값으로 구해져 0.414 값이 나오지만 계산의 간략화를 위해서 0.5 로 한다. 그리고 행렬 C 의 2, 4 번째 행과 행렬 C^T 의 2, 4 번째 열에 2를 각각 곱한다. 이는 변환의 코어부분을 계산할 때 1/2 을 곱하는 연산을 피하기 위함이다. 이 과정을 거쳐 최종적으로 만들어진 순방향 4x4 블록단위의 2차원 정수형 변환은 식 (3)과 같이 정리된다.

$$Y = C_f X C_f^T \otimes E_f$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} X \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \quad (3)$$

역방향 2차원 정수형 변환은 식 (4)와 같이 순방향 2차원 정수형 변환과 반대로 먼저 역방향 스케일 행렬과 양자화된 계수값 사이에 4x4 행렬의 같은 위치의 요소 간에 곱셈을 수행하고 역방향 코어행렬을 곱해준다.

$$X' = C_i^T (Y \otimes E_i) C_i$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left(\begin{bmatrix} Y \\ \otimes \end{bmatrix} \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix} \quad (4)$$

H.264/AVC에서 사용하는 양자화를 나타내는 식은 식 (5)와 같다.

$$Z_{ij} = \text{round}\left(\frac{Y_{ij}}{Qstep}\right) \quad (5)$$

이때 i, j 는 각각 4x4 블록내의 x, y 좌표값을 가리키고 Y_{ij} 는 식 (3)에서 구한 변환 계수값이며 Z_{ij} 는 양자화된 값이다. $Qstep$ 은 양자화 스텝 크기를 나타내는데 H.264/AVC에서는 총 52가지의 $Qstep$ 을 사용하고 QP (Quantization Parameter)값에 의해 표현된다. 표 1은 $Qstep$ 과 QP 와의 관계를 보여준다. $Qstep$ 이 2배 증가할 때마다 QP 는 6씩 증가하는 특징을 가진다. 식 (5)는 PF (Post-scaling Factor)에 의해서 식 (6)으로 다시 표현할 수 있다.

$$Z_{ij} = \text{round}\left(W_{ij} \times \frac{PF}{Qstep}\right) \quad (6)$$

이때 W_{ij} 는 코어 변환과정인 $C_f X C_f^T$ 에 의해 나온 결과값이고 PF 는 식 (3)에서 표현된 E_f 행렬의 x, y 좌표에 해당하는 요소값이다. 식 (7)에서 PF 를 MF (Multiplication Factor)로 바꿔 정수형 양자화 계산을 할 수 있도록 하고, 이를 표 2의 (a)에서 QP 와 4x4 블록 내 위치에 따라 정의한다. 식 (7)은 MF 로 표현된 양자화 과정의 수식을 나타낸다.

$$Z_{ij} = \text{round}\left(W_{ij} \times \frac{MF}{2^{qbits}}\right) \quad (7)$$

식 (7)에서 $PF/Qstep = MF/2^{qbits}$ 이 성립하고, 이때 $qbits = 15 + \text{floor}(QP/6)$ 이며, 양자화 파라미터(f)와 시프트 연산으로 표현한 양자화 수식은 식 (8)과 같다.

$$Z_{ij} = \text{sign}(W_{ij}) \times (|W_{ij}| \cdot MF + f \cdot 2^{qbits}) \gg qbits \quad (8)$$

역양자화 과정은 역변환에 사용된 PF' 을 사용하여 계산된다. 이 과정을 정리하면 식 (9)와 같다. 이때 PF' 는 식 (4)

표 1. H.264/AVC에서 Qstep과 QP와의 관계
Table. 1. Relationship between Qstep and QP in H.264/AVC

QP	0	1	2	3	4	5	6	7	8	9	10	11	12	...
$Qstep$	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25	2.5	...
QP		18	...	24	...	30	...	36	...	42	...	48	...	51
$Qstep$		5		10		20		40		80		160		224

표 2. QP와 4x4 블록 내 위치에 따른 MF와 V
Table 2. Multiplication factor MF and Rescaling factor V

QP	Positions (0,0),(2,0),(2,2),(0,2)	Positions (1,1),(1,3),(3,1),(3,3)	Other positions
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

(a) MF

QP	Positions (0,0),(2,0),(2,2),(0,2)	Positions (1,1),(1,3),(3,1),(3,3)	Other positions
0	10	16	13
1	11	18	14
2	13	20	16
3	14	23	18
4	16	25	20
5	18	29	23

(b) V

에서 사용된 E_i 행렬의 x, y 좌표에 해당하는 요소값을 가리킨다. 식 (10)은 $Qstep, PF'$, 64를 곱하여 $V \cdot 2^{\text{floor}(QP/6)}$ 로 표현한 수식이고 V (Rescaling Factor)는 QP와 4x4블록 위치에 따른 값으로 표 2의 (b)와 같이 정리된다.

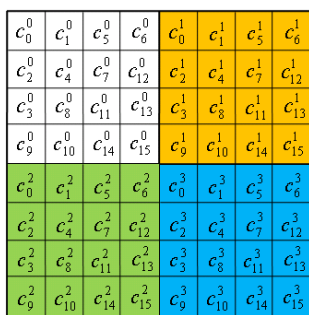
$$W'_{ij} = Z_{ij} \cdot Qstep \cdot PF' \cdot 64 \quad (9)$$

$$W'_{ij} = Z_{ij} \cdot V \cdot 2^{\text{floor}(QP/6)} \quad (10)$$

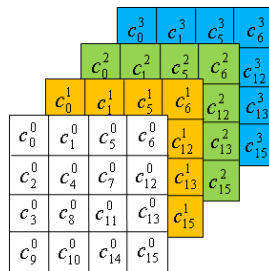
III. 3DT를 사용한 부호화 방법

H.264/AVC에서는 4x4 블록단위로 예측하고 변환함으로써 예측오차를 줄일 수 있는 반면에 4x4 블록보다 큰 블록변환에 비해서 공간적 중복도를 줄이는 효과는 상대적으로 저하된다. 따라서 H.264/AVC가 가지는 예측오차를 유

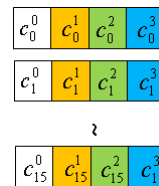
지하면서 공간적 중복도를 줄일 수 있는 3DT를 제안한다. 그림 1과 같이 3DT는 네 개의 4x4 블록에 대해 4x4 HT를 하고 같은 주파수 네 개를 묶어서 1차원 정수형 변환, 즉 4x4 HT의 1차원 코어행렬에 기반한 4x1 변환을 수행한다. 앞으로 4x1 변환을 4x1 HT라 한다. c'_i 는 8x8 블록내에 4x4 HT후의 주파수위치를 나타낸다. 이때 i, j 는 각각 4x4 블록 내 주파수 위치와 8x8 블록안에 4개의 4x4 블록 인덱스를 나타낸다. 그림 2의 실험은 Foreman영상, 100 프레임, IPPP구조, QP=28, High 프로파일에서 4x4 HT, 8x8 HT, 3DT를 사용하고 율-왜곡 최적화 과정(Rate-Distortion Optimization : RDO)을 통해서 최적의 모드가 16x16, 16x8, 8x16, 8x8 중에서 선택되고 최적의 변환이 3DT일때 8x8 블록 내에서의 평균 에너지 분포를 나타낸 것이다. 이 실험은 3DT 및 양자화 후 각각의 주파수별로 계수값을 더하여 축적한다. 8x8 블록내의 각 주파수에 대해서 축적한 모든 계수값을 더하여 이를 전체(T)로 할 때 각각의 축적한



(a) 8x8 블록내 네 개의 4x4 블록에 대한 4x4 HT



(b) 4x4 HT 수행 후 네 개의 같은 주파수를 모으는 과정



(c) 네 개의 같은 주파수에 대해 4x1 HT를 수행

그림 1. 3DT 과정
Fig. 1. 3DT process

주파수 계수값 크기의 비율을 구한다. 식 (11)에서 3DT 및 양자화를 수행한 후 그림 1에서 표현한 주파수 위치를 이용하여 각 주파수별 평균 에너지비율을 수식으로 나타낸다.

$$p_i^j(\%) = \frac{d_i^j}{T} \times 100 \quad (11)$$

식 (11)에서 $d_i^j = \sum_n \sum_{k=0}^4 c_i^j$, $T = \sum_{j=0}^4 \sum_{i=0}^{16} d_i^j$ 을 나타내

고 d_i^j 을 표현하기위해 쓰인 k 는 하나의 매크로블록 내에 존재하는 네 개의 8x8 블록이며, n 은 100 프레임 중 3DT가 최적의 변환으로 선택된 매크로블록 개수이다. i, j 는 각각 4x4 블록내 위치, 네 개의 4x4 블록 인덱스이다. 그림 2의 네 개의 그래프 각각의 세로축이 8x8 블록 내의 각각의 주파수 계수값에 대한 평균 에너지 비율 p_i^j 를 나타낸다.

그림 2에서 나타낸 네 개의 그래프는 8x8 블록내에서

3DT 후 에너지 분포를 쉽게 알 수 있도록 그림 1 (a)의 계수 위치와 일치시켰다. 즉 $c_i^0(i=0,1,\dots,15)$ 는 그림 2 (a), $c_i^1(i=0,1,\dots,15)$ 는 그림 2 (b), $c_i^2(i=0,1,\dots,15)$ 는 그림 2 (c), $c_i^3(i=0,1,\dots,15)$ 은 그림 2 (d) 를 나타낸다. 이 그림은 3DT와 양자화를 수행한 결과로 얻어진 dc 에너지 즉, c_0^0 의 평균 에너지가 매우 크기 때문에 잔차신호 성분에 대하여 기존하는 4x4 HT를 수행한 후에도 동일 주파수 성분 간에 중복성이 잔재하고 있다는 것을 보여준다. 따라서 4x4 HT를 수행한 후 얻어진 주파수 값들에 대하여 4x1 HT를 추가적으로 수행하여 잔존하는 중복성을 줄여줌으로써 부호화 효율을 향상시킬 수 있음을 예측할 수 있다.

본 논문에서는 4x4 HT 후 인접 블록간의 공간적 중복도를 효율적으로 제거하기 위하여 4x4 HT를 수행한 후 같은 주파수 위치의 계수를 모아서 추가적으로 1차원 변환을 하는 3DT(3-Dimensional Transform)를 제안한다. 그리고 3DT는 H.264/AVC에서 사용하는 변환과는 다른 변환이기

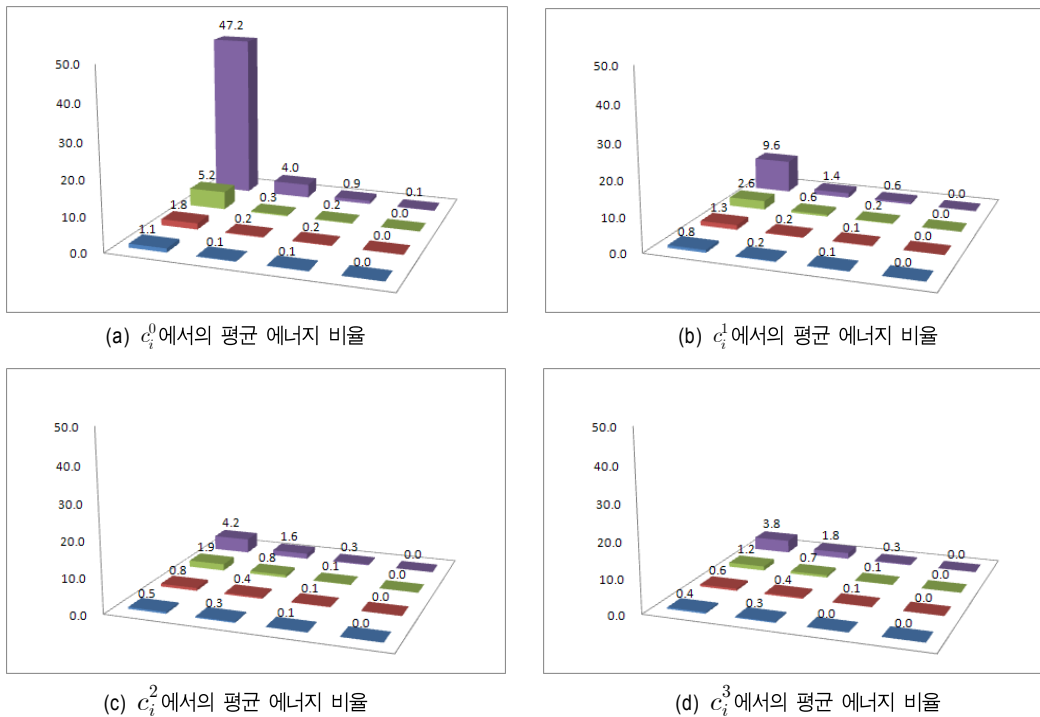


그림 2. 8x8 블록내에서 네 개의 4x4 블록의 같은 주파수 성분들 간의 공간적 중복도
 Fig. 2. Redundancy among each four co-located frequency coefficients in four 4x4 blocks of 8x8 block

때문에 이에 적합한 양자화기를 설계하고, 3DT를 기반으로 부호화 효율을 높이기 위한 모드의 방향성에 따른 블록 형성방법, 3DT에 적합화된 주사 방법 및 3DT후의 에너지 분포특성에 따른 양자화 파라미터 수정, 엔트로피 부호화를 효율적으로 하기위한 새로운 임계값 설정 등을 제안한다. 상기한 3DT와 이를 지원하는 환경을 통합하여 3DTE (3DT Environment)라 부르기로 한다. 3DTE는 화면 간 (Inter) 프레임을 부호화할 때 16x16, 16x8, 8x16, 8x8 모드에 대해 적용될 수 있다.

1. 모드에 따른 3DT 알고리즘

3DT는 그림 1과 같이 네 개의 4x4 블록에 대해 4x4 HT를 수행하고 같은 주파수 계수 네 개를 묶어서 4x1 HT를 수행하여 인접 4x4 블록간에 남아있을 중복도를 줄이는 것이다. 그림 1에서 보여준 과정은 단순히 8x8 블록 내의 네 개의 4x4 블록에 대해서 4x4 HT후 같은 주파수 위치의 계수 네 개의 값을 모아서 4x1 HT한 것을 나타낸 것이다. 제안하는 방법에서는 그림 1의 과정에서 네 개의 계수를 모을 때 그림 3과 같이 화면 간 블록의 모드에 따라 네 개의 인접 4x4 블록 형성 방법을 달리한다.

제안하는 3DT를 위한 네 개의 4x4 블록을 형성하는 방법은 그림 3에서 보여주듯이 8x8 정방향, 16x4 수평방향,

4x16 수직방향 블록이 있다. 일반적으로 모드 방향성은 블록의 상관도에 영향을 미친다.^[5] 따라서 16x8 모드가 RDO를 통해서 최적의 모드로 선택될 경우 수평방향으로 특성이 강하다고 판단할 수 있기 때문에 추가 비트사용 없이 16x4 블록형태를 고정적으로 사용하여 3DT를 수행한다. 마찬가지로 8x16 모드가 RDO를 통해 최적의 모드로 선택될 경우 수직방향으로의 특성을 반영하기 위해 4x16 블록형태를 사용한다. 16x16, 8x8 모드가 최적의 모드일 때 특정 방향성을 가지지 않는 정방향의 8x8 블록형태를 적용한다.

그림 4는 모드에 따른 블록형성방법의 효율성을 통계적으로 정리한 것으로써 16x8, 8x16 각각의 모드에 8x8 블록형성방법을 적용한 경우(블록형성방법 적용전)와 각각 16x4, 4x16 블록형성방법을 적용한 경우(블록형성방법 적용후)의 3DT 선택율을 비교한다. 3DT 선택율은 16x8, 8x16 모드 각각이 최적의 모드로 선택된 개수를 전체로 했을 때 3DT가 선택되는 개수를 비율적으로 나타낸 것이다. 여기서 16x16, 8x8 모드는 H.264/AVC에서 사용하는 정방향 8x8 블록형성방법을 그대로 사용하므로 이에 대한 자료는 제외하였다. 실험영상으로 CIF(352x288) 영상인 Foreman, Container, Tempete과 720p(1280x720) 영상인 Bigships, City 를 사용하였다. 프레임 수는 100장, 적용한 QP 대역은 22, 27, 32, 37 이다.

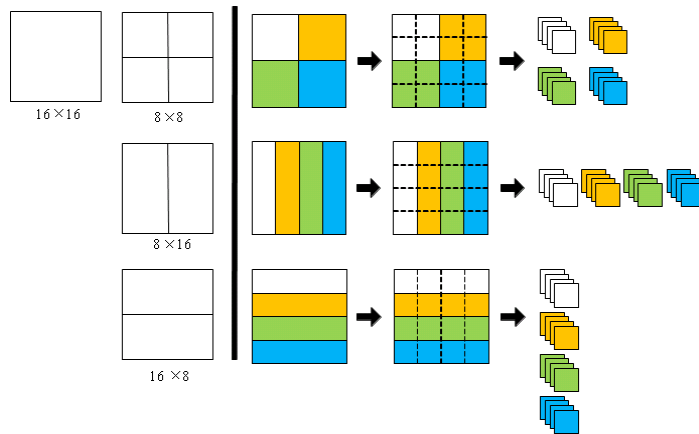


그림 3. 모드에 따른 블록형성 방법
Figure 3. Mode-based block forming method

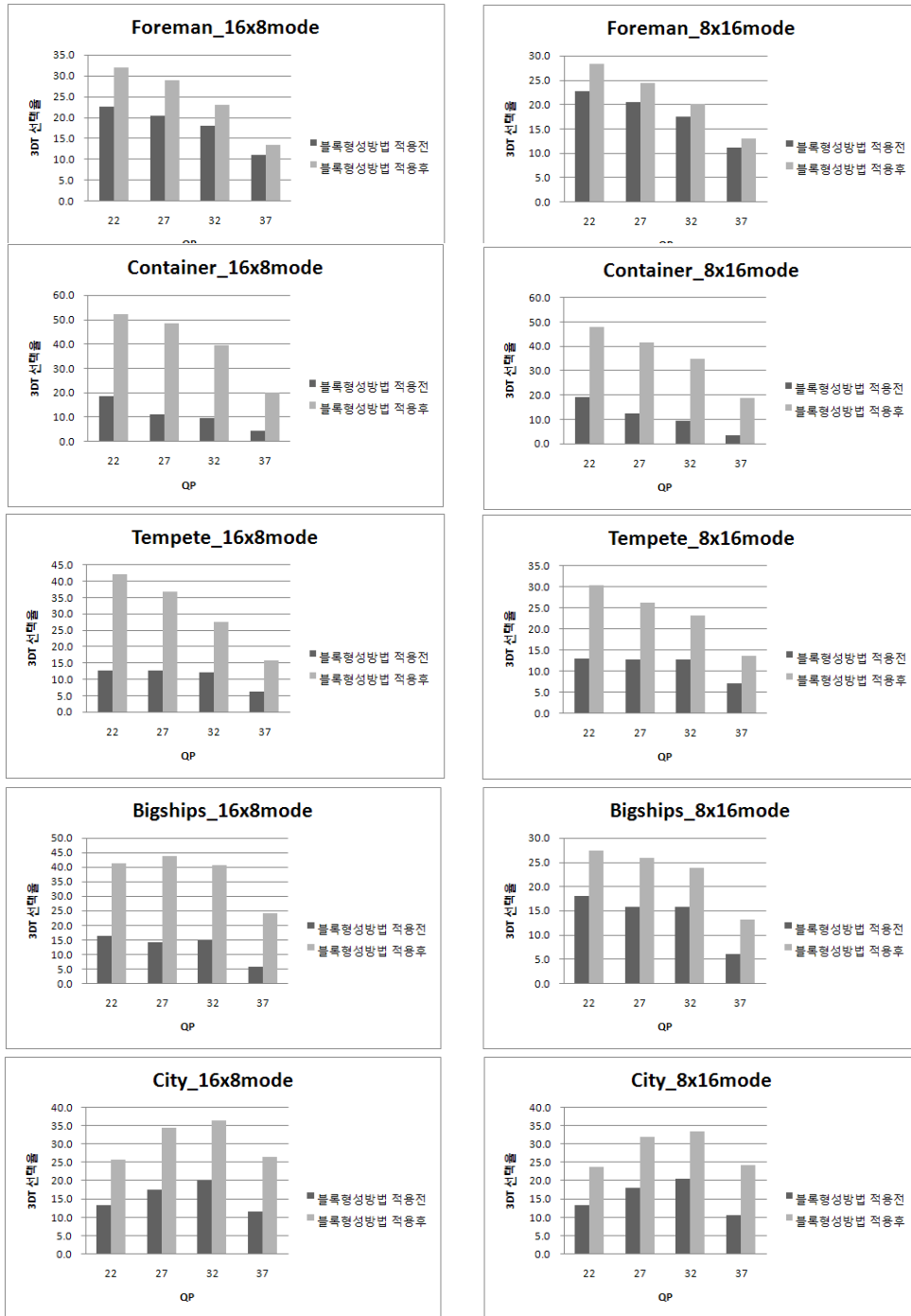


그림 4. 모드에 따른 블록형성방법을 적용하기 전과 후, 16x8, 8x16 모드에서 3DT가 선택되는 비율 비교

Fig. 4. Comparison of selecting ratio for 3DT in 16x8, 8x16 mode when mode dependent block forming method is applied and not applied

그림 4는 16x8와 8x16 모드에서 정방향 8x8 블록으로만 3DT를 한 경우보다 모드의 방향성을 고려하여 블록형성방법을 사용하여 3DT를 하였을 때 16x8, 8x16 모드 내에서 3DT가 선택되는 비율이 높아진다는 것을 보여준다. 이를 통하여 모드 방향성을 고려하지 않은 것보다 고려하여 3DT하는 것이 효율적임을 알 수 있다. 특히 Container 영상이나 Bigships 영상이 다른 영상들보다 블록형성방법을 적용하기 전보다 그 후의 3DT의 선택율이 더 높은 것으로 보아 제안하는 방법이 일정한 방향성의 움직임은 가지는 영상에 대해서 더욱 효율적이라고 말할 수 있다.

앞에서 제안한 방법과 같은 블록모드 방향에 따라 3DT를 수행할 때 CBP(Coded Block Pattern)방식도 이에 맞게 수정되어야 한다. H.264/AVC에서 사용하는 CBP는 부호화기에서 잔차신호를 변환 및 양자화한 후 나온 계수값이 8x8 블록단위로 모두 0인지 아닌지를 복호기에 알려주는 역할을 한다. 본 논문에서는 16x8 모드이고 3DT를 사용하여 부호화하는 경우, 앞서 제안한 블록형성방법에 따라 16x4 블록단위로 변환 및 양자화를 하기 때문에 CBP는 16x4 단위로 부호화한다. 마찬가지로 8x16 모드이고 3DT를 사용하여 부호화하는 경우, CBP는 4x16 단위로 부호화한다. 상기한 내용을 그림 5에 도시하였다.

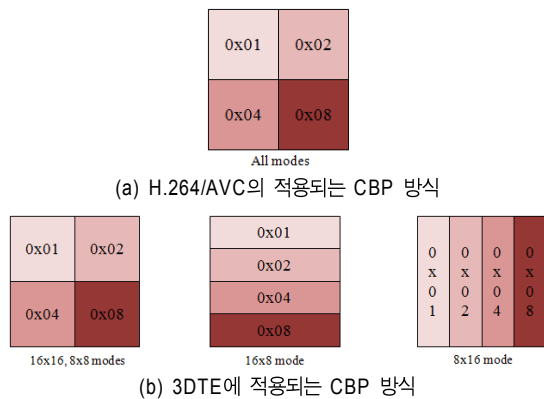


그림 5. 3DTE에서 모드에 따른 블록형성방법에 따른 CBP방식
Fig. 5. CBP by mode dependent block forming applied to 3DTE

2. 3DT를 위한 주사방법

H.264/AVC에서는 엔트로피 부호화를 효율적으로 하기

위하여 4x4 혹은 8x8 블록단위로 지그재그 주사방법을 사용한다. 이는 변환 후 블록 내의 계수간에 발생하는 에너지 몰림현상(energy compactness)을 이용하여 효과적으로 엔트로피 부호화하려는 것이다. 3DT는 네 개의 4x4 블록단위로 두 번의 변환을 하므로 8x8 블록단위의 엔트로피 부호화 방법을 채택한다. 이에 따라 엔트로피 부호화를 할 때 4x4 블록이 아닌 8x8 블록단위의 CABAC을 사용하여 부호화한다.

본 논문에서는 3DTE를 부호화할 때 8x8 블록단위의 엔트로피 부호화를 효율적으로 하기 위한 주사방법을 제안한다. 이는 3DT 및 양자화 후에 주파수 계수값 각각의 에너지를 조사한 내용에 근거하고 그림 6에서 보여준다. 실험은 CIF영상 중 Foreman영상, 50 프레임을 사용하고 IPPP 구조, QP=22 에서 3DT가 최적의 변환으로 선택되었을 때 에너지 분포를 조사한 것이다. 그래프의 가로축은 변환 및 양자화된 계수값인 레벨(level)이고 세로축은 이에 대한 선택율을 나타낸다. 가로축의 레벨값은 3이후의 계수값에서는 거의 선택되지 않기 때문에 c_0^j 를 제외한 나머지 주파수 영역은 0부터 3까지 나타내었다. 그리고 각 레벨 값마다 네 가지 막대그래프로 표현한 것은 네 개의 4x4 블록의 같은 주파수($c_i^0, c_i^1, c_i^2, c_i^3$)를 순서대로 배치하여 같이 나타내기 위함이고 이에 따라 각각의 그래프를 4x4 블록내의 주파수 계수의 위치($c_0^j, c_1^j, \dots, c_{15}^j$)대로 배치한다. 4x4 블록 내 4x4 HT측면에서 본다면 낮은 주파수로 에너지가 몰리고 높은 주파수로 갈수록 에너지를 거의 가지고 있지 않음을 볼 수 있다. 특히 c_0^j, c_1^j, c_2^j 에 에너지가 집중적으로 몰리고 나머지 주파수 영역에서는 에너지가 거의 없다. 4x4 HT를 수행한 후의 4x1 HT 측면에서 본다면 각각의 그래프에서 레벨마다 $c_i^0, c_i^1, c_i^2, c_i^3$ 사이에서 다른 주파수보다 대부분 c_i^0 (dc)에 에너지가 더 많이 집중되는 것을 볼 수 있다. 하지만 4x4 HT로 인해서 이미 c_0^j, c_1^j, c_2^j 에 에너지가 집중되어 있고, 다른 주파수 영역은 이에 비해 상당히 에너지가 적게 분포하기 때문에 이를 고려해주어야 한다. 이는 4x4 HT와 이후에 수행되는 4x1 HT의 에너지의 특성을 동시에 반영해야 한다는 것을 의미한다. 따라서 본 논문에서는 2차원

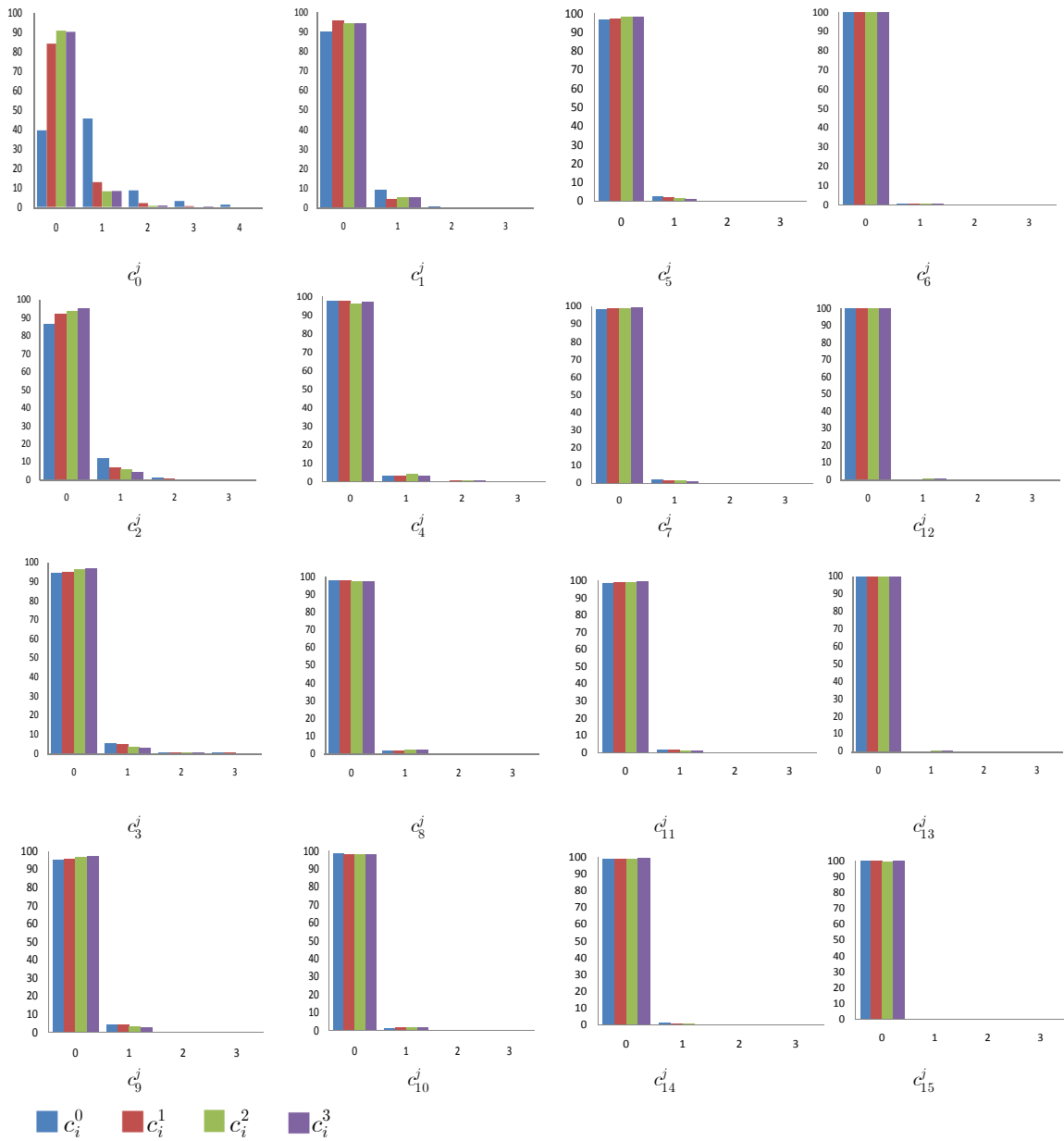


그림 6. 8x8 블록에서의 3DT 및 양자화 후의 계수값 분포
 Fig. 6. After 3DT and quantization, the distribution of coefficients in 8x8 block

변환과 1차원 변환에 의해 나온 주파수 중요도 특성을 동시에 반영할 수 있도록 그림 7과 같이 3DT와 양자화 후에 같은 주파수 위치의 계수를 모아서 각각 순서(0, 1, 2, ..., 15)에 맞게 8x8 블록 내에 배치하고 지그재그 주사방식을

사용하여 엔트로피 부호화한다.

앞에서 3DT를 적용할 때 8x8 블록만을 사용해서 같은 주파수를 모으는 것보다 모드에 따라 세 가지 블록형성방법을 사용한 경우의 3DT 선택율이 높은 것을 보았고 이에

따라 3DT를 적용할 때도 모드의 방향성이 블록 내부의 상관도에 영향을 끼친다는 것을 알 수 있었다. 이는 가로나 세로 방향의 움직임이나 에지가 존재할 때 변환 후 이 방향의 주파수 성분이 크다는 것을 의미한다^[6]. 따라서 16x8, 8x16 모드에 대하여 변환한 경우에 상기한 주파수 계수들의 특성을 고려한 새로운 주사방법을 제안한다. 16x8 모드 일 때 가로방향은 같은 특성을 갖는다고 판단할 수 있고 이에 따라 상대적으로 세로 방향의 주파수 변화도가 크다. 그래서 지그재그 방향이 아닌 세로방향으로 양자화 계수값을 픽업(pick-up)하고 8x8 블록에 그림 7의 (b)와 같이 배치

시켜서 지그재그 방향으로 주사하여 엔트로피 부호화를 한다. 8x16 모드는 이와 반대의 과정을 거쳐 주사하고 그림 7의 (c)에서 이 과정을 보여준다. 아무런 방향성이 없는 16x16, 8x8 모드는 지그재그 방향으로 양자화 계수값을 픽업하고 8x8 블록에 배치시킨다.

3. 3DT에 적용된 양자화 파라미터

H.264/AVC에서 사용하고 있는 양자화 파라미터(f)는 양자화 과정에서 데드존(dead zone)의 범위를 조절해 주는 역

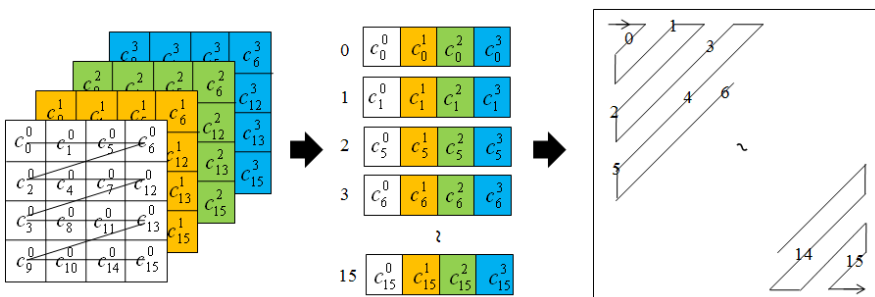
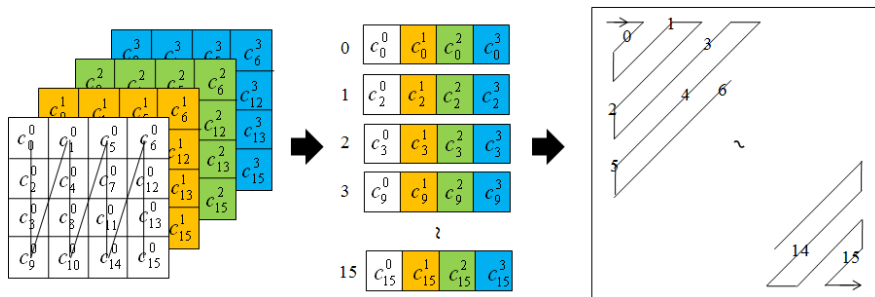
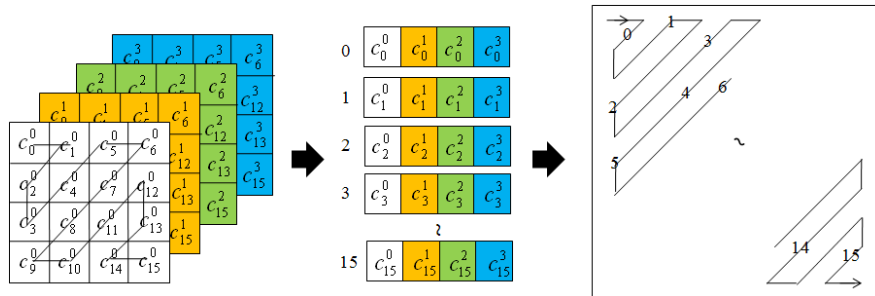
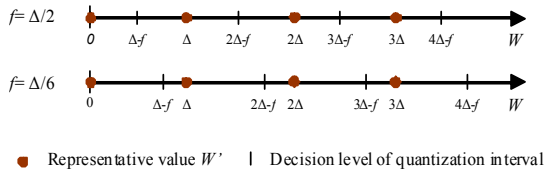


그림 7. 3DTE를 위한 주사 방법
Fig. 7. Scanning method for 3DTE

할을 한다. 데드존은 양자화 했을 때의 계수값(Representative value : W')이 0임을 결정할 수 있는 범위($\Delta - f$)를 말한다. 이를 표현하는 그림 8에서 양자화 파라미터가 클수록 데드존의 범위가 좁아지고 반대로 양자화 파라미터가 작을수록 데드존의 범위가 넓어짐을 볼 수 있다^[7]. 화면간 프레임을 부호화 할 경우 잔차신호를 변환하고 양자화 한 값이 0으로 몰리는 정도는 화면내 프레임을 부호화 할 경우보다 높다. 이때 양자화 파라미터를 이용하여 데드존을 적절하게 조절을 함으로써 부호화 효율을 높인다^[8]. H.264/AVC에서 사용하는 양자화 파라미터는 화면내 프레임 부호화 할 경우 1/3을 사용하고 화면간 프레임을 부호화 할 경우에 1/6을 고정적으로 사용한다.



● Representative value W' | Decision level of quantization interval
 그림 8. 양자화 파라미터가 $\Delta/2$, $\Delta/6$ 일 때의 양자화 과정
 Fig. 8. Visualization of the quantization in H.264/AVC for two different parameters f ($f = \Delta/2$, $f = \Delta/6$)

하지만 3DT는 화면간 프레임을 부호화할 때 사용하는 변환이고 변환을 두 번 하기 때문에 H.264/AVC에서 사용하고 있는 변환과 양자화 특성과는 다르다. 그림 6에서 보았듯이 낮은 주파수 대역에서는 높은 에너지 분포비율을 보이고 높은 주파수 대역에서는 상당히 낮은 에너지 분포비율을 보인다. 즉 낮은 주파수로 갈수록 0이 아닌 레벨값이 높은 주파수 영역보다 많이 분포하고 높은 주파수로 갈수록 레벨값이 거의 다 0이 됨을 볼 수 있다. 그리고 2차원 변환인 4x4 HT와 1차원 변환인 4x1 HT의 특성을 모두 고려해야 한다. 4x4 HT후의 낮은 주파수 영역에서 4x1 HT 했을 때의 dc (c_i^0)는 다른 주파수보다 0이 아닌 레벨값이 많이 존재하고 높은 주파수 영역에서 1차원 변환을 했을 때 가장 마지막 주파수 계수 $ac(c_i^3)$ 값은 거의 다 0이 됨을 알 수 있다. 이 실험 결과에 근거하여 본 논문에서는 4x4 HT와 4x1 HT를 통해서 가장 낮은 주파수 3가지 계수값에 대하여 양자화 파라미터를 1/3, 높은 주파수 9가지 계수값에 대하여 양자화 파라미터를 1/9, 나머지 위치는 1/6 을 사용하여 부호화 효

율을 높인다. 식 (12)는 3DT 및 양자화 후 주파수 위치에 따라 양자화 파라미터를 부여한 것을 나타낸 것이다.

$$f = \begin{cases} 1/3 & \text{for } c_0^0, c_1^0, c_2^0 \\ 1/9 & \text{for } c_4^3, c_7^3, c_8^3, c_{10}^3, c_{11}^3, c_{12}^3, c_{13}^3, c_{14}^3, c_{15}^3 \\ 1/6 & \text{otherwise} \end{cases} \quad (12)$$

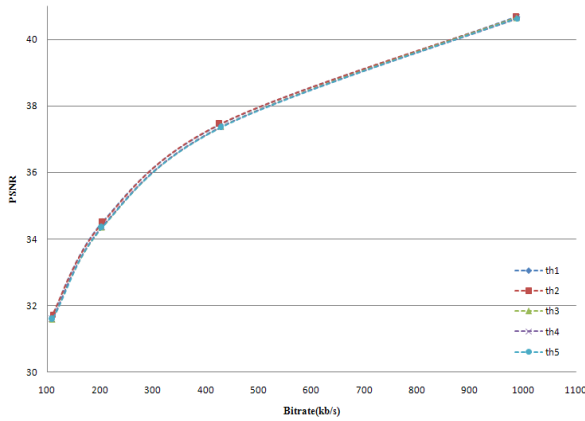
4. 3DT를 위한 8x8 블록단위의 양자화 계수값 가중치와 임계값 제안

H.264/AVC표준의 참조 소프트웨어인 JM^[9]에서는 엔트로피 부호화를 하기 전에 8x8 블록단위로 잔차신호가 변환과 양자화 과정을 거친 뒤 0이 아닌 계수값에 대해서 일정 조건을 만족하면 부호화하지 않고 잔차신호를 보내지 않는 방법을 사용한다. 이는 8x8 블록 내에서 0이 아닌 계수값이 거의 존재하지 않을 때 이를 부호화하는 것이 효율적인지 모든 계수를 0으로 하는 것이 효율적인지를 화질 대비 비트량을 고려하여 성능을 높이기 위함이다. 이를 판단하는 조건식이 식 (13)이다. 이때 i 와 j 는 각각 4x4 블록의 x 와 y 좌표값이고, r 은 제로-런(zero-run), $coeff(i, j)$ 은 레벨, $c(i, j)$ 는 레벨 1값에 곱해지는 가중치이다. 이 식에서 볼 수 있듯이 4x4 블록단위의 제로-런, 레벨 값에 의해서 결정된 가중치를 레벨 값에 곱하여 나온 값을 더한다. 이와 같은 과정을 8x8 블록 내에 네 개의 4x4 블록에 적용하여 나온 결과값들을 더하여 축적한 값이 임계값인 4보다 작거나 같으면 해당 8x8 블록의 계수값은 모두 0으로 부호화한다. 이후 임계값은 th 로 표시한다.

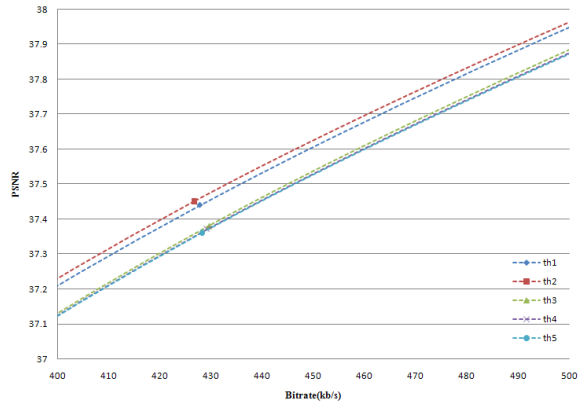
$$c(i, j) = \begin{cases} \infty & \text{if } coeff(i, j) > 1 \\ 3 & \text{if } r = 0 \text{ and } coeff(i, j) = 1 \\ 2 & \text{if } 1 \leq r \leq 2 \text{ and } coeff(i, j) = 1 \\ 1 & \text{if } 3 \leq r \leq 5 \text{ and } coeff(i, j) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$c_{3-D}(k, l) = \begin{cases} \infty & \text{if } coeff_{3-D}(k, l) > 1 \\ 3 & \text{if } 0 \leq r' \leq 3 \text{ and } coeff_{3-D}(k, l) = 1 \\ 2 & \text{if } 4 \leq r' \leq 11 \text{ and } coeff_{3-D}(k, l) = 1 \\ 1 & \text{if } 12 \leq r' \leq 23 \text{ and } coeff_{3-D}(k, l) = 1 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

식 (14)에서 k, l 은 8x8 블록 내에서의 x, y 좌표값이고,



(a) QP=22, 27, 32, 37에서 RD 곡선



(b) Q=27에서 RD 곡선

그림 9. Foreman 영상에서 3DTE에 적용된 여러 임계값(th)에 대한 율-왜곡 곡선
Fig 9. RD curve for variable thresholds(th) applied to 3DTE in Foreman sequence

r' 은 제로-런, $coeff_{3-D}(k,l)$ 는 3DT 및 양자화를 거친 레벨값, $c_{3-D}(k,l)$ 는 적용되는 가중치이다.

3DTE에 적용하는 th은 JM에서 정한 4가 아닌 2를 사용한다. 그림 9는 CIF영상인 Foreman영상, 100프레임, QP=22, 27, 32, 37 대역에서 3DTE에 적용하는 th을 1부터 5까지 나타낸 율-왜곡(Rate-Distortion : RD) 곡선이다. 그림 9의 (a)는 QP=22, 27, 32, 37 대역, 그림 9의 (b)는 QP=27 대역에서 th의 성능을 비교한 것이다. QP 대역별 성능은 비슷하므로 자세하게 나타낸 QP=27 대역을 통하여 자료를 분석하기로 한다. 이 그림을 보면 th=3, 4, 5로 하여 3DTE에 적용했을 때보다 th=1, 2로 적용했을 때 성능 향상이 더 큰 것을 확인할 수 있으며, th=1인 경우보다 th=2인 경우가 3DTE에서 높은 효율을 보이기 때문에 th=2가 3DTE에 가장 적합하다고 판단된다.

IV. 3DT를 위한 양자화기 설계

H.264/AVC에서 사용하는 4x4 HT는 2차원 코어행렬을

사용하여 6비트($\log_2 6^2$)와 잔차신호 값의 범위에 의해 최대 9비트가 소요되고 이를 합치면 최대 15비트의 dynamic range를 가진다. 양자화 과정에서는 $qbits(15+floor(QP/6))$ 를 사용하여 변환과 양자화 계산에 의해 소요된 dynamic range을 낮춰 16비트 연산을 할 수 있도록 한다^[10].

4x4 HT와 4x1 HT를 사용하는 3DT의 dynamic range는 17비트이다. 이는 2차원 코어행렬에 1차원 코어행렬을 곱함으로써 dynamic range는 8비트($\log_2 6^3$)가 되기 때문이다. 그래서 H.264/AVC에서 사용하는 양자화 계산과정에서 소요되는 범위와 맞추기 위하여 3DT에 대한 양자화 과정에서는 $qbits(qbits_{3-D})$ 값을 $17+floor(QP/6)$ 으로 정의한다. 이와 동시에 4x4 HT와 4x1 HT에 의해 생성된 PF_{3-D} 와 PF'_{3-D} 를 설계하고 정수형 계산을 하기 위하여 MF_{3-D} 와 V_{3-D} 를 정의한다. 식 (15)는 3DT에 대한 양자화 과정을 나타내는 계산식이다. 3DT를 위한 양자화 과정은 코어행렬로 변환된 계수값(X_{3-D})과 MF_{3-D} 를 곱하여 정수형 계산을 하고 이를 $qbits_{3-D}$ 만큼 오른쪽시프트(right-shift)시켜 값을 낮춘다.

$$X_q(i, j) = sign(X_{3-D}(i, j)) \times (|X_{3-D}(i, j)| \times MF_{3-D}(Q_M, i, j) + f_{3-D} \times 2^{qbits_{3-D}}) \gg qbits_{3-D} \tag{15}$$

$$PF_{3-D} = PF_{2-D} \otimes PF_{1-D}$$

$$= \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \otimes \begin{bmatrix} a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a \end{bmatrix} \begin{bmatrix} b/2 & b/2 & b/2 & b/2 \\ b/2 & b/2 & b/2 & b/2 \\ b/2 & b/2 & b/2 & b/2 \\ b/2 & b/2 & b/2 & b/2 \end{bmatrix} \quad (16)$$

이때 i, j 는 각각 4x4 블록 내에서 x, y 좌표를 가리킨다. 또한 $Q_M = QP\%6$ 이고 X_{3-D} 는 8x8 블록 내에서 4x4 블록단위로 HT를 수행한 후 네 개의 같은 주파수간의 4x1 HT를 수행한 계수값^[11]을 나타내며, f_{3-D} 는 3DT를 위한 양자화 파라미터를 가리킨다. 앞에서 쓰인 %는 각각 나눗셈 연산을 수행했을 때 나머지를 구하는 연산기호이다.

3DT에 대한 정수형 양자화기를 설계함에 있어서의 핵심은 MF_{3-D} 와 V_{3-D} 를 설계하는 것이다. MF_{3-D} 를 설계하기 위해서는 먼저 PF_{3-D} 를 구해야 한다. PF_{3-D} 는 HT에서 사용하는 $PF(PF_{2-D})$ 와 4x1 블록기반의 1차원 변환에 대한 $PF(PF_{1-D})$ 의 조합으로 구할 수 있다. 식 (16)은 PF_{3-D} 를 구하기 위하여 4x4 블록단위의 PF_{2-D} 와 PF_{1-D} 로 표현한 계산과정을 나타낸 것이다. \otimes 은 행렬의 같은 위치의 요소간의 곱셈을 수행하는 연산이다. 그리고 각각의 행렬요소 a, b 는 각각 $1/2, \sqrt{2/5}$ 로 정의한다. 이들 요소를 표현하기 위하여 $PF_{3-D}(r)$ 을 사용한다. r 을 4x4 블록에서의 x, y 좌표로 표현하는데 $r=0$ 은 $(i, j) = \{(0,0),(0,2),(2,0),(2,2)\}$, $r=1$ 은 $(i, j) = \{(1,1),(1,3),(3,1),(3,3)\}$, $r=2$ 은 그 외의 좌표를 나타낸다.

식 (16)에서 행렬 요소간의 곱을 계산하면, $PF_{3-D}(r)$ 는 두 가지로 나뉜다. 이는

$$PF_{3-D}(r)_{table1} = \{a^3, ab^2/4, a^2b/2\}, \quad (17)$$

$$PF_{3-D}(r)_{table2} = \{a^2b/2, b^3/8, ab^2/4\} \quad (18)$$

으로 정의할 수 있다. 3DT에 의해 도출된 PF_{3-D} 는 MF_{3-D} 를 계산하는 과정에 사용된다. 다음의 식 (19), (20)은 MF_{3-D} 에 대한 계산식이다. PF_{3-D} 는 두 종류가 있기 때문에 MF_{3-D} 또한 두 종류의 계산식이 존재한다. 또한 이 식은 H.264/AVC에서 PF 와 MF 관계를 정의할 때 사용한 $PF/Qstep = MF/2^{qbits}$ 의 관계식을 이용하여 계산된다.

$$MF_{3-D}(Q_M, r)_{table1} = \frac{PF_{3-D}(r)_{table1}}{Qstep} \times 2^{qbits_{3-D}}, \quad (19)$$

$$MF_{3-D}(Q_M, r)_{table2} = \frac{PF_{3-D}(r)_{table2}}{Qstep} \times 2^{qbits_{3-D}} \quad (20)$$

역양자화 과정 역시 양자화 과정에서 PF_{3-D} 를 사용하여 MF_{3-D} 를 정의했듯이 PF'_{3-D} 를 사용하여 V_{3-D} 를 정의하고 이를 양자화된 값(X_q)과 곱하여 $Q_E(\text{floor}(QP/6))$ 만큼 레프트-시프트(left-shift) 시킨다. 다음의 식 (21)은 3DT에 대한 역양자화 과정을 나타내는 계산식이다.

$$X_r(i, j) = X_q(i, j) \times V_{3-D}(Q_M, i, j) \ll Q_E \quad (21)$$

역양자화 과정에 사용되는 V_{3-D} 를 정의하기 위해서 PF'_{3-D} 를 설계해야 한다. 다음의 식 (22)는 PF'_{3-D} 가 PF'_{2-D} 와 PF'_{1-D} 으로 만들어지는 과정을 4x4 블록단위로 나타낸 것이다.

식 (22)에서 행렬 요소간의 곱셈을 통하여 $PF'_{3-D}(r)$ 은

$$PF'_{3-D} = PF'_{2-D} \otimes PF'_{1-D}$$

$$= \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \begin{bmatrix} a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a \end{bmatrix} \begin{bmatrix} b & b & b & b \\ b & b & b & b \\ b & b & b & b \\ b & b & b & b \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \begin{bmatrix} a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a \end{bmatrix} \begin{bmatrix} b & b & b & b \\ b & b & b & b \\ b & b & b & b \\ b & b & b & b \end{bmatrix} \quad (22)$$

식 (23), (24)와 같이 두 종류로 정의된다.

$$PF'_{3-D}(r)_{table1} = \{a^3, ab^2, a^2b\}, \quad (23)$$

$$PF'_{3-D}(r)_{table2} = \{a^2b, b^3, ab^2\} \quad (24)$$

V_{3-D} 은 식 (25)과 (26)에 의해서 계산된다.

$$V_{3-D}(Q_M, r)_{table1} = PF'_{3-D}(r)_{table1} \times Qstep \times 64 \quad (25)$$

$$V_{3-D}(Q_M, r)_{table2} = PF'_{3-D}(r)_{table2} \times Qstep \times 64 \quad (26)$$

이와 같이 계산된 MF_{3-D} 와 V_{3-D} 는 식 (27)을 통해 정확한 값을 정한다. 이렇게 정해진 정수값을 표 3에서 QP와 4x4 블록 내의 위치에 따라 정의한다.

$$MF_{3-D}(Q_M, r) \times V_{3-D}(Q_M, r) \times m(r) \cong 2^{23} \quad (27)$$

이때 $m(r)$ 은 아래식 (28), (29)에 의해 정의된다.

$$m(r)_{table1} = \{4^3, 4 \times 5^2, 4^2 \times 5\}, \quad (28)$$

$$m(r)_{table2} = \{4^2 \times 5, 5^3, 4 \times 5^2\} \quad (29)$$

본 논문에서는 3DT에 대한 MF_{3-D} 와 V_{3-D} 를 정의함으로써 정수형 변환에 알맞은 정수형 양자화기를 설계하였다. 3DT를 위해 설계된 양자화기는 H.264/AVC에서 양자화기를 만든 방법의 과정과 동일하게 하였기 때문에 기존의 표준과 큰 변화 없이 부호화 할 수 있다. 또한 정수형 양자화의 계산은 PF_{3-D} 로 인한 소수점 계산을 피할 수 있다. 이는 부호화

표 3. QP와 4x4 블록 내 위치에 따른 MF_{3-D} 와 V_{3-D} 정의
Table 3. Multiplication factor MF_{3-D} and Rescaling factor V_{3-D}

QP	r=0	r=1	r=2
0	26214	10486	17476
1	26214	9321	14980
2	21845	8389	13107
3	18725	7626	11651
4	16384	6453	10486
5	14564	5992	9532

(a) $MF_{3-D}(QP, r)_{table1}$

QP	r=0	r=1	r=2
0	17476	6711	10486
1	14980	6101	9321
2	13107	5162	8389
3	11651	4793	7626
4	10486	4194	6453
5	9532	3728	5992

(b) $MF_{3-D}(QP, r)_{table2}$

QP	r=0	r=1	r=2
0	5	8	6
1	5	9	7
2	6	10	8
3	7	11	9
4	8	13	10
5	9	14	11

(c) $V_{3-D}(QP, r)_{table1}$

QP	r=0	r=1	r=2
0	6	10	8
1	7	11	9
2	8	13	10
3	9	14	11
4	10	16	13
5	11	18	14

(d) $V_{3-D}(QP, r)_{table2}$

기에서 재생성된(reconstructed) 영상과 복호기에서 복원된(decoded) 영상 사이의 불일치(mismatch)현상을 막을 수 있고 속도측면에 있어서도 정수 계산이기 때문에 효율적이다.

V. 실험결과

제안하는 방법의 부호화 성능 향상을 비교하기 위하여 JM 11.0 소프트웨어를 사용하였고 표 4는 실험조건을 나타낸다.

표 4. 실험 조건
Table 4. Experimental conditions

Profile	High
GOP Structure	I-P-P-P
RD Optimization	ON
Quantization Parameter	22, 27, 32, 37
Entropy coding	CABAC
ME Search Range	16
Transform Size	JM11.0 : 4x4HT, 8x8HT JM11.0+3DTE : 4x4HT, 8x8HT, 3DT
Total Encoded Frames	200

3DTE는 네 개의 4x4 블록을 이용하여 변환과 양자화를 거쳐 8x8 블록단위의 엔트로피 부호화를 해야 한다. 이에 따라 8x8 정수형 변환(8x8 HT)을 선택적으로 사용할 수 있는 High Profile에서 실험했고 엔트로피 부호화는 8x8 블록단위의 CABAC을 사용하였다. 따라서 JM 11.0은 8x8 HT를 선택적으로 사용하였고 3DTE는 2개의 변환에 추가적으로 3DT를 사용하였다. 이에 따라 3DT를 다른 변환과 구별하기 위하여 추가 비트를 사용한다. 본 논문에서 사용하는 방법은 표 5에서 보여주는 것과 같이 4x4, 8x8 HT를 나타내는 변수(luma_transform_size_8x8_flag)와 동일한 변수를 사용하였고 각각의 변환에 다른 번호를 부여하였다.

표 5. 각각의 변환을 나타내는 flags
Table 5. Flags for each transform

luma_transform_size_8x8_flag	변환
0	4x4HT
1	8x8HT
2	3DT

제안하는 방법의 성능향상을 비교하기 위해서 VCEG에서 사용하고 있는 QP대역 22, 27, 32, 37을 사용하여

표 6. 실험 결과
Table 6. Experimental results

Sequence		QP	JM 11.0		JM 11.0 + 3DTE		
			Bitrate(kbps)	PSNR (dB)	Bitrate(kbps)	PSNR (dB)	BD-RATE (%)
CIF (352x288)	Container	22	615.37	40.42	619.93	40.53	-4.36
		27	233.80	36.75	234.18	36.89	
		32	85.90	33.34	86.06	33.54	
		37	37.25	30.37	36.68	30.54	
	Carphone	22	806.20	41.70	805.91	41.81	-3.94
		27	388.44	38.80	389.10	38.97	
		32	184.16	35.56	187.45	35.84	
		37	90.90	32.49	92.48	32.71	
	Foreman	22	1010.14	40.87	1008.19	40.92	-3.48
		27	455.93	37.62	453.08	37.74	
		32	218.99	34.51	220.90	34.75	
		37	118.57	31.72	120.45	31.92	
	Tempete	22	2902.14	39.72	2901.50	39.75	-1.79
		27	1446.36	35.62	1450.43	35.70	
		32	589.25	31.49	598.65	31.66	
		37	232.04	27.94	234.84	28.09	
	Silent	22	539.63	40.30	543.58	40.39	-0.53
		27	260.95	36.64	262.26	36.68	
		32	124.52	33.31	127.02	33.41	
		37	61.89	30.60	63.59	30.74	
Average						-2.82	

WVGA (832x480)	Mobisode2	22	987.05	44.42	991.11	44.48	-3.66
		27	451.59	42.79	452.29	42.89	
		32	221.81	40.62	223.87	40.8	
		37	120.90	38.20	122.51	38.37	
	Flower vase	22	2328.25	43.36	2338.34	43.45	-2.64
		27	998.43	40.13	1008.76	40.28	
		32	417.41	36.57	430.48	36.82	
		37	186.69	33.25	191.49	33.44	
	Mobisode1	22	1501.25	44.04	1503.91	44.11	-2.38
		27	720.56	42.12	722.51	42.21	
		32	360.88	39.78	363.68	39.88	
		37	195.42	37.12	197.01	37.31	
	Keiba	22	6756.83	41.49	6823.32	41.51	-1.44
		27	3420.37	38.78	3419.96	38.81	
		32	1773.96	35.87	1755.78	35.95	
		37	954.21	33.13	963.76	33.23	
Average						-2.53	
720p (1280x720)	City	22	27201.57	40.29	27521.82	40.35	-7.92
		27	7882.15	36.75	7847.64	36.9	
		32	2296.38	33.20	2273.27	33.57	
		37	978.86	29.98	956.07	30.33	
	Raven	22	10236.72	42.82	10343.40	42.96	-7.72
		27	3870.50	40.01	3868.37	40.25	
		32	1619.40	36.85	1627.41	37.26	
		37	812.55	33.84	812.78	34.15	
	Jets	22	5515.51	41.93	5629.36	42.02	-6.89
		27	1274.47	40.03	1275.78	40.21	
		32	559.77	37.66	559.96	37.92	
		37	298.34	35.08	299.70	35.26	
	Bigships	22	17175.21	40.49	17442.81	40.6	-6.10
		27	5690.24	37.12	5784.32	37.3	
		32	1879.64	33.86	1920.56	34.2	
		37	727.52	31.21	728.79	31.44	
	Crew	22	16837.26	41.96	16932.14	41.99	-4.55
		27	5636.87	39.33	5646.32	39.46	
		32	2303.82	36.92	2361.13	37.16	
		37	1111.86	34.59	1140.47	34.83	
	Night	22	29278.09	41.10	29608.68	41.14	-2.54
		27	11223.38	37.71	11295.64	37.8	
		32	4411.90	34.29	4488.44	34.51	
		37	2047.91	31.20	2086.59	31.41	
	Harbour	22	35887.93	40.58	36240.18	40.59	-2.04
		27	16099.33	37.21	16310.80	37.28	
		32	6283.53	33.60	6575.94	33.90	
		37	2608.56	30.23	2688.28	30.57	
Average						-5.40	
Total Average						-3.58	

CIF(352x288), WVGA(832x480), 720p(1280x720) 해상도의 다양한 영상에 대하여 JM 11.0과 제안하는 방법의 성능을 비교하였고 표 6에서 나타낸다. 성능비교는 BD(Bjontegaard Delta)-RATE를 사용하여 측정하였다^[12]. 이때 BD-

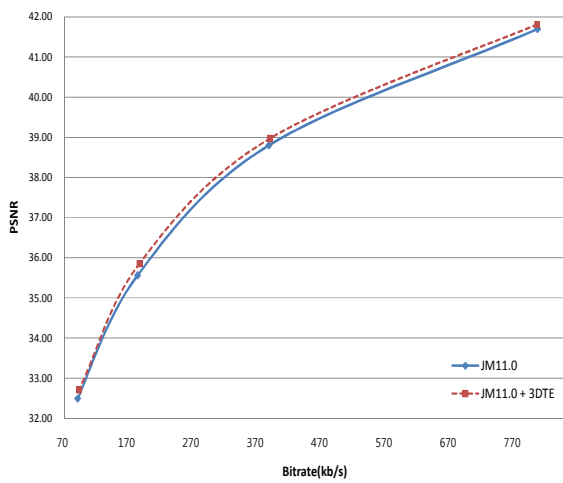
RATE값이 음수로 표시된 것은 JM 11.0의 성능대비 절감된 비트율을 의미한다. 제안하는 방법을 통하여 모든 영상에서 평균 3.58%의 비트율 절감을 얻을 수 있었고 특히 HD(720p) 영상에서 평균 5.40%의 비트율 절감을 보였다.

이는 해상도가 큰 영상에 3DTE가 적합하다고 할 수 있다.

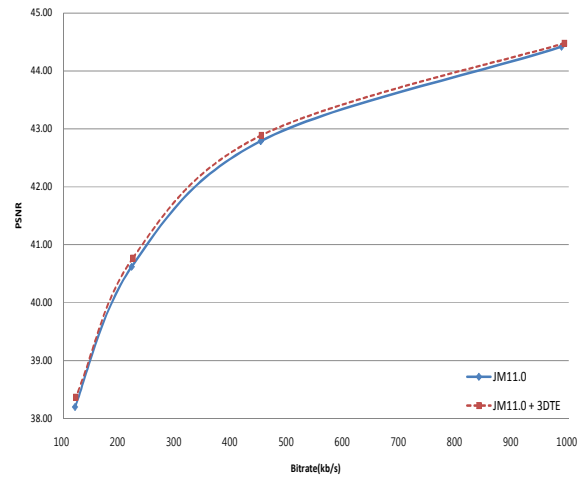
그림 10은 각 해상도 별로 1개의 대표적인 영상에서 JM 11.0과 JM 11.0 + 3DTE에 대한 휘도 성분 율-왜곡 곡선 (R-D curve) 결과이다. 세로축은 왜곡을 통해 영상의 객관적 화질을 나타내는 PSNR이고 가로축은 부호화 된 비트스트림의 비트율을 나타낸다. 그림을 통해서 제안하는 방법은 각각의 여러 해상도뿐만 아니라 다양한 비트율 환경에서도 우수한 부호화 성능향상을 가져오는 것을 확인할 수 있다. 다만 비트율이 높아질수록 JM 11.0 + 3DTE의 비트

가 JM11.0보다 약간 더 높아짐을 확인할 수 있지만 비트 대비 화질의 성능이 우수함을 그림 10의 율-왜곡 곡선에서 확인할 수 있다.

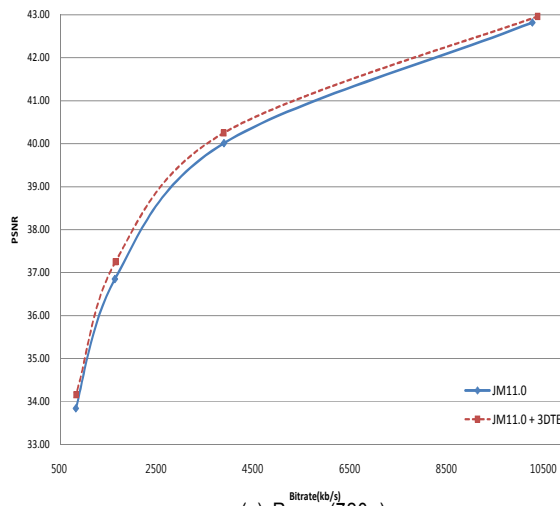
표 6과 그림 10을 정리해 볼 때 전반적으로 성능은 비트 보다는 화질(PSNR)측면에서 높은 이득을 본다는 것을 알 수 있다. 또한 다양한 비트율 환경에서 JM 11.0보다 JM 11.0 + 3DTE의 성능이 우수함을 확인할 수 있다. 영상별로 성능을 비교해 볼 때 일정한 방향으로 움직이거나 화면이 동의 방향이 일정한 영상, 그리고 비교적 적은 움직임은 보



(a) Carphone(CIF)



(b) Mobisode2(WVGA)



(c) Raven(720p)

그림 10. 율-왜곡 곡선
Fig. 10. RD curve

이거나 화면이동이 적은 영상에서 JM 11.0 + 3DTE의 성능이 높음을 볼 수 있다. 하지만 이와 반대로 복잡한 움직임을 가지고 비트가 많이 소모되는 영상에서 성능향상 정도가 앞의 영상들보다 낮다. 종합해볼 때 3DTE는 복잡하거나 빠른 움직임의 영상이 아닌 비교적 적은 움직임을 가지거나 일정한 방향성 있는 영상에서 높은 효율을 보인다고 할 수 있다.

VI. 결 론

기본적으로 4x4 HT를 사용하는 H.264/AVC는 높은 예측 성능을 유지하고 변환하는 반면에 작은 블록크기단위로 변환하기 때문에 공간적 중복도를 줄이는 효율은 낮다. 본 논문에서는 H.264/AVC가 가지고 있는 예측성능을 유지하면서 공간적 중복도를 추가로 줄이는 3DT와 이를 기반으로 효율적으로 부호화할 수 있는 3DTE를 제안하였다. 3DT는 H.264/AVC에서 사용하는 정수형 변환을 4x4 블록 단위로 적용하고 인접 네 개의 4x4 블록의 같은 주파수 위치의 계수 값을 모아서 추가로 4x1 블록 단위로 1차원 변환하는 것이었다. 이에 적합한 부호화 환경을 만들기 위하여 모드를 고려한 세 가지 블록형성방법과 이에 맞게 변형한 CBP, 3DT 에너지 통계를 고려한 주사방식 및 양자화 파라미터 수정, 효율적인 엔트로피 부호화를 위한 임계값 변형, 그리고 정수형 3DT용 양자화기를 설계하였다. 4x4 혹은 8x8 HT의 특성에 맞춰져 있는 H.264/AVC에서 제공하는 도구만으로는 3DT 특성을 살려서 부호화 효율을 향상시킬 수 없기 때문에 이와 같은 도구들로 구성된 환경을 제공한 것이다.

제안한 방법에 대한 성능을 검증하기 위하여 잘 알려진 시험 비디오 시퀀스를 사용하여 JM11.0 High 프로파일 환경에서 제안한 방법과 비교하였다. 제안한 3DTE는 동일한

화질을 유지하면서 JM11.0보다 평균 3.58% 정도의 비트율을 절감시킬 수 있었다. 그리고 특히 HD 영상에서 평균 5.40% 정도의 높은 비트율 절감을 얻음으로써 향후 UHD급 해상도를 부호화하는 차세대 비디오 부호화 방법에 적합할 것으로 기대된다.

참 고 문 헌

- [1] ISO/IEC 14496-10 and ITU-T Rec. H.264, Advanced Video Coding, 2003.
- [2] T. Wiegand, Gary J. Sullivan, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. on Circuit and Systems for Video Tec, vol 13, no 7, pp.560-576, July 2003.
- [3] Mathias and Wien, "Variable Block-Size Transforms for H.264/AVC", IEEE Trans. on Circuit and Systems for Video Tec, vol 13, no 7, pp.560-576, July 2003.
- [4] Iain E. G. Richardson, H.264 and MPEG-4 Video Compression, John Wiley & Sons Ltd, England, 2003.
- [5] J. Vaisey and A. Gersho, "Image compression with variable blocks size segmentation", IEEE Trans. On Signal Processing. vol 40, issue 8, pp.2040-2060 Aug 1992.
- [6] D.Y. Kim, S.Y. Jeong, J.W. Hong, G.H. Park, Y.L. Lee, "Enhanced intra coding", JVT-AH11, Antalya, Turkey, January, 2008.
- [7] Thomas Wedi and Steffen Witmann, "Rate-distortion constrained estimation of quantization offsets", JVT-O066, Busan, KR, April, 2005.
- [8] A.M. Tourapou, J. Boyce, "Quantization Offset Matrices for Fidelity Range Extensions", JVT-L032r4, Redmond Meeting, WA, USA, July 2004.
- [9] JVT Model reference software form ftp://ftp3.itu.ch/jvt-site/reference_software.
- [10] Henrique S. Malvar, Fellow, IEEE, "Low complexity Transform and Quantization in H.264/AVC", IEEE Trans. on Circuit and Systems for Video Tec, vol 13, no 7, July 2003.
- [11] 전수열 외 7인, "다차원 DCT를 이용한 비디오 부호화기 설계", 한국방송공학회 논문지, 제 13권, 제 5호, 732-743쪽, 2008년 9월.
- [12] J. Vaisey G. Bjontegaard, "Calculation of Average PSNR Differences between RD curves", ITU-T SG16/Q6, 13th VCEG Meeting, Doc, VCEG-M33, Austin, Texas, USA, April 2001.

저 자 소 개



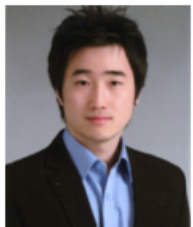
조 재 규

- 2009년 : 광운대학교 전자공학과 학사
- 2009년 ~ 현재 : 광운대학교 전자공학과 석사과정
- 주관심분야 : H.264/AVC, 차세대 비디오 부호화, 영상 및 신호처리



조 혜 정

- 2005년 2월 : 한양여자대학 컴퓨터공학부 졸업(학사)
- 2005년 9월 ~ 현재 : 광운대학교 전자공학과 석박통합과정
- 주관심분야 : 멀티미디어 콘텐츠 보호, 영상처리, 비디오 코덱 시스템, IPTV



이 진 호

- 2007년 2월 : 고려대학교 전자 및 정보공학부 졸업(학사)
- 2009년 2월 : 과학기술연합대학원대학교(UST) 디지털방송공학과 졸업(석사)
- 2009년 3월 ~ 현재 : 한국전자통신연구원(ETRI) 방통융합미디어연구부 연구원
- 주관심분야 : Video coding, HEVC, UHD TV, 3DTV



정 세 운

- 1995년 2월 : 인하대학교 전자공학과 학사
- 1997년 2월 : 인하대학교 전자공학과 석사
- 2006년 3월 ~ 현재 : KAIST 전기및전자공학과 박사과정 중
- 1996년 12월 ~ 현재 : 한국전자통신연구원(ETRI) 방통융합미디어연구부 선임연구원
- 주관심분야 : 비디오 코딩, HEVC, UHD TV, 3DTV



안 창 범

- 1981년 : 서울대학교 전자공학과 학사
- 1983년 : 한국과학기술원 전기및전자공학과 석사
- 1986년 : 한국과학기술원 전기및전자공학과 박사
- 1987년~1991년 : 미국 University of California, Irvine 조교수
- 1992년~현재 : 광운대학교 전기공학과 교수
- Sylvia Sorkin Greenfield Award, 장영실상, 메디슨 의공학상, 광운대 Best Teacher상, 게르베 학술상 등 수상
- 주관심분야 : 영상처리, 영상압축, 의학영상시스템, 다차원 생체신호처리 시스템

저 자 소 개



오 승 준

- 1980년 2월 : 서울대학교 전자공학과 졸업(학사)
- 1982년 2월 : 서울대학교 전자공학과 대학원 졸업(석사)
- 1988년 5월 : 미국 Syracuse University 전기/컴퓨터공학과 졸업(박사)
- 1982년 3월 ~ 1992년 8월 : 한국전자통신연구원 근무(멀티미디어연구실 실장)
- 1986년 7월 ~ 1986년 8월 : NSF Supercomputer Center 초청 학생연구원
- 1987년 5월 ~ 1988년 5월 : Northeast Parallel Architecture Center 학생연구원
- 1992년 3월 ~ 1992년 8월 : 충남대학교 컴퓨터공학부 겸임교수
- 1992년 9월 ~ 현재 : 광운대학교 전자공학부 교수 (VIA-Multimedia Center 멀티미디어연구실)
- 2002년 3월 ~ 현재 : SC29-Korea 의장 및 MPEG Forum 부의장
- 주관심분야 : 비디오 데이터 처리, 비디오 코덱 시스템, 비디오 및 영상압축, 멀티미디어시스템