

# 고속 썸네일 생성 방법에 대한 적응적 가중치를 적용한 오차 보상 방법

이경준<sup>o</sup>, 정제창

한양대학교 전자컴퓨터통신공학부

kjlee888@naver.com, jjeong@hanyang.ac.kr

## 요약

썸네일 이미지는 H.264/AVC 비트스트림 상에서 공간 영역으로 역변환을 거치지 않고 직접 추출이 가능하다. 하지만 부동소수점 연산 후 수행되는 반올림 과정으로 인하여 오차가 발생하며 다음 블록의 예측 과정에서 오차가 포함된 값을 이용하므로 이미지의 오른쪽 아래 방향으로 증폭되어 전파된다. 본 논문에서는 이러한 오차 전파를 고려한 오차 보상 방법을 제안한다. 제안하는 방법은 인트라 예측 모드별로 구한 오차 분포 데이터와 오차 전파 정도를 고려한 가중치 값을 이용하여 오차 보상값을 결정한다.

## 1. 서론

H.264/AVC는 HDTV, 인터넷 TV 또는 DMB와 같은 디지털 비디오 처리를 위하여 널리 사용되고 있다. 고해상도 비디오의 등장과 발전으로 인해 고효율의 영상처리 기법이 필요하게 되었으며, 비디오 인코딩 기술의 발전과 더불어 영상의 저장과 고속 탐색에 유리한 썸네일이라 불리는 원본의 축소된 크기의 이미지를 생성하는 방법들이 제안되어 왔다. 썸네일은 단순 축소 이미지이기 때문에 영상의 직관적인 정보를 제공해 줌으로써 비디오나 이미지의 검색이나 나열 등에 사용된다.

비트스트림 상의 주파수 영역으로 변환된 데이터로부터 직접 DC 계수를 추출하는 방법이 썸네일 추출 기본적인 방법 중 하나로 사용된다. 주파수 영역의 DC 계수는 한 블록의 대푯값으로 사용되며, 이러한 DC 계수의 집합을 통해 썸네일 이미지를 생성할 수 있다. Yeo와 Liu는 MPEG-1 프레임으로부터 DC 계수를 추출하는 방식을 이용하여 썸네일을 생성하는 방법을 제안하였다 [1]. 마찬가지로 MPEG-1/2의 비디오 스트림으로부터 다양한 크기의 썸네일 이미지를 더 적은 연산량을 통해 추출하는 방법이 제안되었다 [2]-[3].

H.264/AVC에서는 공간 영역에서 다양한 모드별로 블록을 예측하고 재구성하는 인트라 예측 과정을 제공한다 [4]. 기존의 썸네일 추출 방법은 인트라 프레임을 공간 영역으로 완전 복호화 한 후 추출하는 반면, 인트라 예측 과정의 분석을 통하여 완전 복호화 과정 없이 고속으로 썸네일을 추출하는 것이 가능하다. Chen은 인트라 예측 과정을 행렬 연산 식으로 표현하는 것이 가능함을 보였으며 이를 이용한 주파수 영역에서의 예측 방법을 제안하였다 [5]. Kim과 Yu는 Chen의 방법을 이용하여

비트스트림의 주파수 영역으로 변환된 데이터에서 바로 DC 계수를 계산하는 썸네일 추출 방법을 제안하였다 [6]-[7]. Kim은 썸네일 추출 과정의 곱셈 연산을 쉬프트 연산으로 대체한 고속 썸네일 추출 방법을 제안하였으며 [8], Yoon은 썸네일 이미지에 대한 오차 보상 방법을 제안하였다 [9].

본 논문에서는 Yoon이 [9]에서 제안한 방법을 기반으로 썸네일 추출 과정에서의 향상된 오차 보상 방법을 제안한다. 제안하는 알고리즘은 인트라 예측 모드별로 오차 분포 데이터를 수집하여 평균과 표준편차 값을 구한다. 그리고 예측 과정에서 오차 값을 참고하여 증폭되어 전파되는 오차를 고려하여 이미지 상의 위치에 적합한 가중치 값을 설정하여 각각 다른 보상값을 부여한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 고속 썸네일 알고리즘에 대해 간단히 소개를 하며 3장에서는 제안하는 알고리즘을 설명한다. 4장에서는 기존의 썸네일 추출 알고리즘과 본 논문에서 제안하는 알고리즘으로 진행된 실험결과를 비교 분석하고 5장에서 결론을 맺는다.

## 2. 기존 알고리즘

### 2.1 고속 썸네일 추출 알고리즘

Chen은 H.264/AVC의 인트라 예측에서 사용되는 9가지 모드에 대하여 행렬 연산으로 나타낼 수 있음을 보였다 [5]. 현재 블록에 대한 예측 블록  $m$ 는 다음과 같은 행렬식으로 계산할 수 있다.

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

식 (1)의  $s$ ,  $x_n$ 은 각각 인트라 예측 모드 번호, 쿼드트 연산 행렬, 예측 시 참조하는 이웃 블록을 나타내며  $c_{n,i}^m$ 는 인트라 예측 모드별로 필요한 상수 행렬을 나타낸다. 인트라 예측 시 참조하는 이웃 블록  $x_n$ 은 그림 1과 같다.

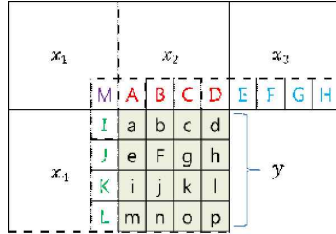


그림 1. 인트라 예측 시 참조되는 이웃블록

식 (3), (4)는 각각 식 (2)의 수직 및 수평 필터 행렬  $v$ ,  $u$ 를 이용하여 계산된 행렬을 나타낸다.

$$\begin{bmatrix} 0001 \\ 0001 \\ 0001 \\ 0001 \end{bmatrix}, h = \begin{bmatrix} 0000 \\ 0000 \\ 0000 \\ 1111 \end{bmatrix} \quad (2)$$

$$(a) = va = a \quad (3)$$

$$H(a) = ah = a_h \quad (4)$$

현재 블록은 다음 블록의 예측 시 이웃하는 블록으로써 참조되므로 미리 수직, 수평 필터 행렬을 이용하여 참조 값을 계산한다.

$$y_{p,v}^m = \left( \sum_{n=1}^3 \sum_{i=1}^4 V(s_i x_n c_{n,i}^m) \right) + \sum_{i=1}^4 V(c_{4,i}^m x_4 s_i) \quad (5)$$

$$y_{p,h}^m = \left( \sum_{n=1}^3 \sum_{i=1}^4 H(s_i x_n c_{n,i}^m) \right) + \sum_{i=1}^4 H(c_{4,i}^m x_4 s_i^T) \quad (6)$$

식 (5), (6)은 다음과 같이 간략화 시킬 수 있다.

$$\begin{aligned}
 y_{p,v}^m &= x_{1,v} c_{1,4}^m + x_{2,v} c_{2,4}^m + x_{3,v} c_{3,4}^m + (p_{4,v}^m x_{1,h})^T \\
 \text{where } p_{4,v}^m &= \sum_{i=1}^4 s_i V(c_{4,i}^m)
 \end{aligned} \quad (7)$$

$$\begin{aligned}
 y_{p,h}^m &= x_{1,v} q_{1,h}^m + x_{2,v} q_{2,h}^m + x_{3,v} q_{3,h}^m + c_{4,4}^m x_{4,h} \\
 \text{where } q_{4,h}^m &= \sum_{i=1}^4 (H(c_{n,i}^m) s_i^T)
 \end{aligned} \quad (8)$$

DC 계수는 행렬의 모든 성분을 행렬의 평균값으로 대체하는 UNI 연산을 이용하여 계산한다.

$$u = \begin{bmatrix} 11111 \\ 11111 \\ 11111 \\ 11111 \end{bmatrix} \quad (9)$$

$$U(a) = \frac{1}{N^2} uau \quad (10)$$

식 (9)와 (10)은 각각 필터  $u$ 와 이를 이용한 UNI 연산을 나타낸다. 이를 이용하여 현재 블록의 DC 계수를 계산하면 다음과 같다.

$$y_{p,uni}^m = \sum_{n=1}^3 x_{n,v} c_{n,i}^m + (c_4^m)^T x_{4,h} \quad (11)$$

$$\text{where } c_n^m = \frac{1}{16} \sum_{i=1}^4 c_{n,i}^m u$$

비트스트림 상의 블록 정보는 주파수 영역의 값이므로 식 (7), (8)과 (11) 또한 주파수 영역으로 변환된 식을 사용한다. H.264/AVC에서 사용하는 변형된 DCT를 이용하여 변환된  $y_{p,v}^m$ ,  $y_{p,h}^m$ ,  $y_{p,uni}^m$ 는 다음과 같이 나타낼 수 있다.

$$\begin{aligned}
 Y_{p,v}^m &= X_{1,v} C_{1,4}^m + X_{2,v} C_{2,4}^m + X_{3,v} C_{3,4}^m \\
 &+ (P_{4,v}^m X_{1,h})^T
 \end{aligned} \quad (12)$$

$$Y_{p,h}^m = \left( \sum_{n=1}^3 X_{n,v} Q_{n,h}^m \right)^T + C_{4,4}^m X_{4,h} \quad (13)$$

$$Y_{p,uni}^m = \sum_{n=1}^3 X_{n,v} C_n^m + (C_4^m)^T X_{4,h} \quad (14)$$

$Y_{p,uni}^m$ 와 비트스트림 상의 오차 정보  $Y_{error}$ 가 합하여진 행렬의 (0,0)위치의 값은 현재 블록의 DC 계수를 뜻하며 블록의 크기만큼 다운 스케일을 거쳐 썸네일 이미지의 픽셀 값으로 설정된다.

## 2.2 오차 보상 알고리즘

썸네일 추출 과정에서 사용하는 부동 소수점 연산은 H.264/AVC의 예측 과정, 변환 과정 및 양자화 과정에서 사용하는 정수 연산으로 인하여 반올림 오차를 생성한다. Yoon은 이러한 반올림 오차에서 발생하는 확률 분포를 이용한 확률 변수  $r$ 을 설정하여 오차 보상값을 결정하는 방법을 제안하였다

[9]. 보상값을  $r$ 에 대한 최소 분산값을 갖게 하는 값으로 정할 때  $s$ 는 다음과 같이 나타낼 수 있다.

$$s = \begin{bmatrix} [r_{0,0}] E[r_{0,1}] E[r_{0,2}] E[r_{0,3}] \\ E[r_{1,0}] E[r_{1,1}] E[r_{1,2}] E[r_{1,3}] \\ E[r_{2,0}] E[r_{2,1}] E[r_{2,2}] E[r_{2,3}] \\ E[r_{3,0}] E[r_{3,1}] E[r_{3,2}] E[r_{3,3}] \end{bmatrix} \quad (15)$$

DC 계수에 대하여 보상하는 대푯값은 오차 분포의 평균값을 이용하여 구할 수 있으며 식 (16)과 같이 나타낸다.

$$D^m = \begin{bmatrix} E[D_{0,0}] E[D_{0,1}] E[D_{0,2}] E[D_{0,3}] \\ E[D_{1,0}] E[D_{1,1}] E[D_{1,2}] E[D_{1,3}] \\ E[D_{2,0}] E[D_{2,1}] E[D_{2,2}] E[D_{2,3}] \\ E[D_{3,0}] E[D_{3,1}] E[D_{3,2}] E[D_{3,3}] \end{bmatrix} \quad (16)$$

확률 분포를 통하여 예측된 오차값 또한 현재 블록에 포함되어 다음 블록의 예측에 이용되므로 식 (3), (4)를 이용하여 수직, 수평 필터링을 거치며 썸네일 이미지를 생성한다.

### 3. 제안하는 알고리즘

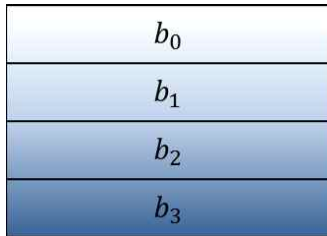


그림 2. 서로 다른 가중치 적용을 위해 분할된 이미지 영역

2장에서 살펴 본 바와 같이 썸네일 추출과정에서는 부동 소수점에 대한 반올림 연산을 수행함으로써 반올림 오차가 발생하게 되며, 순차적으로 현재 블록을 다음 블록의 예측 과정에 사용함으로써 오차의 크기가 증폭되는 문제가 발생한다. 따라서 썸네일 이미지의 화질 개선을 위해서 이러한 오차를 적절히 보상해 주는 과정이 필요하다.

일반적으로 썸네일 추출 과정에서 H.264/AVC의 각각의 인트라 예측 모드는 특정한 오차 분포 특성을 가진다. 따라서 이러한 특성을 이용하여 오차 보상값을 설정할 수 있다. 그림 2는 서로 다른 가중치를 적용하기 위해 4개의 영역으로 분할된 이미지를 나타낸다. 이미지의 상단 영역은 오차의 정도가 작으므로 작은 가중치를 이용하여 보상값을 결정하며 이미지 하단으로 갈수록 증폭되는 오차를 보상하기 위하여 큰 가중치를 설정한다. 인트라 예측 과정에서 사용되는 모드별 평균 및 표준편차 값

을 각각  $\mu^m, \sigma^m$ 이라 할 때,  $k$ 번째 분할 영역의 가중치는  $\omega_k^m$ 로 나타낸다. 마지막으로 이미지의  $(i, j)$ 에서의 보상값  $V_{(i,j)}$ 는 다음과 같이 결정된다.

$$V_{(i,j)} = \mu^m + \omega_k^m \sigma^m, \text{ if } i, j \in b_k \quad (17)$$

### 4. 실험 결과 및 분석

제안하는 방법은 다음과 같은 실험 환경을 통해 기존의 방법과 비교 분석 한다. 실험 이미지는 각각 다른 크기를 가지는 JPEG 이미지를 사용하였으며 이는 표 1과 같다.

표 1. 실험 이미지 및 크기

Test Image	Size
<i>Album1</i>	600×600
<i>Album2</i>	640×640
<i>Hirmer</i>	720×540
<i>Soccer</i>	900×656
<i>Beatles</i>	1024×768
<i>TVshow</i>	1280×720
<i>Flowershop</i>	4272×2848

썸네일 이미지는 세 가지 방법을 이용하여 생성한다. FD(Frequency domain)는 이미지 전체의 복호화 과정을 거쳐 DC 계수를 추출하며 method1은 [6]에서 제안하는 고속 썸네일 추출 방법을 이용한다. method2는 제안하는 방법을 적용한 오차가 보상된 썸네일을 생성한다. method1, method2는 각각 FD와 화질을 비교한다.

그림 3은 에 대한 원본 이미지이며 그림 4는 각각 FD, method1, method2를 통해 생성한 썸네일 이미지이다. 그림 4 (b)와 같이 method1을 이용하여 썸네일을 생성하였을 경우 FD를 이용하여 생성한



그림 3. Hirmer 원본 이미지



(a)



(b)



(c)

그림 4. 썸네일 결과 이미지 (a)FD, (b)method 1, (c)method 2

이미지에 비해 전체적인 밝기가 다소 어두우며, 이미지의 우측 하단부로 갈수록 그 정도가 더 크다. 이에 비하여 그림 4 (c)의 method2를 이용하여 생성한 썸네일 이미지의 경우 오차 보상을 통해 FD와 유사한 화질을 보인다.

표 2. 썸네일 이미지의 PSNR 비교 (dB)

Test Image	method1	method2	
<i>Album1</i>	34.24	35.56	+1.32
<i>Album2</i>	33.30	34.85	+1.56
<i>Hirmer</i>	33.69	34.76	+1.06
<i>Soccer</i>	33.74	35.26	+1.52
<i>Beatles</i>	32.45	34.61	+2.17
<i>TVshow</i>	34.45	35.18	+0.72
<i>Flowershop</i>	24.84	30.86	+6.02

표 2에서는 method1, method2를 통해 생성한 썸네일 이미지와 FD를 통해 생성한 썸네일 이미지 간의 PSNR 결과를 비교하였다. 실험에 사용한 모든 이미지들에 대하여 제안하는 방법인 method2가 method1에 비하여 높은 PSNR 결과를 보였다. *Flowershop*의 경우 이미지의 크기가 매우 크므로 우측 하단으로 갈수록 증폭되는 오차의 크기가 매우 커진다. 따라서 제안하는 방법을 통하여 보상되는 값 또한 크므로 높은 PSNR 향상 결과를 얻을 수 있었다.

### 5. 결론

본 논문에서는 썸네일 추출 과정에서 발생하는 반올림 오차에 대하여 인트라 예측 모드별로 얻은

평균과 표준편차를 이용하여 얻은 오차 보상값을 통해 화질을 개선하는 방법을 제안하였다. 기존의 방법과 비교하여 제안하는 방법은 각각의 인트라 예측 모드를 나누어 다룬다는 점과 이미지의 우측 및 하단으로 갈수록 증폭되는 오차를 적절히 보상할 수 있다는 이점이 있다. 비록 제안하는 방법이 이미지의 특성을 이용하므로 서로 다른 이미지에 대하여 특정 경우에 낮은 효율을 보일 수 있으나 가중치 값을 조정함으로써 이러한 문제를 완화시킬 수 있다.

### 감사의 글

본 연구는 미래창조과학부 및 정보통신산업진흥원에서 추진하는 "정보통신연구기반구축사업"으로 수행한 연구결과임(과제번호 I2221-14-1005)

### 참고문헌

- [1] B. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Video," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 5, no. 6, pp. 533-540, Dec. 1995.
- [2] June-Hwa Song and Boon-Lock Yeo, "Fast Extraction of Spatially Reduced Image Sequences from MPEG-2 Compressed Video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 7, pp. 1100-1114, Oct. 1999.
- [3] Sung-Joo Suh, Seong-Soo Chun, Myung-Hee Lee, and Sang-Hoon Sull, "Efficient Image Down-Conversion for Mixed Field/Frame-mode Macroblocks," *IEEE Electron. Lett.*, vol. 39(6), pp. 514-515, Mar. 2003.
- [4] ISO/IEC JTC1/SC29/WG11 (MPEG), "Coding of audio-visual objects-Part 10: Advanced Video Coding," *International Standard 14496-10*, ISO/IEC, 2004.
- [5] Chen Chen, Ping-Hao Wu, and Homer Chen, "Transform-Domain Intra Prediction for H.264," *IEEE ISCAS*, pp.1497-1500, May. 2005.
- [6] Eun-Seok Kim, Tae-Woong Um, and Seoung-Jun Oh, "A Fast Thumbnail Extraction Method in H.264/AVC Video Streams," *IEEE Trans. Consumer Electronics*, vol. 55, no. 3, pp. 1424-1430, Aug. 2009.
- [7] Sang-Jun Yu, Myung-Keun Yoon, Eun-Seok Kim, Chae-Bong Sohn, Dong-Gyu Sim, and Seoung-Jun Oh, "An Efficient Thumbnail Extraction Method in H.264/AVC Bitstreams," *THE KOREAN SOCIETY OF BROADCAST ENGINEERS*, vol. 13, no. 2, pp. 222-235, Mar. 2008.

- [8] Myoung-Hoon Kim, Hoon-Jae Lee, and Sang-Hoon Sull, "Fast Thumbnail Generation in Integer DCT Domain for H.264/AVC," *IEEE Trans. Consumer Electronics*, vol. 57, no. 2, pp.589-596, May. 2011.
- [9] Myung-Keun Yoon, Yeo-Song Lee, Chae-Bong Sohn, Ho-Chong Park, Chang-Beom Ahn, and Seoung-Jun Oh, "An Efficient Error Compensation Method for Thumbnail Extraction in H.264/AVC Bitstreams," *THE KOREAN SOCIETY OF BROADCAST ENGINEERS*, vol. 13, no. 5, pp. 622-635, Sep. 2008.