

일반논문-08-13-2-06

## H.264/AVC 비트스트림에서 효율적으로 축소 영상을 추출 하는 방법

유상준<sup>a)†</sup>, 윤명근<sup>a)</sup>, 김은석<sup>b)</sup>, 손채봉<sup>a)</sup>, 심동규<sup>a)</sup>, 오승준<sup>a)</sup>

## An Efficient Thumbnail Extraction Method in H.264/AVC Bitstreams

Sang-Jun Yu<sup>a)†</sup>, Myung-Keun Yoon<sup>a)</sup>, Eun-Seok Kim<sup>b)</sup>, Chae-Bong Sohn<sup>a)</sup>, Dong-Gyu Sim<sup>a)</sup>,  
and Seung-Jun Oh<sup>a)</sup>

## 요약

최근 H.264/AVC 표준을 이용한 HDTV, IPTV와 같은 고화질 디지털 서비스가 증가함에 따라 이를 위한 빠른 동영상 처리 기술들이 요구되고 있다. 특히, 빠른 축소 영상 추출 방법은 영상의 색인과 동영상 요약 분야에서 요구된다. 하지만 H.264/AVC는 이전의 표준과는 달리 화소 공간에서도 예측방법을 사용하기 때문에 종래의 주파수 공간상의 DC 영상 추출 방법을 적용할 수 없었다. 본 논문에서는 Chen이 제안한 방법에 근간을 두고 H.264/AVC 인트라 프레임으로부터 축소영상을 주파수 영역에서 이론적으로 추출하는 방법을 제안한다. 제안하는 방법은 변환 계수에 직접적으로 적용하기 때문에 고속으로 축소 영상을 추출할 수 있다. 제안하는 방법은 인트라 예측을 화소 공간에서 일반화된 식으로 정리하였고, 주파수 공간에서 LUT(Look Up Table)을 이용하여 고속으로 썸네일을 추출한다. 제안하는 방법을 통해 얻은 썸네일과 참조 복호화기를 통해 얻은 썸네일간의 주관적인 화질은 크게 차이가 나지 않으면서도, HD급 영상에서는 평균 63% 빠르게 썸네일을 추출할 수 있다.

## Abstract

Recently, as growing of high definition media services like HDTV and IPTV, fast moving picture manipulation techniques need to meet what those services require. Especially, a fast reduced-size image extracting method is required in the areas of video indexing and video summary. Conventional DC image extracting methods, however, can't be applied to H.264/AVC streams since a spatial domain prediction scheme is adopted in H.264/AVC intra mode. In this paper, we propose a theoretical method for extracting a thumbnail image from an H.264/AVC intra frame in the frequency domain. Furthermore, the proposed scheme can extract the thumbnail very fast since all operations are applied to transform coefficients directly, after a general equation for the thumbnail extraction in nine H.264/AVC intra prediction modes is introduced, an LUT(Look Up Table) for each mode is designed. Through the implementation and performance evaluation, while the subject quality difference between the output of our scheme and a conventional output is negligible, the former can extract the thumbnail faster than the latter by up to 63%.

keyword: Thumbnail, Reduced-size image

a) 광운대학교, VIA-Multimedia Center  
Kwangwoon University, VIA-Multimedia Center

b) 삼성전자  
Samsung Electronics

† 교신저자 : 유상준(yusj1004@media.kw.ac.kr)

\* 이 논문은 2007년도 광운대학교 교내 학술연구비(60012007179) 지원과 2007년도 ETRI 프로젝트(90262007023) 지원에 의해 수행된 연구임

## 1. 서론

최근에 디지털 동영상 처리기술이 급격히 발전함에 따라 고화질 디지털 방송(HDTV: High Definition Television), 디지털 멀티미디어 방송(DMB: Digital Multimedia Broad-

casting), 인터넷 TV(IPTV: Internet Protocol Television)등과 같이 다양한 매체를 이용한 디지털 멀티미디어 서비스가 활성화 되고 있다. 특히 고화질 디지털 방송이 일반화되면서 다양한 서비스 어플리케이션이 개발되고 있으며, 고화질, 고해상도의 영상을 위한 고속 동영상 처리 기술들이 요구 되고 있다<sup>[1-3]</sup>.

비디오 데이터의 효율적인 보관 및 전송을 위해 MPEG (Moving Picture Experts Group)과 같은 동영상 압축에 관한 국제 표준안이 널리 사용되고 있다. MPEG은 시각적으로 좋은 화질을 유지하면서도 비디오 자료를 표현하는데 필요한 비트 량을 혁신적으로 감소시키는 압축 기법이지만, 압축된 자료는 영상처리의 목적으로 다루기가 쉽지 않다. 압축된 자료의 효과적인 처리를 위하여 빠른 검색 및 복원을 위한 알고리즘에 대해 많은 연구가 진행되었다<sup>[4-6]</sup>. 이러한 연구 중의 하나가 축소 영상에 대한 연구이다. 축소 영상은 실제 영상보다 작은 크기로 줄여진 영상 또는 블록의 DC계수 값만으로 이루어진 DC 영상을 의미하며, 흔히 썸네일(Thumbnail)이라고 부른다. 이러한 썸네일은 영상의 특성을 대략적으로 표현하는데 필요한 전반적인 요소들을 지니고 있으므로 이를 이용하여 영상의 고속 검색이나 색인에 많이 이용된다. 더욱이 썸네일 영상의 추출은 완전한 크기로의 복원에 비해 보다 적은 메모리가 필요 하며, 또한 더 적은 연산 능력만으로 수행이 가능하므로 MPEG 기반으로 압축된 동영상의 효과적인 탐색에 사용되고 있다<sup>[7-10]</sup>.

블록단위의 DCT 부호화로 부호화된 동영상에서 축소 영상을 추출하는 대표적인 방법은 DCT 영역에서 DC 계수를 직접 추출하는 방법이다. 이 방법은 비트스트림을 완전히 복호화 하지 않고 DCT 영역에서 DC 계수 값을 직접 추출함으로써 연산량을 줄일 수 있다. 때문에 블록단위의 DCT 부호화 방법을 사용하는 MPEG-2, H.263, H.263+, MPEG-4와 같은 종래의 비디오 표준에서 DCT 영역에서 DC 계수 값을 직접 추출하는 방법이 널리 사용되고 있다<sup>[11-15]</sup>.

최근에 표준화가 완료된 H.264/AVC는 ITU-T 비디오 그룹과 ISO/IEC MPEG 그룹이 공동으로 개발한 비디오 표준으로 기존의 비디오 표준에 비해 월등한 성능의 압축률을 제공하지만 새로운 기술들을 채택함으로써 계산량이 매우 높아졌다<sup>[16]</sup>. 새롭게 채택된 기술 중의 하나인 인트라 예측

방법은 인트라 프레임에서 이전의 복호화된 주변 블록을 이용하여 현재 블록을 예측한다. 따라서 특정위치의 블록을 복원하기 위해서는 이전 블록들이 모두 복원되어 있어야 한다. 이러한 문제점 때문에 H.264/AVC 비트스트림으로부터 DCT 영역에서 직접 썸네일을 생성하는 방법이 존재하지 않았다.

Chen은 H.264/AVC의 인트라 예측을 행렬의 곱셈으로 나타내고, 행렬 연산을 이용하여 인트라 예측 블록을 생성할 수 있는 방법을 제안하였다<sup>[17]</sup>. Chen의 방법을 주파수 공간으로 변환하면 주파수 공간에서 인트라 예측을 수행할 수 있다. 이 방법은 주파수 공간에서 MPEG-2 비트스트림을 H.264/AVC 비트스트림으로 전환하는 트랜스코더에 많이 사용되고 있다<sup>[17][18]</sup>. 본 논문에서는 Chen의 주파수 공간에서의 인트라 예측 방법을 이용하여 주파수 공간에서 썸네일을 생성시킬 수 있는 방법을 제안한다.

제안하는 방법은 먼저 수직 여과기(Vertical Filter)와 수평 여과기(Horizontal Filter)를 정의하고, 이를 이용하여 인트라 예측을 일반화된 수식으로 정리 한다. 제안하는 방법은 주파수 공간에서 직접 썸네일을 추출하기 때문에 IDCT (Inverse DCT)과정과 화소 공간에서의 인트라 예측을 생략하여 연산을 줄일 수 있다. 뿐만 아니라 썸네일 추출을 위해 사용되는 연산은 수직/수평 여과기의 주파수 변환특성을 이용하기 때문에 계산이 간단하다.

논문의 구성은 다음과 같다. 2장에서는 H.264/AVC의 인트라 예측에 대한 간략한 소개와 인트라 예측을 행렬 곱의 형태로 일반화시켜 표현한다. 또한 썸네일 생성을 위한 수식과, 이를 위한 연산자와 특수 행렬을 정의한다. 3장에서는 2장에서 정리한 일반식을 바탕으로 주파수 공간에서 썸네일을 추출하는 방법을 제안한다. 4장에서는 제안한 방법을 적용한 실험과 결과를 보여주며, 5장에서는 결론을 맺는다.

## II. H.264/AVC 인트라 예측과 일반화

H.264/AVC에서는 인트라(Intra) 프레임에서 공간상의 중복성을 제거하기 위해 인트라 예측 방법을 이용한다. 인

트라 예측 방법은 기존의 다른 표준에 비해서 H.264/AVC의 압축 효율을 향상시키는 기술 중의 하나이다. 이번 장에서는 H.264/AVC의 인트라 예측 방법에 대해서 살펴보고 일반화된 수식으로 정리한다.

### 1. H.264/AVC 인트라 예측

H.264/AVC의 인트라 예측방법은 블록의 크기에 따라 16×16, 8×8, 4×4, 3가지 분류로 나눌 수 있다. 4×4, 8×8 하위 블록(sub-block)의 경우에 9가지의 예측 모드를 가지며, 16×16 블록과 색차 8×8 블록에서는 4가지의 예측 모드를 가진다. 하위 블록 또는 매크로 블록이 인트라 모드로 부호화 되었을 경우 예측 블록은 현재 부호화될 블록의 이웃하는 위쪽 또는 왼쪽 블록의 복원된 화소로 구성된다. 그림 1은 4×4 블록에 대한 인트라 예측을 위한 참조 화소와 8가지의 예측 방향을 나타낸다.

그림 1(a)에서  $y$ 는 현재 예측할 블록을 나타내며  $X_1, X_2, X_3, X_4$ 는 이전에 부호화 되어 복원된 주위 블록이다.  $M$ 은  $X_1$ 블록내의 참조 화소를 나타내며,  $A, B, C, D$ 는  $X_2$ 블록 내의 참조 화소,  $E, F, G, H$ 는  $X_3$ 블록 내의 참조 화소,  $I, J, K, L$ 은  $X_4$ 블록 내의 참조 화소를 나타낸다. 그림 1(b)는 8가지의 예측 모드에 따른 예측 방향을 나타내며 모드 2(DC)경우는 방향이 없기 때문에 표시 하지 않고 있다. 최종적으로 인트라 예측 블록은 예측 모드의 방향에 따라 주위 참조 화소간의 산술적인 연산을 이용하여 구해진다<sup>[16]</sup>.

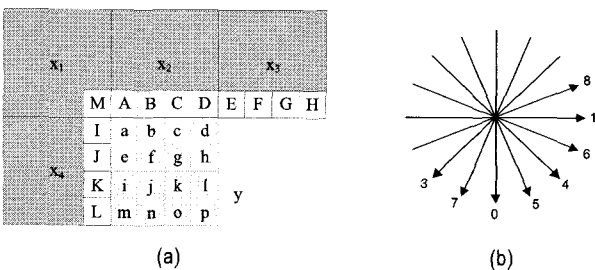


그림 1. 4×4 인트라 예측, (a) 예측 블록과 참조 화소 (b) 예측 방향  
 Fig. 1. 4×4 intra prediction modes, (a) Prediction block and its reference pixels (b) 8 direction of prediction

### 2. 4×4 인트라 예측의 일반화

Chen은 인트라 예측을 행렬의 소인수 분해를 통해서 행렬의 곱으로 나타 낼 수 있음을 보여주었다<sup>[17]</sup>. Chen의 방법은 모드에 따른 각기 다른 수식이 필요하다. 본 논문에서는 먼저 인트라 예측을 일반화된 수식으로 정리한다. 또한 인트라 예측의 주과수 특성을 이용하기 위해 수직 여과기(Horizontal Filter)와 수직 여과기(Vertical Filter)를 정의하며, 이를 이용하여 변형된 인트라 예측 수식으로 전개한다.

$$y_{pred}^m = \left( \sum_{n=1}^3 \sum_{i=1}^4 s_i x_n c_{n,i}^m \right) + \left( \sum_{i=1}^4 c_{4,i}^m x_4 s_i^T \right) \quad (1)$$

수식 (1)은 인트라 예측의 일반화된 수식을 나타낸다. 여기에서  $m$ 은 인트라 예측 모드를 나타내며,  $x_n$ 은 그림 1(a)에서의  $n$ 번째 이웃하는 블록을 나타낸다.  $c_{n,i}^m$ 는 참조 블록에 곱해지는 상수 행렬을 의미하며, 인트라 예측모드에 따른 소인수 분해된 행렬이다<sup>[17]</sup>.  $s_i$ 는 이동 행렬(Shift matrix)을 나타내며 다음의 수식으로 정의된다.

$$s_i = (a_{ij})_{4 \times 4}, \quad a_{ij} = \begin{cases} 1, & \text{if } j=4 \text{ and } i=\alpha (1 \leq \alpha \leq 4) \\ 0, & \text{else} \end{cases} \quad (2)$$

그림 1(a)와 같이 인트라 예측은 현재 예측 하고자하는 블록과 이웃하는 주변 블록과 경계를 이루는 화소만을 참조한다. 따라서  $x_n$ 의 하단 또는 우측의 값만을 추출하기 위한 여과기와 일반화 과정에 필요한 연산자를 다음과 같이 정의한다.

a) 특수 행렬  $\tilde{v}, \tilde{h}, u, z$ 는 다음과 같이 정의한다.

$$\tilde{v} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \tilde{h} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

$$u = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad z = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix}$$

b) 수직복사 연산자  $V(\cdot)$ : 수직 복사 행렬  $\tilde{v}$ 와 임의의 행렬  $a$ 의 행렬 곱으로 정의한다.

$$V(a) = \tilde{v}a = a_v \quad (3)$$

c) 수평복사 연산자  $H(\cdot)$ : 임의의 행렬  $a$ 와 수평 복사 행렬  $\tilde{h}$ 의 행렬 곱으로 정의한다.

$$H(a) = a\tilde{h} = a_h \quad (4)$$

d) 평균 연산자  $M(\cdot)$ : 임의의 행렬  $a$ 의 모든 원소의 합을 행렬의 크기로 나눈 값으로 정의 한다. 이 값은 스칼라(Scalar)를 갖는다.

$$M(a) = \frac{1}{N \times N} \left( \sum_{i,j} a_{ij} \right) \quad (5)$$

e) UNI 연산자  $U(\cdot)$ : 임의의 행렬  $a$ 의  $M(a)$ 로 모든 원소가 구성된 행렬로 정의한다.

$$U(a) = M(a)u = \frac{1}{N \times N} uau \quad (6)$$

이동 행렬과 수직/수평 복사 행렬의 특성에 의하여 다음과 같은 몇 가지 성질이 성립한다.

단,  $A, B$ 는  $4 \times 4$  크기의 임의의 행렬이다.

$$s_i = s_i \tilde{v} = \tilde{h} s_i^T \quad (7-1)$$

$$s_i A = s_i V(A) = s_i A_v \quad (7-2)$$

$$A s_i^T = H(A) s_i^T = A_h s_i^T \quad (7-3)$$

$$V(s_1) = V(s_2) = V(s_3) = 0, V(s_4) = \tilde{v} \quad (7-4)$$

$$H(s_1^T) = H(s_2^T) = H(s_3^T) = 0, H(s_4^T) = \tilde{h} \quad (7-5)$$

$$A_v B_h s_i^T = (s_i A_v B_h)^T \quad (7-6)$$

수식 (7-2), (7-3)의 성질들을 이용하여 수식 (1)을 정리하면 다음과 같다.

$$y_{\text{pred}}^m = \left( \sum_{u=1}^3 \sum_{i=1}^4 s_i x_{n,v} c_{n,i}^m \right) + \left( \sum_{i=1}^4 c_{4,i}^m x_{4,h} s_i^T \right) \quad (8)$$

수식 (1)과 (8)에서 인트라 예측 블록  $y_{\text{pred}}^m$ 을 구하기 위해서는 이웃하는 블록의 모든 값을 필요로 하지 않고,  $x_{n,v}, x_{n,h}$  값만이 필요함을 알 수 있다. 또한 현재 블록은 다음 예측 블록의 참조 블록으로 사용됨으로 현재 블록의 복원된 블록의 수직/수평 여과된 값을 저장하여 다음 블록의 연산에 사용될 수 있다.

수식(9)는 현재 블록  $y$ 의 예측 블록  $y_{\text{pred}}^m$ 이 다음 블록  $y'$ 의 인트라 예측을 위해 수직/수평 여과되는 과정을 보여 준다.

$$y'_{\text{pred}}^m = \left( \sum_{n=1}^3 \sum_{i=1}^4 s_i V(y_{\text{pred}}^m) c_{n,i}^m \right) + \left( \sum_{i=1}^4 c_{4,i}^m H(y_{\text{pred}}^m) s_i^T \right) \quad (9)$$

여기서  $y'_{\text{pred}}^m$ 는  $y'$ 블록의 인트라 예측 블록이다.  $V(y_{\text{pred}}^m)$ ,  $H(y_{\text{pred}}^m)$ 는 각각 다음과 같다.

$$y_{\text{pred},v}^m = V(y_{\text{pred}}^m) = \left( \sum_{n=1}^3 \sum_{i=1}^4 V(s_i x_{n,v} c_{n,i}^m) \right) + \left( \sum_{i=1}^4 V(c_{4,i}^m x_{4,h} s_i^T) \right) \quad (10)$$

$$y_{\text{pred},h}^m = H(y_{\text{pred}}^m) = \left( \sum_{n=1}^3 \sum_{i=1}^4 H(s_i x_{n,v} c_{n,i}^m) \right) + \left( \sum_{i=1}^4 H(c_{4,i}^m x_{4,h} s_i^T) \right) \quad (11)$$

수식 (10), (11)을 수식 (7)의 성질을 이용하여 다음과 같이 간략화 시킬 수 있다.

$$y_{\text{pred},v}^m = x_{1,v} c_{1,4}^m + x_{1,v} c_{1,4}^m + x_{1,v} c_{1,4}^m + (p_{4,v}^m x_{1,h})^T, \quad (12)$$

여기에서  $p_{4,v}^m = \sum_{i=1}^4 (s_i V(c_{4,i}^m))$

$$y_{pred,h}^m = x_{1,v}q_{1,h}^m + x_{2,v}q_{2,h}^m + x_{3,v}q_{3,h}^m + c_{4,4}^m x_{4,h}$$

$$\text{여기에서 } q_{4,h}^m = \sum_{i=1}^4 (H(c_{n,i}^m) s_i^T) \quad (13)$$

8×8 인트라 예측은 4×4 인트라 예측과 동일한 모드와 방법을 사용한다. 따라서 본 논문에서 제안하는 방법을 이용하여 동일하게 구할 수 있다. 본 논문에서는 8×8 인트라 예측에 대한 수식 전개는 생략한다.

### 3. 16×16 인트라 예측의 일반화

16×16 인트라 예측 모드는 매크로 블록 전체에 대해 하나의 예측 모드를 사용하는 것으로 4가지 방향 모드에 의해 수행된다. 16×16 인트라 예측 모드도 4×4 인트라 예측 모드와 유사한 계산 방법이 사용된다. 16×16 인트라 예측 블록은 다음과 같은 4가지 모드를 사용한다. 모드 0은 수직 방향으로 위쪽 블록의 맨 하단의 행을 참조 한다. 모드 1은 수평 방향으로 왼쪽 블록의 맨 우측 열을 참조 한다. 모드 2는 DC 모드로 방향성이 존재 하지 않으며, 위쪽 블록의 맨 하단 행과 왼쪽 블록의 맨 우측 열의 평균값을 이용한다. 마지막 모드는 평면모드(Plane)로 위쪽 블록과 좌측 블록을 이용하여 평면의 방정식을 이용하여 수행한다<sup>[16]</sup>. 그림 2는 16×16 인트라 예측의 평면 모드를 나타낸다. 평면 모드를 제외한 3가지 모드는 앞서서의 방법과 동일하게 정리 가능하기 때문에 생략하며 이 장에서는 평면의 모드에 대해서만 정리하기로 한다.

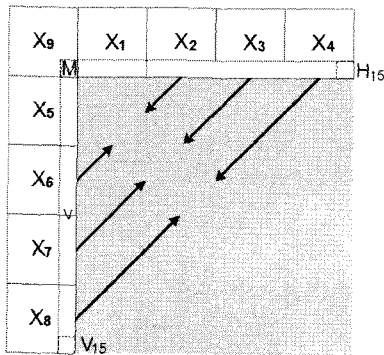


그림 2. 인트라 16×16 평면 예측 방법과 참조 화소  
Fig. 2. Intra 16×16 plane prediction mode and its reference pixels

16×16 평면 모드의 예측 블록을 생성하는 수식은 다음과 같다.

$$y_{pred}^{plane} = a \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} + b \begin{bmatrix} -7 & -6 & \dots & 8 \\ \vdots & \vdots & \vdots & \vdots \\ -7 & -6 & \dots & 8 \end{bmatrix} + c \begin{bmatrix} -7 & \dots & -7 \\ -6 & \dots & -6 \\ \vdots & \dots & \vdots \\ 8 & \dots & 8 \end{bmatrix} \quad (14)$$

여기에서  $a = 16(H_{15} + V_{15})$ ,  $b = 16(5H + 32) \gg 6$ ,  $c = 16(5V + 32) \gg 6$ 이다.  $y_{pred}^{plane}$ 는 평면 모드의 예측 블록을 나타낸다. 평면 모드에서 사용되는 수식을 정리하면 다음과 같다.

$$H = 8(H_{15} - M) + \sum_{h=0}^6 (h+1)(H_{8+h} - H_{6-h})$$

$$V = 8(V_{15} - M) + \sum_{v=0}^6 (v+1)(V_{8+v} - V_{6-h})$$

$$V_{15} = M(\tilde{v}x_{8,h}) = (HT(\tilde{v}x_{8,h}))_{(0,0)}$$

$$H_{15} = M(x_{4,v}\tilde{h}) = (HT(x_{4,v}\tilde{h}))_{(0,0)}$$

$$M = M(x_{9,v}\tilde{h}) = M(\tilde{v}x_{9,h}) = (HT(x_{9,v}\tilde{h}))_{(0,0)} = (HT(\tilde{v}x_{9,h}))_{(0,0)}$$

여기에서 HT(H.264/AVC modified DCT)는 H.264/AVC에서 사용되는 정수 변환이다.  $(\cdot)_{(0,0)}$ 는 행렬에서 (0,0) 위치의 계수 값을 취하는 연산자이다. 주파수 공간에서 (0,0) 위치의 계수 값 즉 DC값은 화소 공간에서 블록의 평균과 같기 때문에  $M(\cdot)$  연산자를 이용하여 나타낼 수 있다. 평면 모드의 연산에 필요한 값들 또한  $x_{n,v}$ ,  $x_{n,h}$ 을 이용하여 계산 될 수 있음을 보여준다.

H, V를 앞서서 정의한 수직/수평 여과기로 처리된 값으로 정리하면 다음과 같다.

$$H = -8M - 7H_0 - 6H_1 \dots + 7H_{15}$$

$$= -8M + M((x_{1,v} + x_{2,v} + x_{3,v} + x_{4,v})z^T) + M((-7x_{1,v} - 3x_{2,v} + x_{3,v} + 5x_{4,v})u)$$

$$V = -8M + M(z(x_{1,h} + x_{2,h} + x_{3,h} + x_{4,h})) + M(u(-7x_{1,h} - 3x_{2,h} + x_{3,h} + 5x_{4,h}))$$

마찬가지로, 수식 (14)를 4×4 행렬로 분해하여 정리 하면 다음과 같이 나타낼 수 있다.

$$y_{pred}^{plane} = a \begin{bmatrix} u & u & u & u \\ u & u & u & u \\ u & u & u & u \\ u & u & u & u \end{bmatrix} + b \begin{bmatrix} z-7u & z-3u & z+u & z+5u \\ z-7u & z-3u & z+u & z+5u \\ z-7u & z-3u & z+u & z+5u \\ z-7u & z-3u & z+u & z+5u \end{bmatrix} + c \begin{bmatrix} z^T-7u & z^T-7u & z^T-7u & z^T-7u \\ z^T-3u & z^T-3u & z^T-3u & z^T-3u \\ z^T+u & z^T+u & z^T+u & z^T+u \\ z^T+5u & z^T+5u & z^T+5u & z^T+5u \end{bmatrix} \quad (15)$$

수식 (15)을 간단히 하기 위하여  $y_{pred}^{plane}$  을 4×4 크기를 갖는  $y_{ij}$  행렬을 이용하여 수식 (16)과 같이 나타낼 수 있다.

$$y_{pred}^{plane} = \begin{bmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{bmatrix} = \begin{bmatrix} y_{00} & y_{00} + 4bu & y_{01} + 4bu & y_{02} + 4bu \\ y_{00} + 4cu & y_{01} + 4cu & y_{02} + 4cu & y_{03} + 4cu \\ y_{01} + 4cu & y_{11} + 4cu & y_{12} + 4cu & y_{13} + 4cu \\ y_{02} + 4cu & y_{12} + 4cu & y_{22} + 4cu & y_{23} + 4cu \end{bmatrix} \quad (16)$$

여기에서  $y_{00} = (a - 7b - 7c)u + bz + cz^T$ 이다.

수식 (16)을 살펴보면 16×16 평면 모드의 예측 블록은 가로 성분의 4×4 행렬은 이전 행렬에 대해서 4bu 만큼 증가함을 볼 수 있으며, 수직 성분의 4×4 행렬은 상위 행렬에 대해서 4cu 만큼 증가 한다. 나중에 이러한 성질을 이용하여 내부 행렬에 대한 연산은 수행하지 않고 맨 우측과 맨 하단의 행렬연산만 수행하여 연산량을 줄일 수 있다.

수식 (17)과 같이 16×16 평면 모드로 예측된 블록에 대해서 다음 블록을 위한 수평/수직 연산을 계산하면 다음과 같다. 이때 “×”는 예측 블록의 내부에 있는 4×4 행렬로 다음 블록의 예측에 참조되지 않기 때문에 연산을 생략한다.

$$y_{pred,v}^{plane} = V(y_{pred}^{plane}) = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ y_{30,v} & y_{30,v} + 4bu & y_{30,v} + 8bu & y_{30,v} + 12bu \end{bmatrix} \quad (17)$$

여기에서  $y_{30,v} = (a - 7b + 8c)u + bz$

$$y_{pred,h}^{plane} = H(y_{pred}^{plane}) = \begin{bmatrix} \times & \times & \times & y_{03,h} \\ \times & \times & \times & y_{03,h} + 4cu \\ \times & \times & \times & y_{03,h} + 8cu \\ \times & \times & \times & y_{03,h} + 12cu \end{bmatrix} \quad (18)$$

여기에서  $y_{03,h} = (a + 8b - 7c)u + cz^T$

지금까지 H.264/AVC의 인트라 예측에 대해서 간략히 설명하였으며, 화소 공간에서 인트라 예측을 일반화된 수식으로 정리하였다. 또한 수직/수평 여과기와 특수 행렬을 정의하여 일반수식을 변형하였다. 변형된 수식은 참조 블록의 전체 값이 아니라 경계를 이루는 화소만을 참조한다. 주파수 공간에서의 수식은 수식(8),(16)의 양변에 HT (H.264/AVC modified DCT)를 수행하여 얻을 수 있다. 색차 성분의 인트라 예측도 16×16방법과 동일하게 처리된다.

다음 장에서는 변형된 수식을 이용하여 썸네일을 추출하는 방법에 대해서 설명한다.

### III. 썸네일 추출 방법

앞서서 인트라 예측에 대한 일반화 과정과 수직/수평 여과기를 통해 수식을 정리하였다.

또한 변형된 인트라 예측 식을 통하여 인트라 예측을 수행할 때 주변 블록의 전체가 필요하지 않고 이웃하는 블록의  $x_{n,v}, x_{n,h}$ 의 값만을 이용하여 다음 블록을 쉽게 예측할 수 있음을 증명하였다. 이번 장에서는 변형된 수식을 이용하여 주파수 공간에서 썸네일을 생성하는 방법을 제안한다.

현재 블록  $y$ 는 인트라 예측 블록  $y_{pred}^m$ 과 예측오차  $y_e$ 의 합으로 나타낼 수 있다.

$$y = y_{pred}^m + y_e = \left( \sum_{n=1}^3 \sum_{i=1}^4 s_i x_{n,v} c_{n,i}^m \right) + \left( \sum_{i=1}^4 c_{4,i}^m x_{4,h} s_i^T \right) + y_e \quad (19)$$

화소 공간에서의 가로, 세로 각각 1/4의 축소된 영상  $y_{dc}$  는 현재 블록  $y$ 의 화소 값의 평균으로 구할 수 있다. 축소 영상은 인트라 예측 블록의 DC 값과 예측 오차에 대한 주파수 공간에서의 DC 값의 합을 이용하여 계산이 가능하다. 화소 공간에서의 블록의 평균은 주파수 공간에서 DC 값과 같기 때문에 수식 (19)에  $U(\cdot)$  연산을 적용하여 다음과 같이 나타낼 수 있다.

$$\begin{aligned} y_{dc} &= (U(y))_{(0,0)} = (U(y_{pred}^m) + U(y_e))_{(0,0)} \\ &= \frac{1}{4} (HT(y))_{(0,0)} \\ &= \frac{1}{4} (HT(y_{pred}^m))_{(0,0)} + \frac{1}{4} (HT(y_e))_{(0,0)} \end{aligned} \quad (20)$$

HT 공간에서 (0,0) 위치 값(DC)은 화소 공간에서의 블록 평균을 나타내며, 화소 공간에서의 블록 평균보다 4배가 크다. 따라서 썸네일 영상  $y_{dc}$ 는 HT변환된 블록의 DC 값을 1/4 곱하여 구할 수 있다.

수식 (20)에서  $U(y_{pred}^m)$ 을  $U(\cdot)$  연산자와  $x_{n,v}, x_{n,h}$  값을 이용하여 정리하면 다음과 같다.

$$\begin{aligned} y_{pred,uni}^m &= U(y_{pred}^m) \\ &= \frac{1}{16} \sum_{n=1}^3 x_{n,v} \sum_{i=1}^4 c_{n,i}^m u + \frac{1}{16} \sum_{i=1}^4 u c_{4,i}^m x_{4,h} \\ &= \sum_{n=1}^3 x_{n,v} c_n^m + (c_4^m)^T x_{4,h} \\ \text{여기에서 } c_n^m &= \frac{1}{16} \sum_{i=1}^4 c_{n,i}^m u \end{aligned} \quad (21)$$

수식 (21)에서  $y_{pred,uni}^m$ 는 현재 블록에 대한 썸네일 값이다. 16×16 인트라 예측 블록에서도 4×4 경우와 비슷하게 썸네일을 구할 수 있다. 16×16 인트라 블록에서는 4×4 블록 크기로 축소된 영상이 생성된다. 16×16 인트라 예측 모드 0, 1, 2는 4×4 인트라 예측 모드와 동일한 방법을 통하여 계산 가능하기 때문에 생략하고, 평면 모드에 대해서만 축소 영상을 추출하는 방법에 대해 설명한다.

수식 (22)는 16×16 인트라 평면 모드에서  $U(\cdot)$  연산을 이용하여 썸네일을 생성하는 방법을 나타낸다.

$$\begin{aligned} y_{pred,uni}^{plane} &= U(y_{pred}^{plane}) = \\ &= \begin{bmatrix} y_{00,uni} & y_{00,uni} + 4bu & y_{01,uni} + 4bu & y_{02,uni} + 4bu \\ y_{00,uni} + 4cu & y_{01,uni} + 4cu & y_{02,uni} + 4cu & y_{03,uni} + 4cu \\ y_{01,uni} + 4cu & y_{11,uni} + 4cu & y_{12,uni} + 4cu & y_{13,uni} + 4cu \\ y_{02,uni} + 4cu & y_{12,uni} + 4cu & y_{22,uni} + 4cu & y_{23,uni} + 4cu \end{bmatrix} \quad (22) \\ y_{00,uni} &= (a - 5.5b - 5.5c)u \end{aligned}$$

최종적으로 16×16 인트라 평면모드에서 수식 (22)에서 각각의 4×4 행렬에  $(\cdot)_{(0,0)}$  연산을 통하여 가로, 세로 각각 1/4로 축소된 썸네일 영상을 얻을 수 있다.

지금까지 화소 공간에서 제안하는 방법을 이용하여 썸네일을 추출하는 방법에 대해서 설명하였다. 제안하는 방법은 부호화 스트림으로부터 완전 복호화 하지 않고 주파수 공간에서 직접 썸네일을 생성하기 위해서는 변형된 수식을 HT 변환하여 처리함으로써 가능하다.

화소 공간에서 유도된 수식들을 주파수 공간으로 수식을 바꾸어 정리하면 다음과 같다.

$$\begin{aligned} Y_{pred,v}^m &= HT(x_{1,v}c_{1,4}^m + x_{2,v}c_{2,4}^m + x_{3,v}c_{3,4}^m + (P_{1,4}^m x_{4,h})^T) \\ &= X_{1,v}C_{1,4}^m + X_{2,v}C_{2,4}^m + X_{3,v}C_{3,4}^m + (P_{1,4}^m X_{4,h})^T \end{aligned} \quad (23)$$

$$Y_{pred,h}^m = \left( \sum_{n=1}^3 X_{n,v} Q_{n,h}^m \right)^T + C_{4,4}^m X_{4,h} \quad (24)$$

$$Y_{pred,uni}^m = \sum_{n=1}^3 X_{n,v} C_n^m + (C_4^m)^T X_{4,h} \quad (25)$$

여기에서  $C_{n,i}^m, X_{n,v}, X_{n,h}$ 는  $c_{n,i}^m, x_{n,v}, x_{n,h}$ 의 HT 변환된 행렬이다. 따라서 주파수 공간에서의 썸네일은 수식 (25)에 1/4를 곱하여 계산된다.

제안하는 구조는 주파수 공간에서 직접 썸네일을 추출한다. 그림 3은 제안하는 썸네일 추출기 구성도를 보여준다.

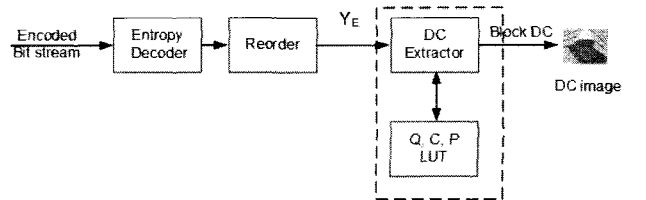


그림 3. 제안하는 H.264/AVC부터 썸네일을 추출하는 방법  
Fig. 3. Proposed thumbnail extraction method in an H.264/AVC bit-streams

DC 추출기(DC Extractor)는 수식 (23) ~ (25)를 수행하여 주파수 공간상에서 빠르게 DC값을 뽑아낸다. 이때 사용되는 Q, C, P 테이블은 모드에 따른 상수 테이블 이므로 미리 계산하여 추가적인 연산을 피한다. DC 추출기에는 다음 블록의 인트라 예측 블록을 복원하기 위해서  $y_{pred,v}^m$  와  $y_{pred,h}^m$  를 계산하는 블록이 포함되어 있다. 그림 4는 DC 추출기의 내부 구조를 나타낸다.

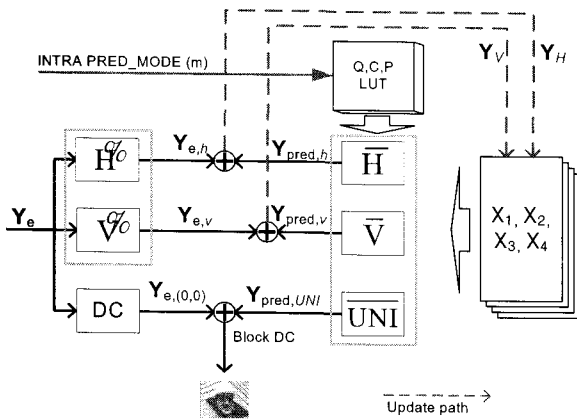


그림 4. 제안하는 DC 추출기의 구조  
Fig. 4. The structure of the proposed DC extractor

$Y_e$ 는 주파수 상의 예측 오차를 나타낸다.  $\tilde{H}, \tilde{V}$ 는 예측 오차의 주파수 값에 대한 여과 과정을 수행하며,  $\bar{H}, \bar{V}$ 는 이웃하는 블록  $X_1, X_2, X_3, X_4$ 을 이용하여  $Y_{pred}$  값을 구하는 연산을 수행한다. 마찬가지로  $\bar{UNI}$ 는  $X_1, X_2, X_3, X_4$ 을 이용하여 썸네일을 구하기 위해  $U(\cdot)$  연산과정을 수행한다. 최종적으로 주파수 영역에서 구하고자 하는 썸네일은

$U(\cdot)$  연산을 수행한 후 얻은  $Y_{pred,uni}$ 의 (0,0)값에  $Y_e$ 의 (0,0) 값을 더하여 얻는다.

제안하는 추출기의 동작 순서는 다음과 같다.

- Step 1) 입력된 비트스트림으로부터 역 양자화 과정까지 거쳐 생성된  $Y_e$ 에 대하여 각각 수직과 수평 여과 과정을 수행하여  $Y_{e,h}$ 와  $Y_{e,v}$ 를 생성한다.
- Step 2) LUT와 이웃한 블록의 이전에 계산된  $X_{n,v}, X_{n,h}$ 을 이용하여  $Y_{pred,uni}, Y_{pred,v}, Y_{pred,h}$ 을 구한다.
- Step 3)  $(Y_{pred,uni})_{(0,0)}$ 과  $(Y_e)_{(0,0)}$ 을 이용하여 썸네일 값을 구한다.
- Step 4)  $Y_{pred,v}$ 와  $Y_{e,v}$  그리고  $Y_{pred,h}$ 와  $Y_{e,h}$ 의 합을 저장하며, 다음의 인트라 예측 블록 생성을 위해  $X_{n,v}, X_{n,h}$ 을 계산하여 저장한다.

#### IV. 실험 및 결과

본 논문에서는 주파수 공간상에서 직접적으로 썸네일을 추출할 수 있는 방법을 제안하였다. 썸네일은 원본의 영상에 대한 대략적인 정보를 제공하여 빠른 검색과 색인에 주로 이용된다. 그렇기 때문에 썸네일은 추출된 영상의 화질보다는 추출 수행시간이 중요하다. 이번 장에서는 제안하는 방법의 성능을 주관적 화질 비교와 수행 시간을 비교하여 보여 준다. 그림 5는 제안하는 방법의 성능을 평가하기 위한 블록도를 나타낸다.

그림 5에서 입력 영상을 화소 공간에서 1/16크기로 다운

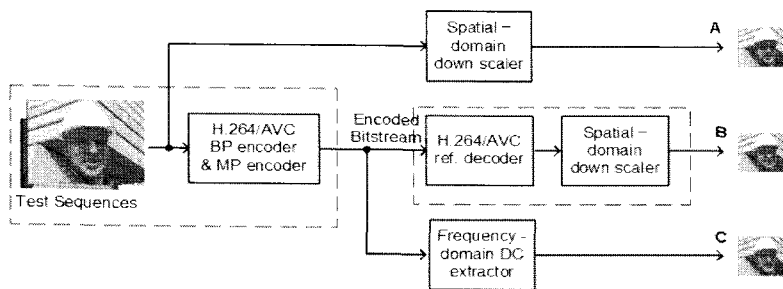


그림 5. 성능평가를 위한 블록도  
Fig. 5. A block diagram for performance evaluation



샘플링(down-sampling)을 수행한 영상 군을 A라 한다. 다운샘플링은 평균 연산자를 이용 하였다. 또한 H.264/ AVC 참조 복호화기에 의해 얻어진 영상을 1/16 크기로 다운샘플링을 수행한 영상 군을 B라 하며 제안하는 방법을 통해 얻은 영상 군을 C라고 한다. 각각의 영상 군을 추출하는 수행 시간과 각 영상군의 주관적 화질 비교를 통해 제안하는 방법의 성능을 검증 하고자 한다.

표 1은 B 영상 군을 추출하기 위한 H.264/AVC 참조 소프트웨어의 환경 변수이다.

표 1. 실험환경

Table 1. Experimental environment

구분	실험 환경	
참조 소프트웨어	JM 12.2 BP, Level 4.0	
프레임 부호화 모드	인트라	
적응형 프레임/필드 모드	프레임 모드	
양자화 계수	9, 21, 33, 45	
실험 영상	352×288	Akiyo, Kettle, Foreman, Mobile
	704×576	City, Crew, Harbour, Soccer
	1920×1080	Clip, Table
썸네일 추출 크기	88×72 / 176×144 / 480×270	

그림 6 - 8은 화소 공간에서의 썸네일 추출 방법과 제안하는 방법과의 수행 시간을 비교 하였다. 그림에서의 SD (Spatial Domain)는 그림 5에서 H.264/AVC 참조 복호화기에서 얻어진 영상을 화소영역에서 다운샘플링을 하여 썸네일을 얻는 방법이며, TD(Transform Domain)는 주파수 공간에서 제안하는 방법을 통해 썸네일을 얻는 방법이다.

화소 공간에서 썸네일을 추출하는 방법에 비해서 제안하는 방법은 모든 영상에서 속도 향상을 가져온다. 그림 6-8은 각각의 실험 영상 중 한 프레임에서 썸네일을 추출하는데 소요되는 수행시간을 나타낸다. 그림 6은 352×288 크기를 갖는 영상을 88×72 크기로 축소할 때 소요되는 수행 시간을 측정된 결과이다. Akiyo, Foreman, Kettle과 같이 경계가 적은 단조로운 영상보다 Mobile과 같이 복잡한 영상에서 두 방법 모두 수행시간이 더 길게 나타난다. SD 방법보다 TD방법의 수행 시간이 더 크게 감소함을 알 수 있다. 그림 7, 8은 영상의 크기를 달리하여 수행한 결과이다.

그림 7, 8의 결과는 그림 6의 경우와 비슷하지만 영상의 크기가 증가 할수록 제안하는 방법이 기존의 방법에 비해 많은 시간 감소를 얻을 수 있음을 보여준다.

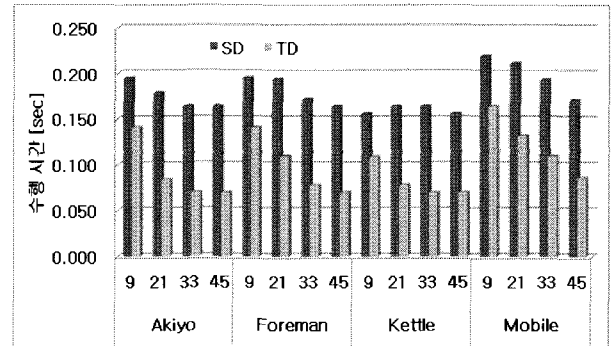


그림 6. QP에 따른 88×72 축소 영상 추출 수행 속도 비교

Fig. 6. Extracting time of SD and TD corresponding each QP

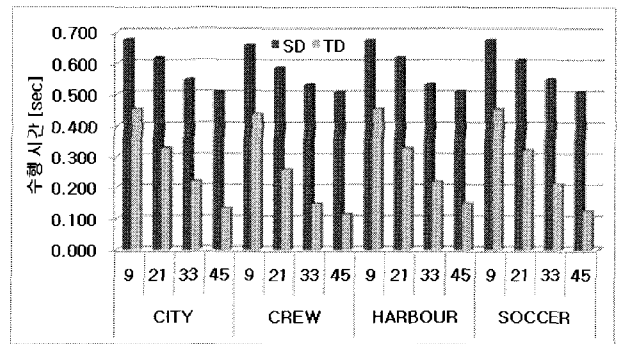


그림 7. QP에 따른 176×144 축소 영상 추출 수행 속도 비교

Fig. 7. Extracting time of SD and TD corresponding each QP

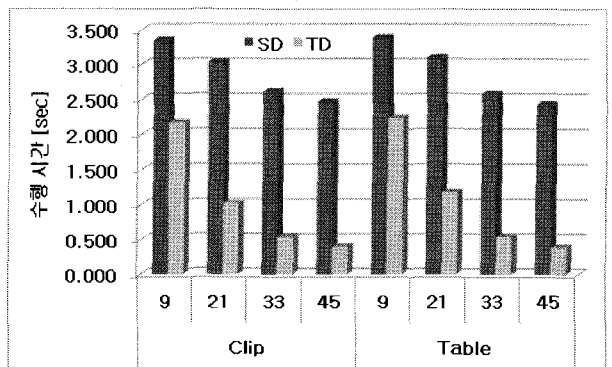


그림 8. QP에 따른 480×270 축소 영상 추출 수행 속도 비교

Fig. 8. Extracting time of SD and TD corresponding each QP

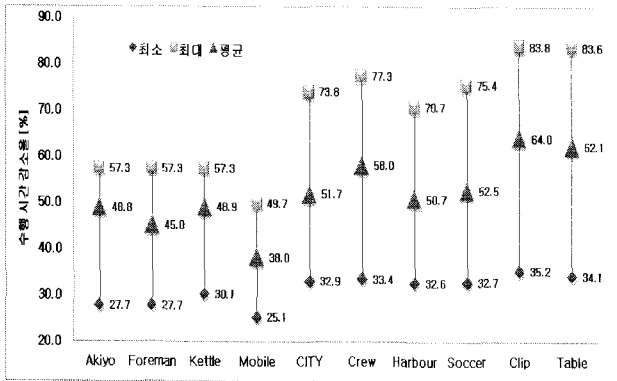


그림 9. 각 영상의 추출 수행시간 감소율  
Fig. 9. Extracting time saving of each test sequences

그림 10 - 12는 주관적인 화질 비교를 보여 준다. 그림 10은 Foreman 영상에 대한 결과이다. 원 영상에 비해 가로 세로 각각 1/4 크기로 감소한 영상에서도 원 영상의 특징을 가지고 있음을 알 수 있다. 따라서 작은 영상을

이용하여 원 영상에 대한 대략적인 정보를 얻을 수 있다. 그림 11, 12는 704×576의 크기를 갖는 Harbour, Soccer 영상에 대해 양자화 파라미터 6을 적용하였을 때 결과이다. 이 경우에서도 주관적인 화질은 크게 차이나지 않으며, 원 영상에 대한 대략적인 정보를 포함함을 알 수 있다.

H.264/AVC는 인트라 예측 과정과 오차 블록에 대한 주파수 공간상으로 변환과정에서 모두 정수 연산을 사용한다. 하지만 제안하는 방법에서는 주파수 공간상의 인트라 예측 전 과정에 대하여 부동소수점 연산을 사용한다. 주파수 공간상에서는 부동소수점 연산에 대한 반올림이나 버림 연산을 수행하기 힘들기 때문에 필연적으로 불일치가 존재하게 된다. 하지만 주관적 화질 비교에서는 큰 차이를 느낄 수 없으며, 썸네일에 대한 근본적인 요구 사항을 충분히 충족시킬 정도의 화질을 제공함을 알 수 있다.

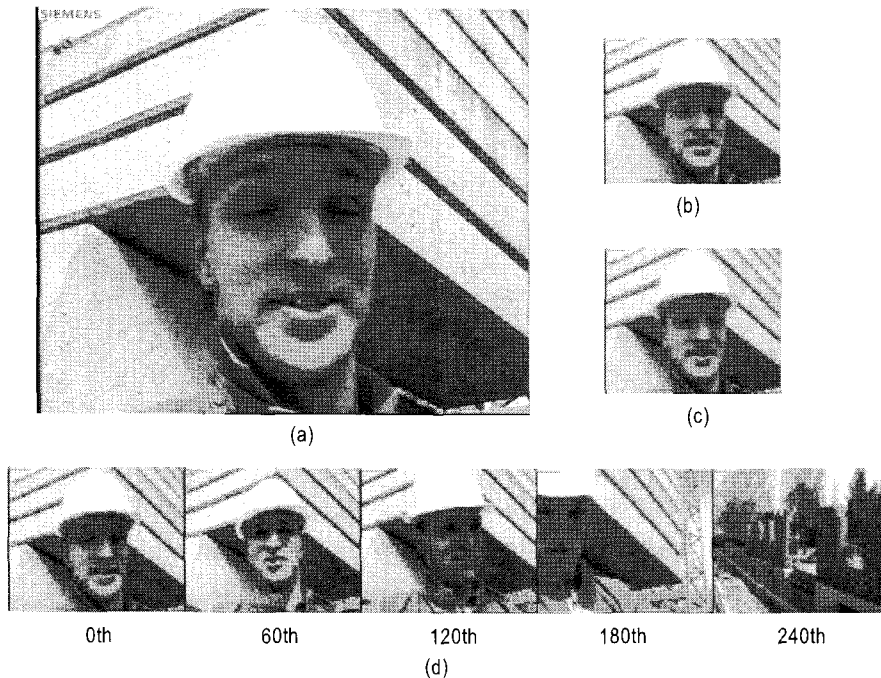


그림 10. Foreman 영상에서의 화질 비교, (a) 352×288 크기의 원 영상, (b) 원 영상을 1/16 크기로 축소된 영상, (c) QP 6에서 H.264/AVC 참조 복호화기를 통해 얻은 썸네일 (d) QP 6에서 매 60프레임마다 제안하는 방법을 통해 추출한 썸네일  
Fig. 10. Subjective quality performance of Foreman sequence. (a) Original size image (352×288) (b) 1/16 size image (88×72) (c) Thumbnail gained by H.264/AVC reference software decoder in QP 6 (d) Thumbnail extracted from proposed method on every 60th frame in QP 6

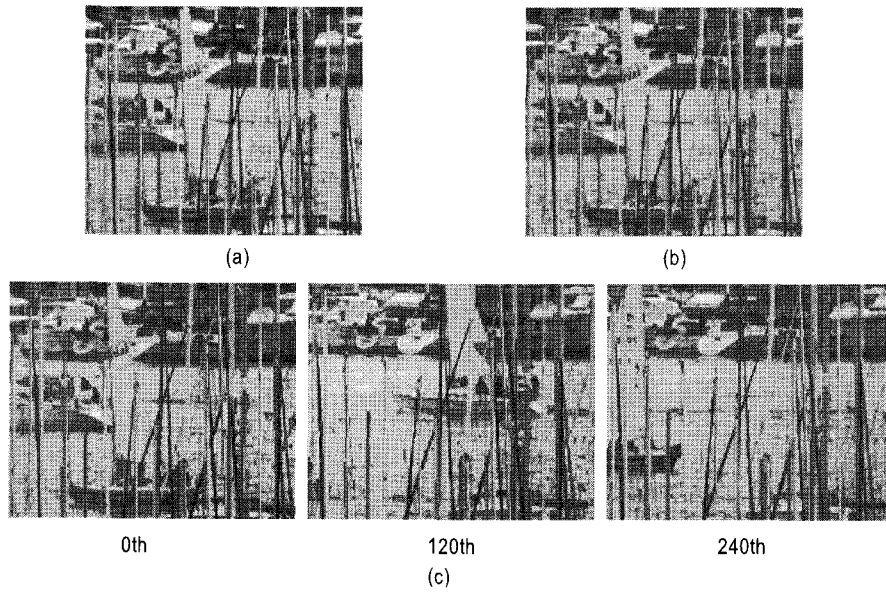


그림 11. Harbour 영상에서의 화질 비교, (a) 원 영상을 1/16크기로 축소한 영상, (b) QP 6에서 H.264/AVC 참조 복호화기를 통해 얻은 썸네일 (c) QP 6에서 매 120프레임마다 제안하는 방법을 통해 추출한 썸네일

Fig. 11. Subjective quality performance of Harbour sequence. (a) 1/16 size image (88×72) (b) Thumbnail gained by H.264/AVC reference software decoder in QP 6 (c) Thumbnail extracted from proposed method on every 120th frame in QP 6

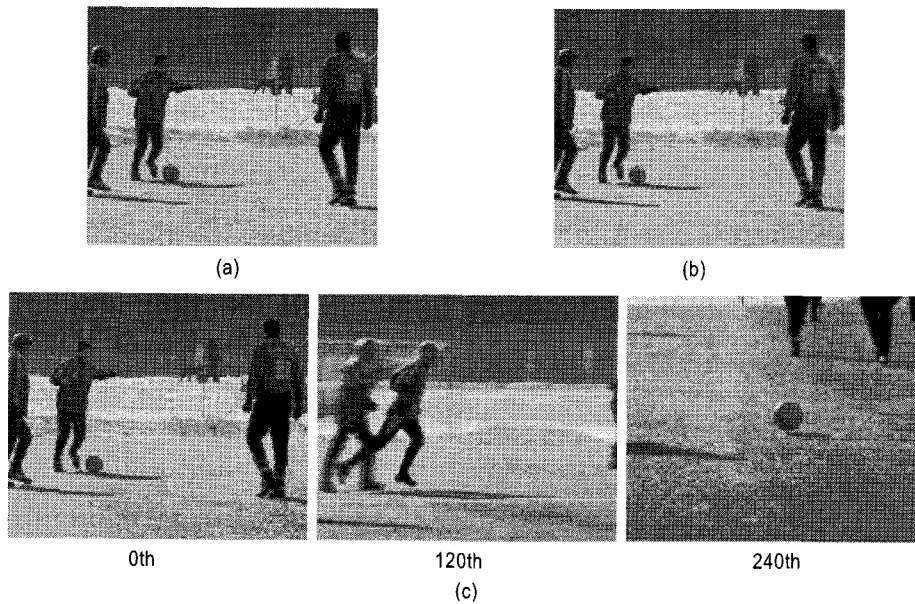


그림 12. Soccer 영상에서의 화질 비교, (a) 원 영상을 1/16크기로 축소한 영상, (b) QP 6에서 H.264/AVC 참조 복호화기를 통해 얻은 썸네일 (c) QP 6에서 매 120프레임마다 제안하는 방법을 통해 추출한 썸네일

Fig. 12. Subjective quality performance of Soccer sequence. (a) 1/16 size image (88×72) (b) Thumbnail gained by H.264/AVC reference software decoder in QP 6 (c) humbnail extracted from proposed method on every 120th frame in QP 6

### V. 결 론

본 논문에서는 H.264/AVC의 인트라 프레임으로부터 가로 세로 각각 1/4크기로 축소된 영상을 추출할 수 있는 방법을 제안하였다. 제안하는 방법은 기존의 일반적인 화소 공간에서 원영상 크기로 복원한 후 다운샘플링을 수행하여 축소 영상을 생성하는 방법과는 달리 주파수 공간에서 직접적으로 썸네일을 추출 할 수 있다. 제안하는 방법은 H.264/AVC의 인트라 예측을 화소 공간상의 일반화된 식으로 정리하였고, 그 수식을 바탕으로 주파수 공간에서 직접적으로 썸네일을 빠르게 추출한다.

제안하는 방법을 통해 얻어진 썸네일은 기존의 방법에 비해 화질의 차이가 존재 하지만, 영상의 전체적인 특성을 모두 표현할 수 있으며, 기존의 방법에 비해 최대 63%의 속도 향상을 얻을 수 있어서 영상의 인덱스와 탐색 등에 사용되는 썸네일의 근본적인 목적에 잘 부합됨을 보여 주었다. 제안하는 방법은 반올림 오차를 고려하지 않았기 때문에, 비트스트림을 완전히 복호화한 후에 축소하는 방법에 비해 주관적 화질 차이가 존재 한다. 향후에 정수 연산과 부동소수점 연산 사이에서 발생하는 불일치와 반올림 오차를 줄이기 위한 연구가 수행되어야 할 것이다.

### 참 고 문 헌

[1] B. Yeo and B. Liu, "Rapid scene analysis on compressed video", IEEE Trans. Circuits and Systems for video Technology, vol. 5, no. 6, pp. 533-540, 1995.

[2] Wing-San Chau, Oscar C. Au, Tak-Song Chong, Tai-Wai Chan, Chi-Shun Cheung, "Efficient Scene Change Detection in MPEG Compressed Video Using Composite Block Averaged Luminance Image Sequence," IEEE ICICS, pp.688-691, 2005.

[3] 윤자천, 설상훈, "압축영역에서 빠른 축소 영상 추출을 위한 다중부호 록업테이블 설계," 한국방송공학회, vol. 10, no.3, pp.413-421, May 2005.

[4] 김명준, 송병철, 장성규, 나종범, "변형된 IDCT 기저함수를 이용한 동

영상 축소 전환 기법," 한국통신학회 논문지, 제24권, 제11B호, pp.2147-2153, Nov. 1999.

[5] 김성득, 나종범, "압축된 동영상에서의 AC 예측 기법을 이용한 DC 영상 추출기법," 대한전자공학회 학술대회 논문집, 제21권 2호, pp.867-870, 2005.

[6] N. Merhav and V. Bhaskaran, "Fast algorithms for DCT-domain image down-sampling and for inverse motion compensation," IEEE Trans. Circuits Syst. Video Technol., vol. 7, pp. 468-476, June 1997.

[7] Taskiran, C. Jau-Yuen Chen, Bouman, C.A. Delp, E.J. "A compressed video database structured for active browsing and search," IEEE image Processing, vol.3, pp.133-137, Oct. 1998.

[8] Jung-Rim Kim, Sungjoo Suh, Sanghoon Sull, "Fast scene change detection for personal video recorder," IEEE Trans. Consumer Electronics, vol. 49, no. 3, pp. 683-688, Aug. 2003.

[9] Ja-Cheon Yoon, Hyeokman Kim, Seong Soo Chun, Jung-Rim Kim, Sanghoon Sull, "Real-Time Video Indexing System for Live Digital Broadcast TV Programs," Lecture Notes in Computer Science, CVIR2004, vol. 3115, pp. 261-269, Jul. 2004.

[10] G.C. Feng and J. Jiang, "Image extraction in DCT domain," IEE Proc.-Vis. Image Signal Process., Vol. 150, No. 1, pp.20-27, Feb. 2003

[11] ITU-T Recommendation H.263. Video coding for low bitrate communication, Nov. 1995.

[12] ISO/IEC, JTC1/SC29/WG11, Coding of moving pictures and associated audio for digital storage media at up to about 1,5Mbits/s -- Part 2: Video, FDIS 11172-2, June 1996.

[13] ISO/IEC, JTC1/SC29/WG11, Information Technology-coding of audio-visual objects, Part 2: Video, FDIS 14496-2 Oct. 1998.

[14] ISO/IEC, JTC1/SC29/WG11 (MPEG), Information technology - Generic coding of moving pictures and associated audio information: Video, Inter-national Standard 13818-2, ISO/IEC, 2000.

[15] J. Song and B.L. Yeo, "Fast Extraction of Spatially Reduced Image Sequences from MPEG-2 Compressed Video," IEEE Trans. Circuits and Systems for Video Technology, vol. 9, no.7, pp. 1100-1114. Oct. 1999.

[16] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), 2003.

[17] Chen Chen, Ping-Hao Wu and Chen H, "Transform-Domain Intra Prediction for H.264," IEEE ISCAS, pp.1497-1500, May 2005.

[18] C. Chen, P.-H. Wu, and H. Chen, "MPEG-2 to H.264 transcoding," Picture Coding Symposium, Dec. 2004.

---

 저 자 소 개
 

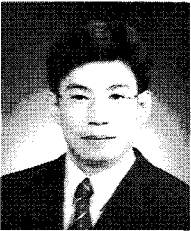
---

**유 상 준**

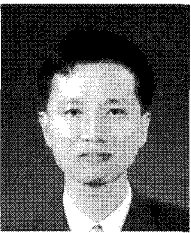
- 2003년 2월 광운대학교 전자공학과 공학사
- 2005년 2월 광운대학교 대학원 전자공학과 공학석사
- 2008년 2월 광운대학교 대학원 전자공학과 공학박사
- 주관심분야 : 영상신호처리, 멀티미디어 시스템, 병렬 신호처리

**윤 명 근**

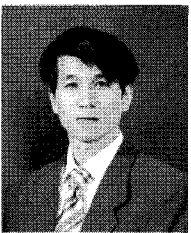
- 2007년 2월 : 광운대학교 전자공학과 졸업 (학사)
- 2007년 3월 ~ 현재 : 광운대학교 대학원 전자공학과 석사과정
- 주관심분야 : H.264/AVC, Thumbnail, 영상압축

**김 은 석**

- 1995년 2월 : 광운대학교 전자공학과 공학사
- 1997년 2월 : 광운대학교 전자공학과 공학석사
- 2007년 2월 : 광운대학교 전자공학과 공학박사
- 2003년 2월 ~ 현재 : 삼성전자, 디스플레이 사업부 근무
- 주관심분야 : HDTV, 영상압축

**손 채 봉**

- 1993년 2월 : 광운대학교 전자공학과 공학사
- 1995년 2월 : 광운대학교 대학원 전자공학과 공학석사
- 2006년 2월 : 광운대학교 대학원 전자공학과 공학박사
- 1991년 8월 ~ 1993년 2월 : 삼성전자 소프트웨어 멤버십 1기
- 2000년 3월 ~ 2006년 2월 : 한양여자대학 인터넷정보과 전임강사
- 2001년 3월 ~ 현재 : 주식회사 손랩 대표이사
- 2006년 3월 ~ 현재 : 광운대학교 전자통신공학과 조교수
- 주관심분야 : HDTV, UDTV, IPTV, DMB, DVB-H System

**심 동 규**

- 1993년 2월 : 서강대학교 전자공학과 공학사
- 1995년 2월 : 서강대학교 전자공학과 공학석사
- 1999년 2월 : 서강대학교 전자공학과 공학박사
- 2000년 9월 : 현대전자, 기반기술 연구소, 선임연구원
- 2002년 3월 : 바로비전, 뉴미디어 연구소, 선임연구원
- 2005년 2월 : Univ, Washington, Senior research engineer
- 2005년 3월 ~ 현재 : 광운대학교 전자정보대학, 컴퓨터 공학과 조교수
- 주관심분야 : 영상처리, 컴퓨터비전, 멀티미디어

---

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