

# 근사 인코딩 기법을 이용한 H.264/AVC 비트율 제어 알고리즘

\*이록규 전광길 정제창

한양대학교

\*gunboy83@gmail.com

## A New Rate Control Scheme for H.264/AVC Video Using Pseudo Encoding Model

\*Lee, Rokkyu Jeon, Gwanggil Jeong, Jechang

Hanyang University

### 요약

본 논문에서는 근사 인코딩 기법을 이용한 H.264/AVC 비디오 코덱에서의 비트율 제어 알고리즘을 제안한다. H.264는 기존의 동영상 압축 표준보다 월등한 압축 성능을 나타내지만, 구조적 복잡성으로 인해 비트율 제어 측면에서는 과거에 제안된 H.264를 위한 비트율 제어 알고리즘들의 성능은 기대에 미치지 못하였다. 제안된 알고리즘은 근사 인코딩 기법을 사용하여 실제 H.264 인코딩이 이루어지기 이전에 향후 발생될 인코딩 비트를 미리 예측할 수 있고, 비트율 제어에서 매우 높은 중요성을 차지하는 프레임의 복잡도 예측에서 우수한 성능을 나타낸다. 알고리즘의 연산량 측면에서도 제안된 근사 인코딩 기법은 간단한 구조로 이루어져 있어 장점을 나타낸다. 본 논문에서는 DCT 영역에서의 각 프레임의 zero의 개수를 분석하여 얻어낸 영상의 특성을 비트율 제어에 활용한다. 실험결과는 제안된 알고리즘이 H.264 레퍼런스 소프트웨어의 가장 최신 버전인 JMI2.2 환경에서 기존의 알고리즘에 비해 우수한 성능을 나타낸다는 것을 알 수 있다

### 1. 서론

최근 동영상의 전송량이 급격히 증가하는 반면 그것을 전송하기 위한 채널의 용량은 충분하지 못함에 따라서 동영상 압축 기술의 중요성이 최근 더욱 부각되고 있다[1]. 특히, 노트북과 PDA, PMP와 같은 휴대용 단말기의 사용이 많아지면서 무선 채널을 이용하는 경우의 제한된 대역폭으로 인해 고효율의 압축 기술과 가변적인 채널의 특성을 고려한 고성능 비트율 제어 기법이 요구된다. H.264/AVC는 기존의 압축 기술에 비해 월등한 압축효율을 발휘함과 동시에 최근 가장 널리 사용되고 있는 압축표준이다[2]. 비트율 제어 알고리즘은 전송채널의 용량을 고려하여 변화하는 인코딩 비트의 특성을 일정하게 유지하기 위해 인코더에 포함되어 사용되며, 이것은 각각의 동영상 프레임의 복잡도를 예측함으로써 가능하다[3].

그러나 H.264의 구조적 특성으로 인해 효율적인 비트율 제어 알고리즘을 고안하는 데는 어려움이 있다[4]. H.264의 rate distortion optimization (RDO) 프로세스[5]는 잘 알려진 chicken and egg dilemma를 야기하여[6], 기존의 많은 비트율 제어 알고리즘들이 사용해 온 mean absolute difference (MAD)를 비트율 제어에 사용하는데 어려움이 있기 때문이다. MAD는 동영상에 포함된 각각의 프레임의 복잡도를 예측하기 위한 가장 대표적인 수단이다.

MAD를 비트율 제어에 활용하기 위해서는 RDO 프로세스 이전에 인코딩중인 해당 프레임의 MAD값이 필요하지만, H.264에서는 RDO 프로세스 이후에 MAD 값이 얻어지기 때문에 기존의 다른 코덱에서와 같은 방법으로 사용하기는 불가능하다.

이러한 문제를 해결하기 위해서 H.264 비트율 제어를 위한 MAD 이외의 새로운 개념이 요구된다. 제안하는 알고리즘에서는 DCT영역에서의 각 프레임의 zero의 개수를 이용하여 전반적인 비트율 제어 성능을 향상시키는데 초점을 맞춘다.

본 논문에서 제안하는 근사 인코딩 기법은 크게 DCT와 양자화 단계로 구성된다. 실제 H.264 인코딩 프로세스가 진행되기 이전에 개별적으로 8X8 DCT와 양자화 과정을 수행함으로써 각각의 프레임의 인코딩 관점에서의 복잡도를 미리 예측한다. 근사 인코딩 과정에서 사용되는 양자화 계수 값은 모든 프레임에서 30으로 동일하다. 실제 H.264가 사용하는 4X4 DCT를 이용하지 않고 8X8 DCT를 사용하는 이유는 DCT블록의 크기가 어느 정도 커야 변화하는 zero의 개수를 좀 더 분명하게 관찰할 수 있기 때문이다. 만일 DCT영역에서 한 프레임 내의 zero의 개수가 비교적 많다면 이것은 인코딩 관점에서 해당 프레임이 비교적 낮은 복잡도를 갖는다고 판단할 수 있는 근거가 된다. 반면에 비교적 적은 수의 zero의 개수는 비교적 높은 복잡도를 나타낸다고 볼 수 있다.

이와 같이 zero에 대한 정보를 비트율 제어에 이용함으로써 H.264에서의 MAD 이용의 한계를 보완할 수 있다. 근사 인코딩 기법과 기존의 MAD 예측기법을 병합함으로써 본 논문에서 제안한 새로운 프레임 복잡도 측정기준인  $\Phi(j)$ 을 구하게 된다.  $\Phi(j)$ 값은 제안할 버퍼 모델에 의하여 추가적으로 최적화 과정을 거치게 되고, 최적화된 파라미터인  $\Phi_{opt}(j)$ 을 이용하여 최종적으로 출력 비트율을 효과적으로 제어할 수 있다. 실험 결과에서는 제안된 비트율 제어 기법이 MAD값을 프레임

\* 본 연구보고서는 정보통신부 출연금으로 ETRI, SoC산업진흥센터에서 수행한 IT SoC 핵심설계인력양성사업의 연구결과입니다.

복잡도 측정 기준으로 사용하는 JM12.2의 비트율 제어 알고리즘에 비해 우수한 성능을 나타냄을 확인할 수 있다.

## 2. 근사 인코딩 기법을 이용한 비트율 제어

본 장에서는 근사 인코딩 기법을 이용한 H.264에서의 비트율 제어 기법을 설명한다. 근사 인코딩을 통해 실제 H.264 인코딩이 이루어지기 이전에 발생할 결과 영상의 용량을 예측할 수 있게 된다. 그림1에 나타난 바와 같이, 근사 인코딩 알고리즘은 DCT 영역에서의 프레임 복잡도 예측과 간결한 버퍼모델, 그리고 양자화 계수 값 결정 프로세스를 포함한다.

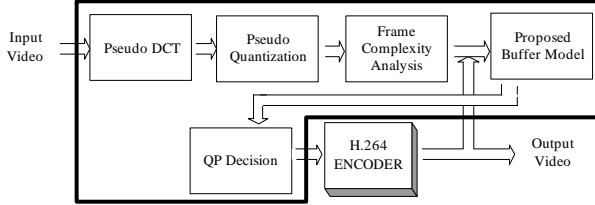


그림 1. 근사 인코딩 알고리즘의 구조

### 가. DCT 영역에서의 프레임 복잡도 분석 방법

인코더에서 압축하려는 프레임의 복잡도를 예측하기 위해서는 실제 인코딩 프로세스가 진행되기 이전에 제안된 근사 인코딩 과정이 먼저 수행되어야 한다. 근사 인코딩 과정에서는 8X8 DCT와 양자화 단계를 통해 DCT 영역에서의 각각의 프레임 내의 zero의 개수가 얻어진다. 각각의 프레임에서 발생하는 zero의 개수의 변화를 분석함으로써, 인코딩 하려는 프레임의 복잡도를 구할 수 있다. 예를 들어, 만일 특정 프레임이 비교적 많은 수의 zero를 DCT영역에서 갖는다면 해당 프레임은 비교적 낮은 복잡도를 갖는다고 판단할 수 있다. 이러한 경우에 해당프레임을 인코딩함으로써 발생할 비트의 양은 다른 프레임에 비해서 적기 때문에, 양자화 계수 값을 작게 조절함으로써 발생할 비트의 양을 목표치에 근접하도록 조절할 필요성이 생긴다.

현재 프레임  $j$ 의 zero개수와 이전의 모든 프레임들의 zero 개수의 평균의 비율은  $R(Z_j)$ 로 정의하며 아래와 같이 나타낼 수 있다.

$$R(Z_j) = \begin{cases} 1, & \text{if } j = 1 \\ \frac{Z_j}{\sum_{i=1}^{j-1} Z_i} \times (j-1), & \text{if } j = 2, 3, \dots, N_{total} \end{cases} \quad (1)$$

여기서,  $Z_j$ 는 현재 프레임  $j$ 의 DCT 영역에서의 zero 개수를 나타내고,  $N_{total}$ 은 인코딩할 전체 프레임의 수이다. 결과적으로,  $R(Z_j)$ 는 동영상 시퀀스들의 평균적인 복잡도와 비교했을 때의 현재 프레임의 복잡도를 나타낸다.

프레임의 복잡도를 예측하기 위한 추가적인 파라미터인  $R(MAD_j)$ 는 인코딩할 현재 프레임  $j$ 의 MAD값과 기존의 모든 프레임의 평균 MAD값의 비율로 정의되고<sup>[7]</sup>, 아래와 같이 나타낸다.

$$R(MAD_j) = \begin{cases} 1, & \text{if } j = 1 \\ \frac{MAD_j}{\sum_{i=1}^{j-1} MAD_i} \times (j-1), & \text{if } j = 2, 3, \dots, N_{total} \end{cases} \quad (2)$$

여기서,  $MAD_j$ 는 현재 프레임  $j$ 의 MAD 값을 나타내며, H.264에서 RDO 프로세스 이후에 얻어지는 MAD 값이 아닌 인코딩 이전에 현재 프레임과 이전 프레임의 차이를 계산함으로써 구한 것이다. 얻어진  $R(Z_j)$ 값과  $R(MAD_j)$ 값을 조합하여, 프레임 복잡도 측정 파라미터  $\Phi(j)$

를 다음과 같이 구할 수 있다.

$$\Phi(j) = \alpha \cdot R(MAD_j) + (1 - \alpha) \cdot R(Z_j) \quad (3)$$

여기서,  $\alpha$ 는 가중치 계수로서 이론적으로 값을 유추해내기에는 매우 어려우므로 실험적인 방법을 통해 구해야 한다. 본 논문에서는 0.7을  $\alpha$ 값으로 사용하였다. 프레임 복잡도 측정 파라미터  $\Phi(j)$ 는 다음 절에서 제안하는 버퍼 모델을 통해 최적화 과정을 거쳐게 된다.

### 나. 버퍼모델

전송채널이 갖는 대역폭 특성에 대응하기 위해 비트율 제어 알고리즘은 각각의 프레임의 예측된 인코딩 비트수와 실제 발생하는 인코딩 비트수가 일치하는지 확인하여 관리해야 한다.

본 논문에서 제안하는 버퍼 모델에서는 각각의 프레임에 할당할 비트수를 계산하게 되며, 현재 인코딩할 프레임  $j$ 에 할당될 비트인  $\tau_{reserved}(j)$ 는 다음과 같이 계산된다.

$$\tau_{reserved}(j) = \frac{u - \sum_{i=1}^{j-1} \tau_{actual}(i)}{N_{total} - (j-1)} \quad (4)$$

여기서,  $u$ 는 입력 영상에 대한 목표 대역폭이고,  $\tau_{actual}(i)$ 는  $i$ 번째 프레임의 실제 발생한 인코딩 비트수이다.  $\tau_{reserved}(j)$ 는 목표 대역폭과 이미 사용된 비트수를 고려하여 남아있는 비트수와 앞으로 인코딩해야 할 프레임수를 고려하여 얻어진다.

$\tau_{reserved}(j)$ 를 이용해 제안된 버퍼 모델의 파라미터  $R(Buffer)$ 를 다음과 같이 구할 수 있다.

$$R(Buffer) = \begin{cases} 1, & \text{if } j = 1 \\ \beta \cdot \frac{\tau_{actual}(j-1)}{\tau_{reserved}(j-1)}, & \text{if } j = 2, 3, \dots, N_{total} \end{cases} \quad (5)$$

여기서,  $\beta$ 는 증폭계수로서 첫번째 P 프레임에서는 0.138에서 0.145의 값을 갖고, 나머지 P 프레임에서는 1의 값을 갖는다.  $R(Buffer)$ 는 이전 프레임의 실제 발생 비트수에 대한 할당 비트수의 비율로 정의되고 I 프레임과 P 프레임의 특성이 고려되어야 한다.

그림 2에서 보는 바와 같이 JM12.2에서 I 프레임은 P 프레임에 비해 인코딩을 위해 보통 더 많은 비트가 필요하기 때문에  $\beta$ 는 첫번째 P 프레임에 대해서는 특수한 값을 가져야 한다. 결과적으로  $R(Buffer)$ 는 인코딩 비트의 변동을 제어하는 피드백 파라미터로서 작용하게 된다.

얻어진  $R(Buffer)$ 값과  $\Phi(j)$ 값을 조합함으로써, 최적화된 비트율 제어 파라미터  $\Phi_{opt}(j)$ 가 식 (6)과 같이 얻어진다.

$$\Phi_{opt}(j) = \theta \cdot R(Buffer) + (1 - \theta) \eta \cdot \Phi(j) \quad (6)$$

실험적인 방법에 의해서, 제안된 알고리즘은 가중치 계수  $\theta$ 가 0.2로 설정되었을때 최적의 성능을 나타냄을 알 수 있고, 가중치 계수  $\eta$ 는 0.98로 설정하였다.

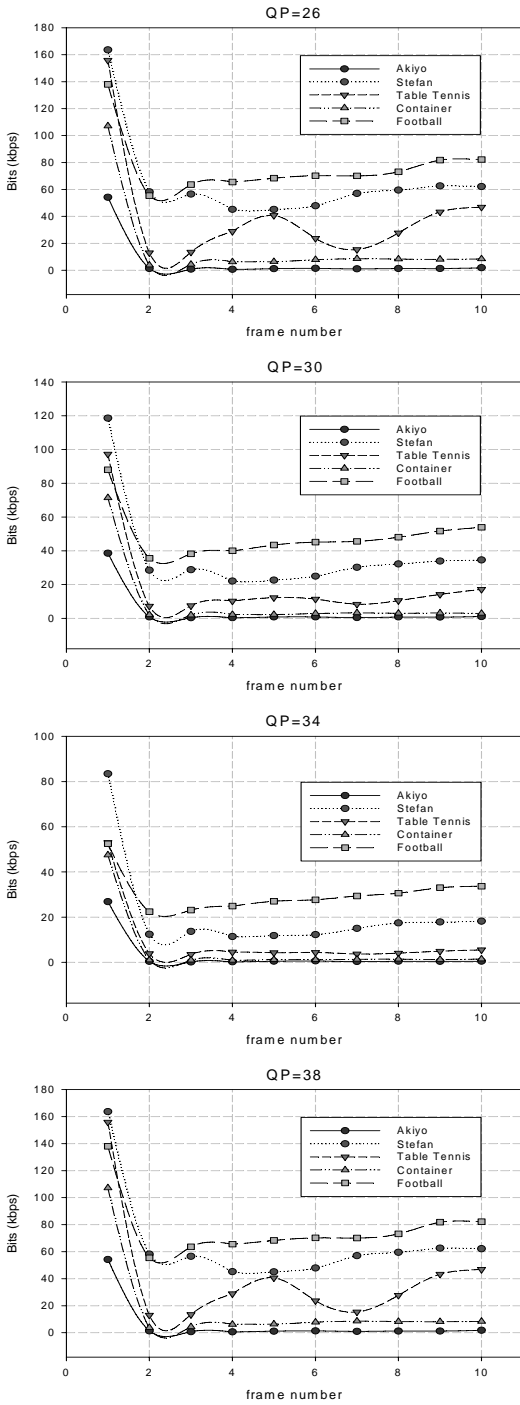


그림 2. JM12.2에서의 I와 P 프레임의 용량 비교

#### 다. 양자화 계수값의 결정

이 단계에서는 구해진 최적화된 비트율 제어 파라미터  $\Phi_{opt}(j)$ 를 이용해서 양자화 계수값을 조절하는 과정을 설명한다. 그 과정은 (7)에 나타난 바와 같이 5가지 경우에 대해 QP값을 변화시키면서 비트율 제어를 수행하게 된다.

각각의 문턱값들은 다음과 같이 설명된다: *big upper bound* ( $k_{BUB}$ ); *small upper bound* ( $k_{SUB}$ ); *small lower bound* ( $k_{SLB}$ ); and *big lower bound* ( $k_{BLB}$ ). 이 값들은 실험적인 방법에 의해 구해지며, 실험결과에 자세한 값들을 나타내었다.

$$\begin{aligned}
 & \text{if } (k_{BUB} \leq \Phi_{opt}(j)), \\
 & \quad QP_j = QP_{j-1} + 2 \\
 & \text{else if } (k_{SUB} \leq \Phi_{opt}(j) < k_{BUB}), \\
 & \quad QP_j = QP_{j-1} + 1 \\
 & \text{else if } (k_{SLB} \leq \Phi_{opt}(j) < k_{SUB}), \\
 & \quad QP_j = QP_{j-1} \\
 & \text{else if } (k_{BLB} \leq \Phi_{opt}(j) < k_{SLB}), \\
 & \quad QP_j = QP_{j-1} - 1 \\
 & \text{otherwise,} \\
 & \quad QP_j = QP_{j-1} - 2
 \end{aligned} \tag{7}$$

사용자 관점의 주관적 화질 측면에서 급격한 화질 변화를 방지하기 위해, 인접한 프레임 간의 양자화 계수값의 변화는 최대 2를 넘지 않도록 정하였다.

### 3. 실험결과

본 논문에서는 JM12.2를 비교대상으로 선정하여 실험을 진행하였다. 2.3절에서 소개한 문턱 값들은 다음 표 1과 같다.

표 1. 할당된 문턱 값

Threshold	Value
$k_{BUB}$	1.10
$k_{SUB}$	1.03
$k_{SLB}$	0.97
$k_{BLB}$	0.91

사용된 모든 테스트 영상의 크기는 QCIF이고, 시퀀스의 타입은 IPPP이고, QP의 초기값은 30이고, P슬라이스에 대한 참조영상의 수는 5장이고, motion estimation은 full search기법이 사용되었고, 영상의 길이는 90프레임을 사용하였다.

제안된 알고리즘과 JM12.2의 성능 비교는 동일한 비트율에서의 두 알고리즘의 화질을 비교함으로써 수행되었다. 영상의 화질을 객관적으로 비교하기 위해서 peak signal-to-noise ratio (PSNR) 값이 사용되었다. 식 (8)은 PSNR의 정의이고, 여기서  $MAX_i$ 는 이미지에 있는 픽셀이 가질 수 있는 최대값이다.

$$PSNR = 10 \log_{10} \left( \frac{MAX_i^2}{MSE} \right) \tag{8}$$

표 2에 나타난 바와 같이 제안된 알고리즘은 MAD만을 프레임의 복잡도 측정 기준으로 삼는 JM12.2의 비트율 제어 알고리즘에 비해 우수한 PSNR 화질을 보여준다. 실험결과를 통해 근사 인코딩 기법이 우수한 프레임 복잡도 예측 성능을 발휘하고, 다양한 종류의 입력 영상에 대해 제한된 채널 대역폭 환경에서 우수한 화질을 나타내는 것을 확인할 수 있다. 특히 저비트율에서의 채널환경에 높은 성능을 나타냄을 확인할 수 있다.

### 4. 결론

본 논문에서는 근사 인코딩 기법을 이용한 H.264/AVC에서의 새로운 비트율 제어 알고리즘을 제안하였다. 근사 인코딩 기법은 비교적 적은 계산량으로 우수한 프레임 복잡도 측정 성능을 제공하고, JM12.2의 비트율 제어 알고리즘과 비교해서 같은 비트율에서 더 좋은 화질을 나타낸다.

한편, 본 알고리즘의 한계점은 H.264 압축에서 실제 영상 데이터 비

트열의 제어에만 초점을 맞추었다는 점이다. H.264의 경우에는 인코딩된 데이터에서 헤더 부분이 차지하는 용량이 실제 영상 데이터 못지않게 큰 경우가 많다. 앞으로의 연구에서 이러한 헤더 정보에 대한 비트율 제어까지 통합적으로 할 수 있는 기술을 추가하여 본 알고리즘의 단점을 극복할 수 있을 것이다.

### 참 고 문 헌

- [1] Z. He and S. Mitra, "A Unified Rate-Distortion Analysis Framework for Transform Coding," IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, pp. 1221-1236, December 2001.
- [2] G. Sullivan et al, "Editors' Text for ISO/IEC 14496-10:2005 (AVD 3rd Edition)," International Organisation for Standardisation. Busan, Korea. 2005.
- [3] J. Yang et al, "A Joint Rate Control Scheme for H.264 Encoding of Multiple Video Sequences," In IEEE Transactions on Consumer Electronics, vol. 51, pp. 617-623, May 2005.
- [4] S. Ma et al, "Rate control for advance video coding (AVC) standard," Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on. Bangkok, Thailand, pp. II892-II895. 2003.
- [5] T. Wiegand et al, "Rate-Constrained Coder Control and Comparison of Video Coding Standards," In IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, pp. 688-703. July 2003.
- [6] Z. G. Li et al, "Adaptive rate control for H.264," IEEE International Conference on Image Processing. Singapore, pp. 745-748, 2004.
- [7] M. Jiang et al, "Frame-layer bit allocation scheme for constant quality video," IEEE International Conference on Multimedia and Expo. Taipei, Taiwan, pp. 1055-1058, 2004.

표 2. 제안한 비트율 제어 알고리즘과 JM12.2의 PSNR(dB) 성능 비교

Sequences	Frame rate	Target bitrate	JM 12.2		Proposed		PSNR Gain
			Bitrate	Avg. PSNR	Bitrate	Avg. PSNR	
News	30	64kbps	63.50kbps	36.14dB	62.57kbps	36.49dB	+0.35dB
Container	30	64kbps	64.31kbps	37.48dB	64.27kbps	37.73dB	+0.25dB
Salesman	30	64kbps	64.34kbps	35.22dB	64.14kbps	35.40dB	+0.18dB
Salesman	30	128kbps	127.57kbps	38.91dB	128.39kbps	38.97dB	+0.06dB
Mobile	30	64kbps	63.90kbps	24.71dB	64.02kbps	25.18dB	+0.47dB
Mobile	30	128kbps	127.83kbps	28.41dB	127.57kbps	28.45dB	+0.04dB
Carphone	30	32kbps	32.28kbps	31.56dB	32.00kbps	31.88dB	+0.32dB