

H.264/AVC에서 다차원 변환 기반의 동영상 압축 방법

*조재규, *이여송, **정세윤, **이진호, *오승준
**한국전자통신연구원, 광운대학교 *방통융합 플랫폼 센터
*jaeku220@media.kw.ac.kr

MDT-based Compression Scheme in H.264/AVC

*J.K. Jo, *Y.S. Lee, **S.Y. Jeong, **J.H. LEE, *S.J. Oh
**ETRI, *BnC Convergence Platform Center, Kwangwoon University

요 약 문

본 논문에서는 현재 비디오 코덱 표준인 H.264/AVC 에서 사용하고 있는 정수형 변환을 두 번 사용하는 다차원 변환 즉, MDT(Multi-Dimensional Transform) 을 제안한다. 제안하는 알고리즘은 H.264/AVC 7가지 인터 모드중에서 16x16, 8x16, 16x8블록 모드에 적용된다. H.264/AVC에서 사용하는 정수형 변환을 4x4블록 단위로 적용하고 인접하는 4개의 4x4블록으로부터 같은 주파수 위치의 계수값을 모아서 16개의 4x1벡터 변환을 추가로 수행하는 방법이다. MDT를 수행함으로써 H.264/AVC 표준에서 사용되는 변환이 가지는 예측오차를 유지하면서 공간적 중복도를 추가로 줄일 수 있다. MDT 된 계수들을 정수형으로 양자화하고 양자화된 계수값을 효율적으로 주사하는 방법을 제안하여 압축효율을 높였다. 제안된 알고리즘의 제안하는 MDT 기반 방법은 H.264/AVC High profile 표준보다 평균 3.00%의 비트 절감을 보였다.

1. 서론

최근에 표준으로 제안된 H.264/AVC가 기존 방법보다 높은 성능을 가지는 가장 큰 이유는 블록 크기를 4x4까지 줄여서 예측 성능을 높여서 잔차신호(residual)를 작게 하여 변환하기 때문이다.

일반적으로 변환 블록의 크기에 따라 시공간적 예측을 정확하게 하여 잔차신호에 대한 값을 줄일 수 있지만, 이산 변환을 통한 공간적 중복성 제거 효율은 감소한다. H.264/AVC 표준에서는 변환 블록의 크기를 4x4까지 줄여서 처리하기 때문에 기존의 비디오 표준방법과 비교하여 상대적으로 공간적 중복성 제거 효과가 감소하였다. 나중에 이를 보완하기 위해 H.264/AVC에서도 FRExt 프로파일을 제정하여 8x8 변환을 선택적으로 사용하도록 하였다.[1][3]

그러나 본 연구에서는 H.264/AVC 가 갖는 예측성능을 유지하면서 공간적 중복성을 추가로 제거하기 위하여 다차원 변환 방법 (MDT : Multi-Dimensional Transform)을 제안한다.

MDT는 H.264/AVC에서 쓰고 있는 4x4블록 기반의 2차원 정수형 변환 후에 변환된 블록들을 추가변환을 수행한다. 그리고 잔차신호 값이 불규칙적인 인트라프p임에는 기존 변환 방법을 그대로 적용하고 인터 프레임에만 적용하는 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안된 방법인 인터 프레임 MDT에 대해 기술하고, 3장에서는 실험결과에 대해서 설명하며, 4장에서 결론을 맺는다.

2. MDT

MDT는 H.264/AVC와 같은 효율을 가지는 4x4블록 기반의 2차원 정수형 변환을 하고 추가로 같은 주파수의 계수값에 대해서 변환을 한다. 인트라 프레임 부호화는 화면 내에 공간적 중복성을 줄이지만 주변 픽셀값들로 예측을 해서 잔차신호를 만들기 때문에 움직임 예측 (Motion Estimation)을 통하여 만든 인터 프레임 부호화보다는 공간적 중복성이 적게 남아 있

다. 본 논문에서는 인터 예측 모드에서 16x16, 16x8, 8x16, 8x8 모드에 3차원 MDT를 적용한다. MDT

가. 3차원 MDT

3차원 MDT는 4x4블록에 대해 2차원 변환을 하고 8x8블록 단위로 4개의 4x4블록의 같은 주파수 계수값 4개를 묶어서 4x1블록기반의 1차원 정수형 변환을 한다. 즉 2차원 변환 한번과 1차원 변환 한번을 한다. 이를 그림 1에 나타내었다.

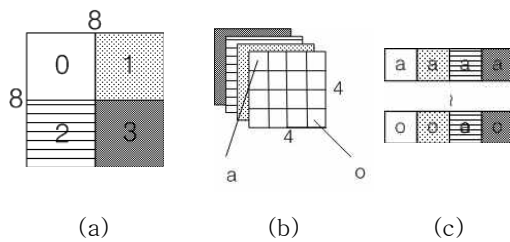


그림 1. 3차원 MDT

- (a) 4x4블록단위의 2차원 변환
- (b) 2차원 변환후의 4개의 같은 주파수 계수값
- (c) 4개의 같은 주파수 값에 대한 1차원 변환

그림1과 같은 방식은 기본적으로 2차원 변환된 4x4블록 4개를 정방향으로 모아서 8x8블록 단위로 형성하고 3차원 MDT를 수행하지만 특정 방향에 대해서 같은 특성을 가지면 그 특성에 맞게 블록 단위로 형성해서 3차원 MDT를 적용한다. 16x16모드와 같은 경우는 특별한 방향성이 없을 경우 최적의 모드로 선택된다. 그러므로 8x8블록의 정방향으로 3차원 MDT를 수행한다. 16x8모드의 경우는 가로 방향으로 같은 특징이 있지만 세로 방향으로는 다른 특징을 가질 때 최적의 모드로 선택되게 되므로 가로 방향으로 4x4블록 4개를 모아서 3차원 MDT를 수행한다. 반대로 8x16모드는 세로 방향으로 같은 특징이 있지만 가로 방향으로 다른 특성을 가질 때 최적의 모드로 선택되므로 정방향인 8x8블록단위로 3차원 MDT를 한다. 이를 그림 2에서 표현한다.[2]

H.264/AVC에서는 변환된 값에 대하여 항상

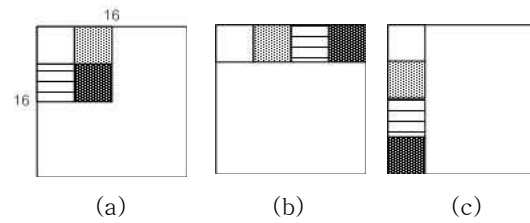


그림 2. 모드에 따른 블록 형성 방법

- (a) 8x8블록 형성 방법
- (b) 16x4블록 형성 방법
- (c) 4x16블록 형성 방법

8x8블록 단위로 양자화된 계수값 0이 없는지 또는 하나 이상 있는지 CBP (Coded Block Pattern)를 통해 표현한다. 그러나 3차원 MDT에서는 모드에 따라 방향성을 고려해서 2차원 변환을 한 4x4블록 4개를 모아 양자화를 하게 되어 이에 따라 CBP를 표현한다. 16x16, 8x8모드는 H.264/AVC와 같이 8x8블록 단위로 CBP를 나타내고 16x8모드는 16x4블록 단위로, 8x16모드는 4x16블록 단위로 CBP를 나타낸다. 한편 8x8블록 혹은 16x16블록 안에 하나의 변환되고 양자화된 nonzero 계수값이 존재할 가능성이 많아진다. 따라서 H.264/AVC에서는 현재 부호화된 레벨(level) 값이 1인 경우 즉, level 1에만 앞의 level 값이 0 즉, level 0인 run 값에 따라서 level 1에 적절한 가중치를 준다. 8x8블록 단위에서 가중치된 값들을 다 더했을 때 임계값 4보다 작거나 같으면 잔차신호를 보내지 않는다. 그러나 MDT를 적용하여 얻은 값은 기존 변환 값과 다른 성격을 가지기 때문에 실험적으로 임계값을 수정하였다. 수정한 임계값은 2이다. 아래에 나와 있는 식 (1)은 H.264/AVC 8x8변환에서 적용하고 3차원 MDT에도 적용하는 가중치이다.

$$c(i, j) = \begin{cases} \infty & \text{if } coeff(i, j) > 1 \\ 3 & \text{if } 0 \leq r \leq 3 \text{ and } coeff(i, j) = 1 \\ 2 & \text{if } 4 \leq r \leq 12 \text{ and } coeff(i, j) = 1 \\ 1 & \text{if } 13 \leq r \leq 24 \text{ and } coeff(i, j) = 1 \end{cases} \quad (1)$$

j : position in 8x8 block

r : run when level is 0 in 8x8 block before level is 1

i : 8x8 block index

$coeff(i, j)$: transform coefficient when after quantization

나. 양자화 설계

H.264/AVC 에서 4x4변환을 쓸 때 2차원 'core' 행렬을 씌으로써 6 비트($\log_2(6)^2$)와 잔차 신호 값의 범위에 의해 9비트를 쓸 수 있다. 그래서 2차원 변환을 함으로써 최대 15비트(dynamic range)를 쓸 수 있고 양자화 과정에서 qbits로 15를 써서 이 값의 범위를 낮추어준다. 이는 양자화 결과값이 16비트값이 되도록 하기 위한 것이다.[4]

3차원 MDT 는 기존의 2차원 'core' 행렬에 1차원 'core' 행렬을 곱한다. 이는 dynamic range 에 영향을 준다. 그래서 3차원 MDT 에 대한 양자화를 설계하기 위해서는 qbits값을 수정한다. 아래의 수식 (2)는 2차원 변환과 1차원 변환을 했을 때 정수형 양자화 수식을 나타낸다.

$$X_q(i, j) = \text{sign}\{X_{MDT}(i, j)\} \cdot \left[(|X_{MDT}(i, j)| \cdot MF(Q_M, i, j) + f \cdot 2^{(17+Q_E)}) \gg (17+Q_E) \right] \quad (2)$$

where,

$$\begin{aligned} Q_M &= Q \% 6, \quad Q_E = Q / 6 \\ X_{MDT}(i, j) &= \mathbf{H}(\mathbf{H}\mathbf{X}\mathbf{H}^T) \\ MF_{MDT}(QP, i, j) &: \text{scaling matrix for PF in MDT} \\ f &: \text{quantization offset} \end{aligned}$$

3차원 MDT 의 양자화는 2차원 변환의 'core' 변환에 추가적으로 1차원 변환을 함으로써 8비트($\log_2((6)^2 \times 4)$)를 쓸 수 있다. 잔차신호의 9비트에 의해 3차원 MDT의 'core'변환으로 최대 17비트가 표현된다. 4x4변환에서의 양자화와 마찬가지로 양자화된 값을 16비트 값으로 표현하기 위해 qbits를 17로 정한다.

또한 정수형 변환을 사용하므로 2차원변환에 추가적으로 1차원변환을 하면 2차원 scaling 행렬 PF 값이 바뀌게 되고 이로 인해 MF 값을 3차원 MDT 에 적합하게 설계할 수 있다. 식 (3)에서는 PF 에 대한 계산식을 보여준다. 3차원 MDT 에 대한 PF 값은 각각 2차원 변환 PF 값과 같은 자리위치에서 1차원 변환 PF 값을 곱한다. 구해진 PF 값을 이용하여 MF 값을 구할 수 있다. 이에 대한 수식은 식(4)에서 표현한다. table 이 2가지로 나뉘어진 것은 1차원 변환의 PF 값으로 2가지 값(a, b/2)이 있다. 이에 따라 table 2가지로 고정해서 MF 값을 사용할 수 있다. 그래서 1차원 변환의 PF 값 중에서 a값이

$$PF_{MDT} = \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix} \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}$$

$$\otimes \begin{bmatrix} a & a & a & a \\ a & a & a & a \\ a & a & a & a \\ a & a & a & a \end{bmatrix} \begin{bmatrix} b/2 & b/2 & b/2 & b/2 \\ b/2 & b/2 & b/2 & b/2 \\ b/2 & b/2 & b/2 & b/2 \\ b/2 & b/2 & b/2 & b/2 \end{bmatrix} \quad (3)$$

where,
a = 1/2
b = $\sqrt{2/5}$

곱해진 자리는 $MF(QP, r)_{MDT_table1}$ 이라고 고정하여 사용하고 b/2 자리는 $MF(QP, r)_{MDT_table2}$ 으로 고정하여 사용한다.

$$\begin{aligned} MF(QP, r)_{MDT_table1} &= \frac{PF(r)_{MDT_table1}}{Qstep} \cdot 2^{17+Q_E} \\ MF(QP, r)_{MDT_table2} &= \frac{PF(r)_{MDT_table2}}{Qstep} \cdot 2^{17+Q_E} \end{aligned} \quad (4)$$

where,

$$\begin{aligned} r=0 & \text{ for } (i, j) = \{(0,0), (0,2), (2,0), (2,2)\} \\ r=1 & \text{ for } (i, j) = \{(1,1), (1,3), (3,1), (3,3)\} \\ r=2 & \text{ for otherwise} \\ i, j &: \text{position in } 4 \times 4 \text{ block} \end{aligned}$$

이를 각각 표1, 2로 나타낸다.

표 1. $MF(QP, r)_{MDT_table1}$

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

표 2. $MF(QP, r)_{MDT_table2}$

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

rescaling 행렬도 이와 반대로 구하면 되므로 생략한다.

다. 모드에 따른 스캐닝 방식

3차원 MDT 를 적용하는 모드 중 16x8모드와 8x16 모드는 방향성을 갖는 모드이다. 만약 16x8모드가 최적의 모드로 선택되었다면 가로방향으로 같은 특성을 갖는다고 판단할 수 있고 8x16모드가 최적의 모드로 선택되었다면 세로방향으로 같은 특성을 갖는다고 판단할 수 있다.[2] 이 특성을 스캐닝 방법에 적용한다. 16x8 모드의 스캐닝 적용 방법을 예로 들면 아래 그림 3과 같다. 16x8모드는 모드 방향성을 고려하여 16x4 블록 단위로 코딩한다. 모드 방향성을

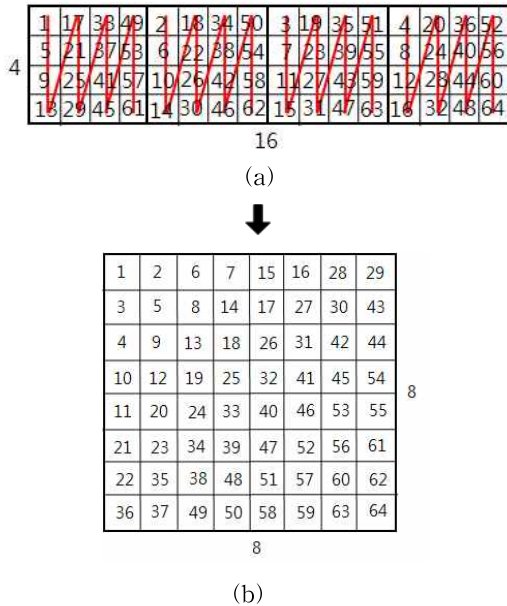


그림 3. 모드에 따른 블록 형성 방법

- (a) 16x4블록 단위에서 양자화 후의 계수값 스캐닝 방법
- (b) (a)방법에서 스캐닝한 후 8x8블록으로 계수값을 가져오는 순서

스캐닝 방법에도 반영하여 가로방향은 같은 특성을 가지므로 가로방향 주파수 성분 에너지 값은 상대적으로 작아진다. 그래서 세로방향의 주파수 성분을 먼저 스캐닝 하는 방법을 썼다. 이를 8x8블록에 그림3의 (b)에 썬진 숫자 순서대로 맵핑시켜서 8x8 엔트로피 코딩을 한다. 8x16 모드 또한 4x16블록 형태에서 이와 같은

과정이므로 생략한다.

3. 실험 결과

본 논문에서는 JM 11.0을 기반으로 High 프로파일인 FRExt 에서 실험을 하였다. 픽처 구조는 IPPP 구조이고, QP는 22, 27, 32, 37대역, 프레임 수는 200장이다. 엔트로피 부호화는 CABAC 에 대해 실험하였다. 제안된 방법은 2차원 변환, 3차원 MDT 를 사용한 것과 H.264/AVC 에서 2차원 변환과 8x8 변환을 사용했을 때의 성능을 비교하여 표 3으로 정리하였다.

표 3. BDBR 성능

Entropy	Size	Sequence	BDBR(%)
CABAC	CIF	Carphone	-3.50
		Foreman	-2.89
		Container	-2.06
		Tempete	-1.24
		Silent	0.92
	Average		-1.75
	720p	City	-6.95
		Bigships	-4.56
		Crew	-3.91
		Night	-1.58
Average		-4.25	
Total		-3.00	

BDBR : Bjontegaard Delta Bit Rate

4. 결론

본 논문은 인터 프레임에서 H.264/AVC 의 4x4변환이 갖는 예측성능을 그대로 유지하면서 공간적 중복성을 추가로 제거하기 위한 방법으로 다차원 변환과 이에 맞는 양자화 설계를 하였고 모드의 특성을 반영한 스캐닝 방식을 택하여 제안된 방법을 적용하였다. 3차원 MDT는 CABAC을 사용했을 때 평균 -3.00% 의 비트절감을 보였다. 특히 영상 크기가 커질수록 더 높은 비트절감율을 보였다.

감사의 글

본 연구는 방송통신위원회, 지식경제부 및 한국산업기술평가관리원의 IT원천기술개발사업의 일환으로 수행한 연구[과제관리번호:2008-F-011, 과제명:차세대DTV핵심기술 개발]와 지식경제부, 한국산업기술진흥원의 전략기술인력양성사업으

로 수행된 결과임

참고 논문

- [1] T.Wiegand, Gary J. Sullivan, and A.Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. on CSVT, vol. 13, no.7, July 2003
- [2] 전수열외 10인, "다차원 DCT를 이용한 비디오 코덱 설계", IPIU 2008, February 2008
- [3] Irin E. G. Richardson, *H.264 and MPEG-4 Video Compression*, John Willey & Sons Ltd, England, 2003
- [4] A. Hallapuro, M. Karczewicz and H. Malvar, " Low Complexity Transform and Quantization - Part 1: Basic Implementation," JVT-B038, Geneva, February 2002