

일반논문-07-12-5-13

DMB를 위한 요약영상 추출 알고리즘

권용광^{a)†}, 원치선^{a)}

A Thumbnail Extraction Algorithm for DMB

Yong Kwang Kwon^{a)†} and Chee Sun Won^{a)}

요 약

DMB에서 사용되는 H.264/AVC (이하 H.264)는 새로운 기술들의 적용으로 높은 압축율과 화질을 제공하는 가장 앞선 동영상 압축 표준이다. 그러나 그 기술들로 인해 기존의 압축 표준에서 적용되던 알고리즘들이 H.264에 그대로 적용되지 않은 문제를 가지고 있다. 그 중 한가지로 압축 비트열에서 영상의 축소와 확대에 대한 새로운 연구가 필요하며, 본 연구에서는 그 문제의 해결 방법으로 요약영상을 추출하는 방법을 제안한다. 제안된 방법들은 기존의 H.264 복호과정을 최대한 이용하면서도 연산량을 50%와 70%를 줄이면서 1/16 크기의 요약영상을 추출함을 보인다.

Abstract

H.264/AVC for DMB is the most advanced video compression standard which provides a high compression rate and a video quality by adopting new technologies. However, these new technologies prevent us from applying some conventional algorithms directly to the H.264/AVC compression domain. For example, we need to study new image resizing schemes in the compressed bit-stream. So, in this paper, we propose a new thumbnail extraction method which can be applied to H.264. The proposed method shows that it can extract a 1/16 size thumbnail video, saving the operation time up to 50% to 70%.

Keywords : H.264/AVC, Thumbnail Extraction, Image Compression

1. 서 론

최근 DTV와 DMB와 같은 디지털 환경의 방송 서비스가 활성화 되고 일반인들도 UCC와 같이 손쉽게 영상 콘텐츠를 제작할 수 있는 방법이 대중화되면서, 동영상 압축과 함께 다양한 응용들이 현실화되고 있다. 특히 방송의 디지털화로 기대되는 다양한 부가 서비스와 사용자 편의성을 위해서는 동영상 요약 기술이 필요하다. 이를 위해 최근 수년간

MPEG-2 기반의 다양한 요약영상 (Thumbnail) 추출 방법들이 개발되었다. 요약영상이란, 원영상의 특징을 대부분 포함하고 있는 축소영상을 의미하며, MPEG-2에서는 연산량을 줄이기 위해 주로 DC 영상 추출 방법이 사용된다. 제안되었던 방법들은 주로 DCT 수식을 변형하거나^[1], 룩업 테이블을 이용^[2]하여 요약영상을 추출하였고, 이렇게 생성된 요약영상은 동영상 검색과 장면 전환 검색, PIP (Picture in Picture), 동영상 요약, 트랜스코딩 등에 사용되었다. 그러나 최근 주목받고 있는 H.264/AVC^[3]에서는 기존의 방법들을 직접적으로 이용한 요약영상 추출은 매우 어렵다.

가장 최근에 표준화된 동영상 압축 표준으로 DMB, DTV, 네트워크 영상 전송, 디지털 미디어 저장 등 다양한 응용 분

a) 동국대학교 전자공학과

Dept. of Electronics Engineering, Dongguk Univ.

† 교신저자 : 권용광(fifthave1@gmail.com)

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학IT연구센터 지원사업의 연구결과로 수행되었음 (ITA-2006-(C1090-0603-0011))

야에 성공적으로 접목되어 기존 압축표준을 밀어내고 새로운 표준으로 자리 잡고 있는 H.264는 픽셀 기반의 화면 내 예측부호화 기법을 포함한 다양한 새로운 알고리즘 들을 도입하여 연산량은 증가한 반면 압축효율이 2배 이상 좋아졌다. 그러나 픽셀 기반 압축 방법의 도입으로 기존 응용 알고리즘을 H.264 환경에서 접목하는데 많은 어려움을 안게 되어 새로운 방법들이 제안되었다^{[4][5]}.

요약영상의 추출에서도 MPEG-2에서는 DCT 계수가 영상의 특징을 완전하게 포함하지만 H.264에서는 영상의 특징이 DCT 공간과 픽셀 공간의 압축으로 분산되어 DCT 만으로는 요약영상을 추출하는 것이 매우 어려우며, 또한 에러로 인한 영상 왜곡과 전파가 기존에 비해 매우 심각하다. 따라서 본 연구에서는 이러한 문제를 해결하여 압축스트림에서 연산량을 최소화한 요약영상 추출 알고리즘을 제안한다.

논문에서 제안하는 H.264 동영상 압축스트림에서의 요약영상 추출 알고리즘은 특히 DMB에서의 다양한 응용이 기대된다. 최근 보편화 되고 있는 DMB 장비에서 고려되어야 할 문제 중 하나가 전력 소모 문제와 속도이다. 따라서 일반적인 DMB 시청 장비는 디블록킹 필터를 제거하는 등의 방법으로 연산량을 줄이고 있다. 그러나 요약영상이 필요한 경우 모든 복호 과정을 수행하고 있기 때문에 전력낭비가 발생한다. 특히 휴대용 시청 장비의 경우 요약영상을 요구하는 여러 응용 분야에서 최소한의 연산으로 요약영상의 추출이 가능한 방법이 제시될 필요가 있다. 차후에는 H.264 표준으로 DTV 방송을 하는 국가에서 작은 화면을 갖는 휴대장비를 이용하여 영상으로 시청하는 응용도 기대된다.

본 논문의 구성은 다음과 같다. II장은 H.264에서 연구와 관련된 특징을 설명하고, III에서 연구의 필요성을 증명한다. IV장에서 요약영상 추출 방법을 설명하고, V장에 인간 시각적인 측면과 PSNR로 성능을 평가한다. 마지막으로 VI장에서 본 논문에 대한 결론을 맺는다.

II. H.264/AVC

1. H.264/AVC Overview

ITU-T와 ISO/IEC에 의해 표준화된 H.264 또는 MPEG-4

part 10 AVC (Advanced Video Coding)는 H.263+나 MPEG-4와 같은 기존의 동영상 표준에 유사한 기능을 제공하면서도 더 높은 압축효율과 높은 전송 신뢰성을 제공하는 것을 목적으로 한다. 이를 위해 H.264는 화면 내 예측부호화 (Intra Prediction)를 통한 공간적 예측, 화면 간 예측부호화 (Inter Prediction)를 통한 움직임 보상(Motion Compensation), 4x4 정수형 변환 (Integer Transform), CAVLC, CABAC 등을 채택하였고, 그림 1의 구조를 갖는다. 실험 결과^{[6][7]}, H.264가 기존에 비해 (H.263, MPEG-2/4) 50%이상 높은 압축율을 제공한다.

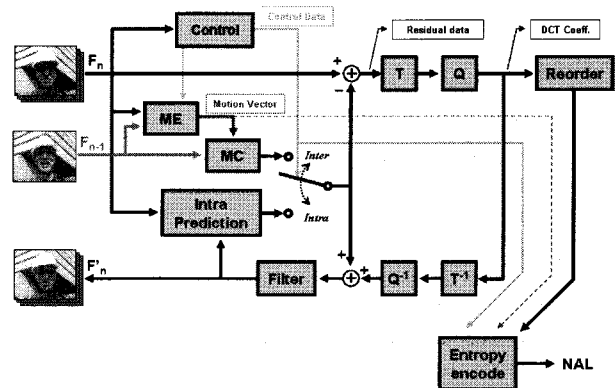


그림 1 H.264 부호화기
Fig. 1 H.264 Encoder

2. 예측모드

H.264는 화면 간 예측부호화와 화면 내 예측부호화를 이용한다.

화면 내 예측 부호화는 MPEG-2에서의 움직임 추출과 유사하며, 가변 블록 크기 움직임 보상(4x4, 4x8, 8x4, 8x8, 8x16, 16x8, 16x16 블록)과 1/4화소 정밀도의 움직임을 제공한다.

본 연구에서 관심을 두는, 화면 내 예측부호화는 한 프레임 내에서 이웃 블록의 픽셀을 참조하여 화소 상관도가 높은 블록을 생성함으로써 압축율을 향상시키는 방법이다. 휘도에서는 그림 2, 3의 9가지 4x4 블록과 4가지 16x16 블록 생성 방법이 제공되며, 각각의 계산된 블록과 원영상과의 SAE (Sum of Absolute Error)가 가장 작은 블록의 모드

가 선택된다^[8]. 색차에는 그림 4의 4가지 모드가 제공된다.

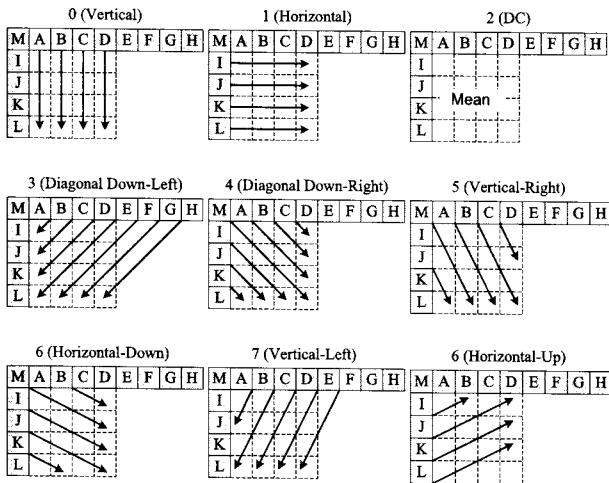


그림 2. 휘도성분 4x4 인트라 예측 모드
Fig. 2. Luma 4x4 intra prediction mode

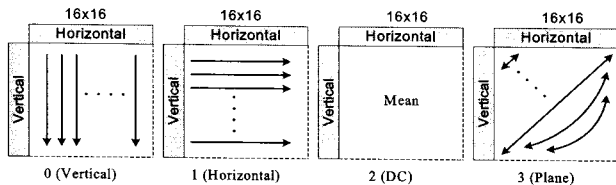


그림 3 휘도성분 16x16 인트라 예측모드
Fig. 3. Luma 16x16 intra prediction mode

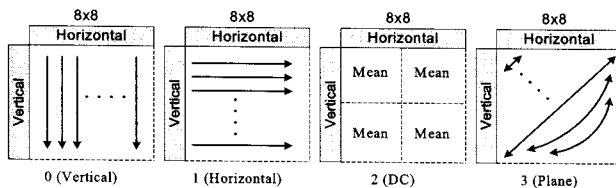


그림 4. 색차성분 8x8 인트라 예측모드
Fig. 4. Luma 8x8 chroma prediction mode

3. 변환과 양자화 과정

DCT는 통계적으로 최적 KLT (Karhunen- Loeve Transform)에 근사되므로 MPEG와 JPEG의 블록 변환에 널리 사용된다^[9].

H.264에서는 블록의 최적 예측 모드와 원 영상 간의 오차 데이터 (잔여 영상, residual image) X를 변환하여 변환

출력 Y를 얻기 위해, 식 (1)의 4x4 DCT를 기반으로 한 정수형 DCT를 사용한다. 이때, $a = 1/2$, $b = \cos(\pi/8)/\sqrt{2}$, $c = \cos(3\pi/8)/\sqrt{2}$ 이다.

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

실제 부호화기 설계에서는 곱셈 연산을 제거하기 위해 위 식을 식 (2)와 같이 변형하여 적용된다.

$$Y = (CXC^T) \otimes E = \left\{ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} X \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \right\} \otimes E \quad (2)$$

이때 E는 후-스케일링 행렬 (post-scaling matrix)이고 표 1의 PF로 정의된다. PF는 다음에 설명되는 양자화 계수와 결합되어 MF (multiplication by a factor)이다. 결과적으로 부호화기는 덧셈과 쉬프트 연산만으로 변환 과정을 수행하게 하고, MF와의 곱셈으로 양자화 과정을 거친다^[10].

표 1. 위치 성분
Table 1. Position Factor

Position	PF
(0,0), (2,0), (0,2) or (2,2)	a^2
(1,1), (1,3), (3,1) or (3,3)	$b^2/4$
other	$ab/2$

H.264에서는 잔여 영상을 16x16 매크로블록(이하 MB) 단위로 4x4 DCT 변환과 하다마드 변환, 양자화를 수행한다. 그림 5에서 각 서브블록(이하 SB)은 16개의 픽셀(4x4)로 구성되며, 숫자(0-25)는 변환 순서를 의미한다. 16번과 17번은 각각 색차 성분의 DC 성분의 2x2 블록으로 2x2 하다마드 변환을 수행한다. 특히 16x16 예측 모드에서는 그림 5의 왼쪽 상단과 같이 휘도 성분의 DC 값으로 4x4 휘도 DC 블록을 만들어 4x4 하다마드 변환을 먼저 실행하고 나머지 변환을 수행한다.

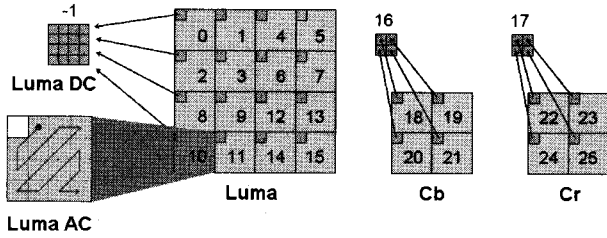


그림 5. H.264 변환 후 재배열 순서
Fig. 5 Reordering after H.264 transform

H.264는 각 MB마다 52개의 QP (Quantization Parameters) 중 하나를 선택하며 선택된 QP에 의해 Qstep이 결정되어 양자화를 수행한다[10]. 이때 QP와 Qstep의 관계를 식(3)과 같이 정의할 수 있다.

$$Q_{step}(QP+6) \cong 2 \times Q_{step}(QP) \quad (3)$$

예측 과정, 변환, 양자화를 차례로 수행하면 그림 1의 재배열기(Reorder)에서 그림 5의 Luma AC에 보이는 순서로 MB의 픽셀들을 1차원 배열한다. 재배열된 데이터 열은 마지막으로 엔트로피 코딩이 되어 저장된다.

4. H.264 디코딩

그림 6은 H.264 복호화기의 블록도이다. 부호화기와 반대로 역 엔트로피 코딩을 수행하고, 재배열, 역변환, 역양자화를 수행하여 인트라 예측된 MB에 더해지면 원래 영상이

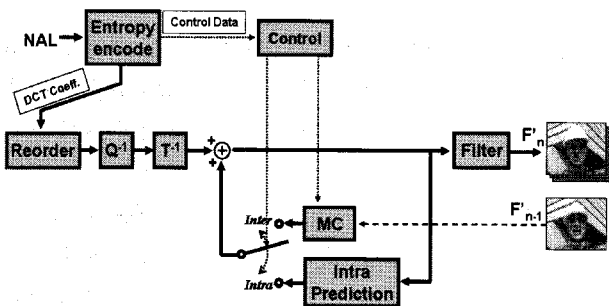


그림 6. H.264 디코더
Fig. 6 H.264 Decoder

복원된다. 식 4는 역양자화 및 역변환 식이며, 이때 Ef는 가중치 성분이다.

$$X = C^T(Y \otimes E_f)C \quad (4)$$

III. 기존 요약영상 추출 방법들과 문제점

가장 우수한 요약영상을 얻는 방법은 영상을 완전하게 복호하고 이를 축소하는 방법이다. 그러나 실시간 방송 및 컴퓨터 응용을 위하여 빠르고 간편한 방법의 요구가 발생하였고, 변형된 DCT를 이용한 영상의 확대와 축소에 대한 연구[11]가 진행되면서 여러 요약영상 기법이 제안되었다. 이 중 가장 간단한 방법은 NxN 단위 블록의 평균 밝기를 의미하는 DC 계수를 구하여 요약영상을 구성하는 방법과 변형된 IDCT를 이용하여 연산량을 최소화한 방법[2]이 있다.

그러나 H.264에서는 다음의 두 가지 이유에서 기존의 방법이 사용될 수 없다. 첫째, 2.3절에서 변환 및 양자화 되는 값은 예측 영상과 원영상과의 오차값(-128~127)으로 변형된 IDCT로는 요약영상을 구성할 수 없다. 그림 7에서 왼쪽 그림은 역양자화와 역변환한 결과를 +128로 레벨 이동하여 구성한 그림으로, 영상의 기존의 방법으로는 요약영상의 구성이 불가능함을 알 수 있다. 두 번째, H.264에서는 2.2절의 인트라 예측모드의 사용으로, 복호화 과정에서 참조 픽셀 (그림 2의 A~X)에서 발생된 에러가 현재 영상 블록의 값에 큰 변화를 주게 되며, 슬라이스의 마지막으로 가



그림 7 H.264의 잔여 데이터
Fig. 7 Residual data in H.264

면서 이러한 에러는 누적되어 화질의 열화가 높아진다. 즉, 4x4 인트라 예측의 경우 그림 2의 13개의 참조블록을 모두 획득해야 영상 블록을 복호할 수 있으며, 복호되는 블록의 오른쪽과 아래쪽 픽셀 7개를 정확하게 복호해야 추가적인 에러전파를 막을 수 있다.

IV. 제안 알고리즘

1. 개요

그림 6의 H.264 복호화기에서는 필터가 전체 연산량의 50%를 차지하기 때문에 상용 시스템에서는 필터를 구현하지 않는 경우가 많다. 그리고 나머지 연산은 대부분 역양자화, 역변환, 예측블록 복원이 차지하고 있다. 따라서 본 논문에서는 전체 과정 중 디블록킹 필터를 제외하고 H.264의 예측모드를 기반으로 하여 기존의 절반이하의 연산량으로 요약영상을 추출하기 위한 두 가지 알고리즘 TNext model 1과 TNext model 2를 제안한다.

2. TNext model 1

H.264에서는 앞 장에서 설명한 바와 같은 문제로 에러전파를 막기 위해 전체 블록 중에서 그림 9의 1번과 2번 픽셀을 정확하게 복호하는 것이 가장 중요하다. 실제 요약영상에는 4x4 블록 단위로 가장 마지막 픽셀이 사용된다. 그림 8은 제안된 TNext model 1 (이하 Model 1)의 내부 블록도이며, 그림 9는 제안 시스템의 출력 위치를 도식으로 나타낸다.

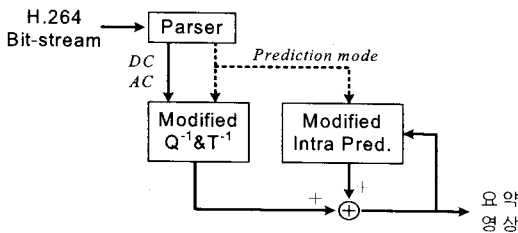


그림 8. TNext 시스템 블록도
Fig. 8. Diagram of TNext system

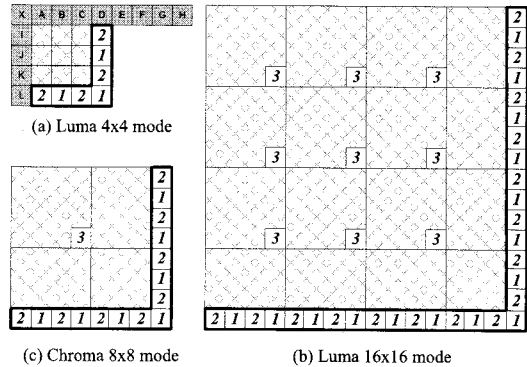


그림 9. TNext 시스템에서 복호되는 픽셀
Fig. 9. Decoded pixel by TNext system

Model 1은 그림 9의 각 블록에서 중요도가 높은 1번과 2번 픽셀 모두를 복호하되 나머지 픽셀의 복호는 생략해도 약 절반의 계산량으로 요약영상을 생성하는 것을 목적으로 한다 (계산량은 다음 절에서 비교). 단, 16x16 휘도성분 예측모드와 8x8 색차 성분 예측모드의 3번 픽셀이 복호되지 않아 영상 구성에 문제가 있다. 그러나 3번 픽셀은 이웃 블록의 복호에 영향을 주지 않으므로 계산량과 구조적 복잡성을 줄이기 위해 해당 블록에서 복호된 픽셀의 대표값을 단순 대입하여 해결하였고, 실험의 PSNR 비교에서 문제가 없음을 확인하였다.

그림 9의 NxN 블록에서 픽셀 좌표의 행과 열을 0~(N-1)의 값을 갖는 j와 i로 각각 정의하면, 한 블록에서 요약영상을 위한 복호된 픽셀 X'_{ji} 을 구하는 과정을 수식으로 표현하면 다음과 같다.

$$X'_{ji} = \begin{cases} (T_{ji} + P_{ji} + 32)/2^5, & \text{if } i = N-1 \\ & \text{or } j = N-1 \\ \text{No Value} & , \text{otherwise} \end{cases} \quad (5)$$

위 식에서, T_{ji} 와 P_{ji} 는 블록의 (i, j) 위치에서의 각각 역양자화 및 역변환의 결과값과 예측 모드에 의해 계산된 결과를 25로 업-스케일링한 값이며, (j=N-1 or i=N-1)인 경우에만 연산이 이루어진다.

3. TNext model 2

요약영상은 영상의 내용을 빠르게 확인하는 것이 목적이

므로 필요에 따라 영상의 품질 저하를 감수하면서 계산량을 크게 줄일 수 있는 알고리즘의 제안도 필요하다. 따라서 TNext model 2 (이하 Model 2)는 영상의 선명도에서 손실을 보는 대신 계산량을 70%이상 줄이는 것을 목적으로 한다. 이를 위해 Model 2에서는 그림 9의 1번 픽셀만을 복호하고 2번은 복호과정에서 생략한다.

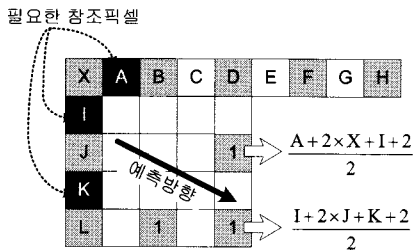


그림 10. TNext model 2에서 4x4 블록의 복호
Fig. 10. 4x4 block decoding by TNext model 2

자세한 설명을 위해 예측모드가 Horizontal-Down인 경우에 Model 2가 SB를 복호하는 모습을 그림 10에 도식하였다. 그림에서 시스템은 이웃 블록으로부터 (X, B, D, F, H, J, L)만을 참조픽셀로 가져올 수 있다. 제안된 방법은 예측모드를 수행하기 위해 이웃 블록에서 (B, J, L, X)를 가져오고, 추가로 필요한 참조픽셀 (A, I, K)는 이웃 참조픽셀의 중간값을 취하여 1번 픽셀의 예측값 P_{13} 를 다음과 같이 계산한다.

$$\begin{aligned}
 P_{13} &= (A + 2X + I + 2) / 2 \times 2^5 \\
 P_{31} &= (J + 2K + L + 2) / 2 \times 2^5 \\
 P_{33} &= (I + 2J + K + 2) / 2 \times 2^5
 \end{aligned}
 \tag{6}$$

그리고 그 결과를 그림 8에서와 같이 역양자화 및 역변환된 출력값(그림 10의 1번 픽셀의 위치만을 계산)과 더하여 SB를 생성한다. 그 이후 과정은 model 1과 동일하다.

3. 연산량 비교

표 2과 3는 JM과 Model 1, Model 2의 연산량을 비교한 내용이다.

표 2의 4x4는 한 SB에서 수행되는 역양자화, 역변환 그

표 2. 역변환 및 역양자화 연산량 비교

Table 2. compare of a operating time on the re-quantization and inverse transform [회/MB]

Block Type	JM12.2		TNext Model 1		TNext Model 2	
	Op.					
+	4x4	1928	4x4	1200	4x4	728
	16x16	1992	16x16	645	16x16	613
×	4x4	388	4x4	352	4x4	352
	16x16	404	16x16	208	16x16	208
shift	4x4	772	4x4	424	4x4	266
	16x16	788	16x16	220	16x16	188

* 4x4 : (Luma 4x4 intra prediction mode X 16)
16x16 : Luma 16x16 intra prediction mode

표 3. 예측블록 연산량 비교

Table 3. compare of a operating time for the prediction block

Luma 4x4		[회/4x4 블록]					
mode	JM12.2		TNext Model 1		TNext Model 2		
	+	shift	+	shift	+	shift	
Vertical	16	16	7	7	3	3	
Horizontal	16	16	7	7	3	3	
DC	24	17	15	8	7	4	
Diagonal Down-Left	37	29	19	16	11	9	
Diagonal Down-Right	37	30	28	21	16	13	
Vertical-Right	42	28	26	19	15	12	
Horizontal-Down	42	28	26	19	15	12	
Vertical-Left	41	26	23	17	15	12	
Horizontal-Up	31	22	13	11	7	6	

Luma 16x16		[회/16x16 블록]								
mode	JM12.2			TNext Model 1			TNext Model 2			
	+	*	shift	+	*	shift	+	*	shift	
Vertical	256	-	256	31	-	31	15	-	15	
Horizontal	256	-	256	31	-	31	15	-	15	
DC	287	-	257	62	-	62	30	-	30	
Plane	1569	784	527	188	81	64	98	41	32	

Chroma 8x8		[회/8x8 블록]							
mode	JM12.2			TNext Model 1			TNext Model 2		
	+	*	shift	+	*	shift	+	*	shift
Vertical	64	-	64	15	-	15	7	-	7
Horizontal	64	-	64	15	-	15	7	-	7
DC	90	-	69	-	-	20	23	-	11
Plane	629	140	133	237	42	35	173	26	19

리고 일반화의 연산량에 SB의 개수 16을 곱한 값이다. 16x16은 4x4 연산량에 그림 5의 회도 DC를 위한 4x4 하다 마드 변환을 포함하였다. 표에서 16x16에서의 연산량 감소가 큰 이유는 그림 9(b)의 SB 내에서 1-2번 픽셀이 아래쪽이나 오른쪽에만 위치한 경우에는 해당 위치에 대한 역양

자화만이 수행되었고 특히, 1-2번이 포함되지 않은 SB는 복호 과정에서 생략되었기 때문이다. 또한 Model 2에서는 추가적으로 역변환에서 큰 계산 감소가 이루어졌다.

표 3의 예측블록의 연산량 비교는 4x4, 16x16, 8x8 모드 단위로 각 예측모드의 계산 회수에 식(5)에서 역변환과 더하기 위한 업-스케일링과 덧셈을 포함한다. 표에서 Luma 4x4의 경우 model 1은 평균 44%, model 2는 평균 69%의 계산량 감소가 발생하였고, 특히 표 1에서와 같은 이유로 그림 9(b), (c)의 Luma 16x16과 Chroma 8x8에서도 큰 계산량 감소가 발생하였다.

V. 실험 결과

제안된 두 방법을 위해 그림 11과 같이 H.264 압축 비트열을 파싱하고 요약영상을 생성하여 컴퓨터에 출력하는 프로그램을 제작하였다. 오른쪽에는 H.264 영상압축 데이터 완전 복호화된 결과를, 왼쪽에는 제안 방법에 따른 요약영상을 출력하며, 전체 복호화 과정에서 제안된 부분을 최대한 비교를 위하여 디블록킹 필터는 제거되었다.

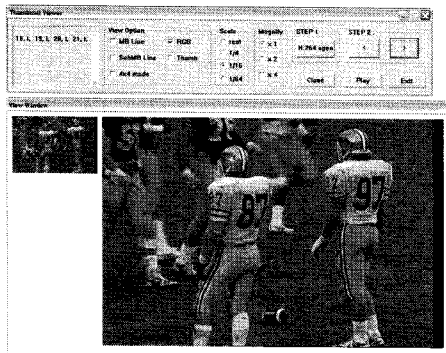


그림 11. TNext 프로그램
Fig. 11. TNext program

실험에는 CIF (Bus, Crew, Foreman) 352x288 영상과 D1 (Football, Garden) 720x480 영상을 JM12.2로 I 부호화하여 사용하였다. 영상은 베이스라인 프로파일(Baseline Profile)에서 영상크기에 따라 CIF 영상은 Level 2.0, D1은 Level 4.0으로 설정하였고, QP만을 다양하게 설정하고 나머지 설

정은 동일하게 하였다. 영상의 평가를 위한 기준 요약영상은 YUV 원영상에서 4x4 블록당 1개의 대표 픽셀을 추출하는 방법으로 생성하였고, JM의 PSNR 평가를 위해 각 QP 별로 264 압축파일을 JM12.2로 복호한 YUV 영상들에서 같은 방법으로 요약영상을 생성하여 비교하였다. 또한 논문 목적에 따라 모든 영상은 I 슬라이스만으로 부호화 하였다. P와 B 프레임에서는 단순히 I 슬라이스에 의한 움직임 보상을 수행하기 때문에, I 슬라이스에서의 요약영상이 정상적으로 추출된다면 P와 B슬라이스의 기존의 MPEG-2에서의 연구결과들을 접목하는 것이 가능하다고 판단하여 위와 같이 실험 데이터를 결정하였다.

그림 12와 13은 QP가 10, 20, 30일 때 Foreman (CIF)과

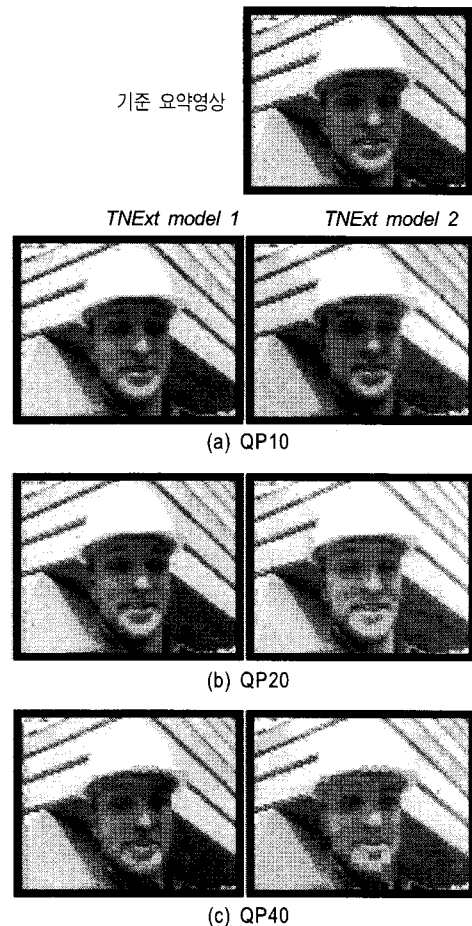


그림 12. Foreman (CIF, 352x288)의 요약영상 비교
Fig. 12. thumbnail compare (Foreman, CIF, 352x288)

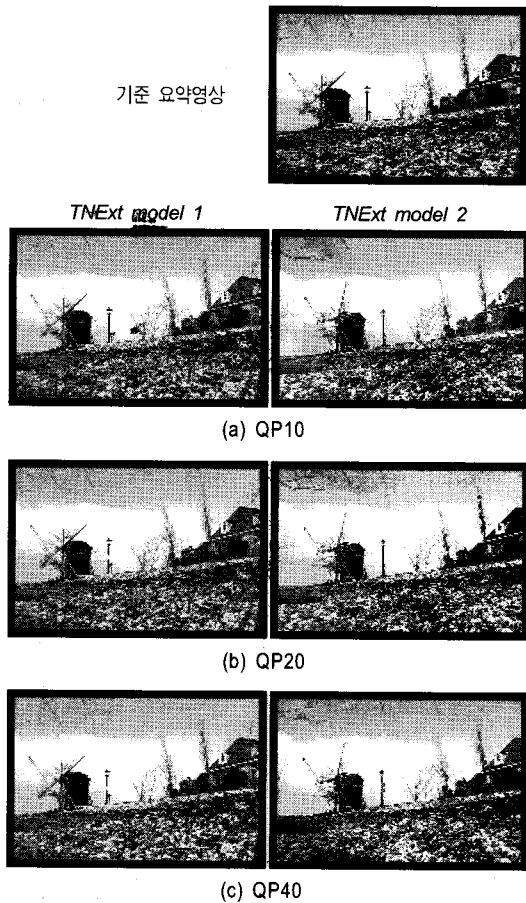


그림 13. Garden (D1, 720x480)의 요약영상비교
Fig. 13. thumbnail compare (Garden, D1, 720x480)

Garden (D1)의 H.264 압축 파일에서 추출된 요약영상을 YUV 원영상에서 추출된 기준 요약영상과 비교한 그림이며 각각 Model 1과 Model 2의 결과이다. Model 1의 경우에는 기준 요약영상과 비교하여 영상의 특징을 대부분 포함하고 있으며, QP가 증가함에도 영상 정보를 확실하게 파악할 수 있다. 반면에 Model 2의 방법을 이용하여 추출된 요약영상은 휘도와 색차 성분의 복호화 과정에서 생략된 픽셀 (그림 9의 2번 픽셀) 로부터 발생하는 화면 내 예측 부호화 손실로 인하여 에러가 발생하였다. 따라서 연속적인 에러 전파로 인해 영상의 전체 또는 약 -40° 방향으로 국지적인 휘도와 색차에 변화가 발생된다. 그러나 전체적으로는 Model 1에서 추출된 요약영상과 비교하여 영상의

특징(형태, 에지 등)을 충분히 포함하고 있다.

그림 14와 15는 JM과 Model 1 및 Model 2의 QP 변화에 따른 PSNR를 보인다. 그림에서 Model 1은 JM에 비해 평균 2dB 낮은 결과를 보이며, 앞에서 고려된 계산량 비교를 고려하면 요약영상 검출에서 좋은 성능을 보인다. 이때 발생한 2dB 감소는 그림 9의 3번 픽셀이 대표값으로 대체되면서 발생하는 오차로 연산량 감소를 고려할 때 2dB의 오차를 감수하는 것이 유리하다고 판단된다.

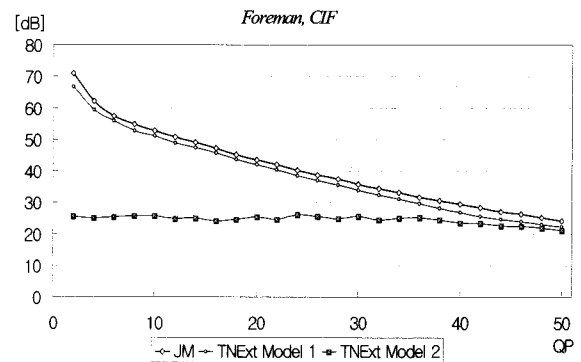


그림 14. Foreman CIF(352x288) 요약영상의 PSNR 비교
Fig. 14. PSNR of the Foreman thumbnail video

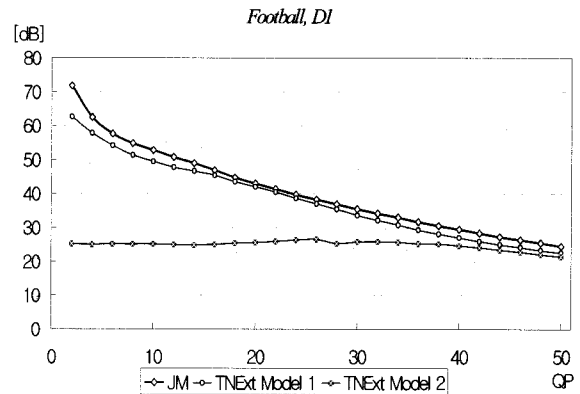


그림 15. Football D1(720x480) 요약영상의 PSNR 비교
Fig. 15. PSNR of the Football thumbnail video

반면 Model 2의 경우에는 각 QP 단계별로 시각적인 판단에서는 우수한 성능을 보이는데 반해 단순 PSNR 비교에서는 낮은 QP에서 나쁜 결과를 보이고 QP가 높아지면서

상대적으로 JM의 성능에 근접함을 보인다. 낮은 QP에서 PSNR이 차이가 나는 이유는 다음과 같다. H.264에서 QP는 양자화 단계값과 비례하는 구조를 갖는다^[12]. 즉 QP가 상승하면 양자화 단계값이 커지고 영상의 화면 내 예측으로 만들어진 블록과 결합될 역변화의 값이 점차 사라지게 된다. 이는 MPEG-2에서 QP가 상승하면 영상이 부드러워지는 것과 유사하다. 따라서 그림 16과 같이 QP가 낮으면 예측된 블록과 결합될 역변환 값이 많아지고 영상이 세밀해짐을 보이고 있다.

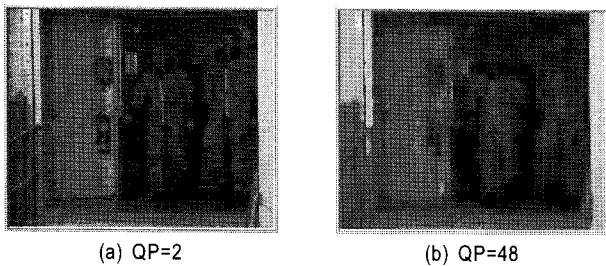


그림 16. Crew 영상의 QP에 JM의 따른 화질 비교
Fig. 16. video quality compare of JM depending on QP

그런데 Model 2에서는 QP가 낮은 경우에 그림 10의 이웃 참조픽셀로부터 추정된 참조픽셀 (A, C, E, G, I, K)의 값이 원래 가져야할 픽셀 값과 차이가 발생할 가능성이 존재하며 이러한 경우 예측블록의 생성에서 오차가 발생하게 된다. 즉 솔트 영상과 같이 픽셀 단위로 매우 텍스처하거나 예측 픽셀 사이에 큰 에지가 생성된 경우 이에 대한 대처가 낮아지게 되어 PSNR 저하가 심하다 그러나 블록의 생성에는 여러 픽셀들이 참조되기 때문에 일정 수준에서 에러 전파는 차단된다. 반면에 QP가 높아질수록 위의 참조픽셀의 오차가 작아지기 때문에 복호 픽셀의 오차가 작아진다. 결과적으로 JM과 비교하여 QP가 20 이하에서는 PSNR이 차이가 크지만, 상업용으로 많이 지정되는 QP30 부근에서부터 PSNR 차이가 근접하여 QP35 이상에서는 그 차이가 매우 줄어들음을 볼 때, JM보다 70% 절약된 계산량을 고려하면 우수한 성능을 보인다고 판단된다.

표 4은 실험 영상들의 QP 20, 30, 40에서 PSNR을 비교한다. 표에서도 QP가 높아질수록 PSNR의 차이가 줄어드

는 것을 확인하였다. 그러나 수치적인 평가와는 별도로, 시각적인 평가에서는 QP에 상관없이 Model 2에서도 내용을 충분히 파악할 수 있는 요약영상을 얻을 수 있었다. 따라서 응용에 따라서 고품질의 요약영상을 요구되는 경우에는 Model 1을 빠른 속도를 요구하는 경우에는 Model 2를 선택하는 것이 유리하다고 판단한다.

표 4. 영상별 PSNR 비교
Table 4. PSNR of several videos

QP	video model	CIF			D1	
		Foreman	Bus	Crew	Football	Garden
20	I	43.55	43.27	43.81	42.99	43.81
	II-1	41.96	41.06	41.95	42.02	43.13
	II-2	25.43	25.29	31.12	25.64	23.45
30	I	35.83	34.31	36.63	35.49	34.74
	II-1	33.95	32.51	33.87	33.68	34.02
	II-2	25.62	24.80	29.81	25.85	25.17
40	I	29.53	26.90	30.92	29.53	26.53
	II-1	26.90	24.43	27.77	27.07	25.59
	II-2	23.67	22.27	26.60	24.70	22.78

* I:JM12.2, II-1:TNext model 1, II-2:TNext model 2

VI. 결론

본 논문에서는 H.264 동영상 파일에서 1/16 크기의 요약영상을 추출하기 위한 두 가지 방법을 제안하였다. 제안된 방법들은 H.264 복호화기에서 픽셀을 선택적으로 복원하여 기존의 알고리즘을 가능한 유지하면 연산량을 줄여서 시간과 전력사용량을 줄이는 것을 목적으로 하였다. 그리고 실험결과 제안 방법으로도 충분히 내용 파악이 가능한 요약영상의 추출이 가능함을 보였으며 평균 50% 이상 연산량을 줄일 수 있었다. 차후 Model 2의 계산량 수준에서 화질을 더욱 높일 수 있는 방법의 연구가 필요하다.

참고 문헌

[1] Jianmin Jiang, Ying Weng, "Video Extraction for Fast Content Access to MPEG Compressed Video", IEEE TCS vol 14 no 5, pp595-605, May 2004
[2] 윤자친, 실상훈, "압축영역에서 빠른 축소 영상 추출을 위한 다중부호

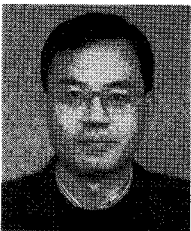
- 록업테이블 설계”, 방송공학회논문지, 제10권 제3호, pp 413-421, 2005
- [3] ISO/IEC 114496-10, "Information technology - Part 10: Advanced Video Coding", 2th edition, Dec. 2005
- [4] Y.T. Hong etc, "High Speed Architecture for MPEG-2/H.264 Video Transcoding", ISCT06, pp. 674-678, Oct. 2006
- [5] Wang TianJiang, Zhang Yi, "A Digital Watermarking Method in H.264 Video Transmission", ICCT06, pp. 1-3, Nov. 2006
- [6] T. Wiegand, G. Sullivan, "'Overview of the H.264/AVC video coding standard,'" IEEE TCS vol 13, pp. 560-576, Jul. 2003.
- [7] S. Saponara, C. Blanch, K. Denolf, and J. Bormans, "'The JVT advanced video coding standard: complexity and performance analysis on a tool-by-tool basis,'" Proc. 13th Int. Packetvideo Workshop, Nantes, France, Apr. 2003
- [8] Iain E.G. Richardson, "H.264 and MPEG-4 Video Compression", John Wiley & Sons, pp. 177-183, 2003
- [9] K. Rao, P. Yip, Discrete Cosine Transform: Algorithms, Advantages, Applications. Boston, MA: Academic, 1990
- [10] A. Hallapuro, M. Karczewicz, "Low Complexity Transform and Quantization", JVT-B038
- [11] E. Dugad, M. Ahuja, "A Fast Scheme for Image Size Change in the Compressed Domain," IEEE Trans, CSVT, vol. 11, no. 4, pp. 461-474, Apr. 2001
- [12] 정제창, "H.264/AVC 비디오 압축 표준", 홍릉출판사, 152-160, 2005

저 자 소 개



권 용 광

- 2002년 2월 : 동국대학교 대학원 전자공학과 (공학석사)
- 1999년 8월 ~ 2002년 2월 : (주)메텔 VRS개발팀
- 2004년 3월 ~ 현재 : 동국대학교 대학원 전자공학과 박사과정
- 주관심분야 : H.264, MPEG-7, MVC



원 처 선

- 1982년 2월 : 고려대학교 전자공학과 (공학사)
- 1986년 : Univ. of Massachusetts 전기 및 컴퓨터공학 (공학석사)
- 1990년 : Univ. of Massachusetts 전기 및 컴퓨터공학 (공학박사)
- 현재 : 동국대학교 전자공학과 정교수
- 주관심분야 : MRF, MPEG-7, Watermarking, H.264