

## Arithmetic Coding을 통한 Enhanced aacPlus의 비트율 감소

\*구자성 \*함우규 \*김기준 \*\*강경옥 \*박호중

\*광운대학교 \*\*한국전자통신연구원

\*koo16@kw.ac.kr

## Bit Rate Reduction of Enhanced aacPlus by Arithmetic Coding

\*Ku, Ja-Seong \*Ham, Woo-Gyu \*Kim, Ki-Jun \*\*Kang, Kyeongok \*Park, Hochong

\*Kwangwoon University \*\*Electronics and Telecommunications Research Institute

## 요약

본 논문에서는 enhanced aacPlus 부호화기의 스펙트럼 계수 무손실 부호화에 arithmetic coding을 적용하여 비트율을 감소시키는 방법을 연구하였다. USAC의 arithmetic coding을 enhanced aacPlus 구조에 맞게 변경하여 적용하였다. 기존 방법과 arithmetic coding 방법에 의한 부호화 비트 수를 비교하여 성능을 평가하였고, 모노 신호에서 최대 9.3%, 스테레오 신호에서 최대 6.6%의 비트 감소율을 확인하였다.

## 1. 서론

다양한 오디오 서비스를 위한 다양한 특성의 오디오 부호화기가 개발되었으며, 대표적으로 MP3, AAC (advanced audio coder), enhanced aacPlus, HE-AAC v2 (high efficiency AAC version 2) 등이 있다 [1-4]. Enhanced aacPlus는 AAC, PS (parametric stereo), SBR (spectral band replication) 기능을 포함하여 매우 높은 압축률을 제공한다.

Enhanced aacPlus는 AAC와 동일하게 스펙트럼 계수의 무손실 부호화에서 Huffman coding을 사용하며, 다수의 Huffman codebook을 사용하여 신호의 특성에 따라 효율적인 부호화를 진행한다. 최근에 개발된 USAC (unified speech and audio coding)에서는 무손실 부호화의 성능 향상을 위해 arithmetic coding을 사용하며, 기존 Huffman coding에 비하여 향상된 압축률을 제공한다 [5]. 특히, CABAC (context-adaptive binary arithmetic coding) 구조의 부호화를 통하여 신호의 context에 따른 최적의 부호화를 구현하여 매우 높은 압축률을 제공한다 [6, 7].

본 논문에서는 USAC arithmetic coding을 enhanced aacPlus에 적용하여 비트율을 감소시키는 실험을 진행하였다. 이를 위해 CABAC 동작을 분석하고 enhanced aacPlus 구조에 맞도록 동작을 조정하여 최종 변경된 enhanced aacPlus의 encoder와 decoder를 구현하였다. 변경된 enhanced aacPlus와 기존 enhanced aacPlus가 동일한 양자화된 스펙트럼 계수에 대하여 동일한 decoder 출력을 제공하는 것을 확인하였고, 이 때 전송 비트 수가 감소하는 정도를 측정하여 비트 감소율을 검증하였다.

## 2. Arithmetic Coding 동작

USAC arithmetic coding은 양자화된 MDCT 계수 두개를 하나로 묶어 2-tuple (a, b)의 형태로 부호화 한다. 또한, 과거의 2-tuple 부호화 결과를 context 정보로 사용한다. 그림 1은 MDCT 계수가 2-tuple이 되는 과정이다. MDCT 계수를 2-tuple로 묶게 되면 MDCT 계수가 반으로 줄어 Long window에서는 512개, Short window에서는 64개의 2-tuple이 생성된다.

그림 2는 2-tuple로 새롭게 정의된 MDCT 계수를 time-frequency 축으로 재배치하여 context로 사용하는 모습을 도식화 한 것이다. Context-adaptive arithmetic coding은 현재 부호화하려는 2-tuple에서 time-frequency축으로 가장 가까운 2-tuple을 context로 사용한다.

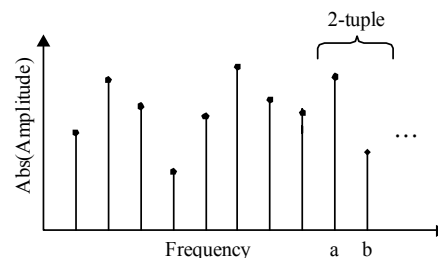


그림 1. MDCT 계수가 2-tuple이 되는 과정.

Fig. 1. Generation of 2-tuple of MDCT coefficients

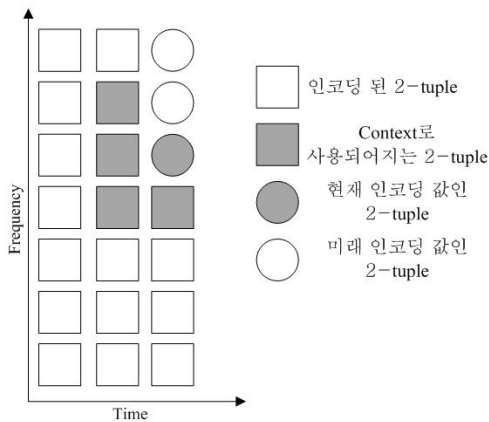


그림 2. Context-adaptive arithmetic coding에서 context 생성 과정  
 Fig. 2. Generation of context in context-adaptive arithmetic coding.

Long window와 Short window의 2-tuple 개수가 다르기 때문에 window switching시 context 활용에 문제가 발생한다. 이를 위해 Long window의 context를 Short window의 주파수에 맞게 축소하고, Short window의 context는 Long window에 맞게 확장해야 한다. 그림 3은 window별 2-tuple을 확장 및 축소를 하는 방법을 도식화 한 것이다. 그림 3 (a)는 Long window에서 Short window로 2-tuple이 축소되는 과정이다. 512개의 Long window 2-tuple을 64개의 Short window 2-tuple로 만들어 주기 위해 Long window 2-tuple에서 8의 배수인 값을 사용하여 Short window 2-tuple을 만든다. 그림 3 (b)는 Short window에서 Long window로 2-tuple이 확장되는 과정이다. 64개의 Short window 2-tuple을 512개의 Long window 2-tuple로 확장하기 위해 각각의 Short window 2-tuple을 8번씩 복사한다. 위 2-tuple 변환 과정을 통해 window switching시 서로 같은 주파수의 context를 사용하여 arithmetic coding을 한다.

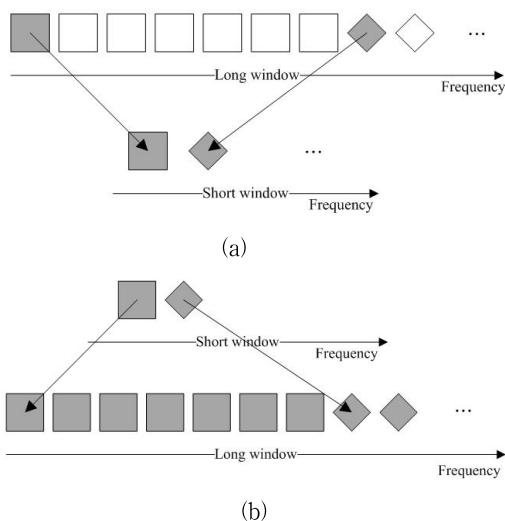


그림 3. Window 별 context 변환 방법. (a) Long window에서 Short window로 switching. (b) Short window에서 Long window로 switching.  
 Fig. 3. Context conversion in window switching. (a) From Long window to Short window. (b) From Short window to Long window.

Bit rate에 따라 복호화기에 전송되는 MDCT 개수도 변하는데, 기존 Huffman coding시 전송되는 MDCT를 그대로 arithmetic coding에 적용하면 에러가 발생된다. 이는 window switching에서 Long window MDCT 길이가 Short window MDCT 길이의 8배가 되어야 정확한 context 확장 및 축소를 하는데, 기존 Huffman coding이 사용하는 window 별 MDCT 길이가 이를 만족시키지 못하기 때문이다.

표 1은 기존 Huffman coding시 복호화기에 전달하는 MDCT 개수와 Arithmetic coding을 하기 위해 변경한 MDCT 개수를 나타낸 것이다. 표 1에 변경된 MDCT 개수는 Long window와 Short window의 MDCT 개수가 8배 차이이다. 변경된 MDCT 개수는 기존 MDCT 개수보다 약간 많지만 arithmetic coding 비트율에는 크게 영향을 미치지 않는다.

표 1. Bit rate 별 MDCT 계수의 수.

Table 1. The number of MDCT coefficients for each bit rate.

Bit Rate (kbps)	Huffman coding시 MDCT 계수		Arithmetic coding시 MDCT 계수	
	Short window	Long window	Short window	Long window
16 ~ 24	64	432	64	512
24 ~ 32	64	552	76	608
32 ~ 44	76	600	76	608
44 ~	108	896	112	896

MDCT 계수 그룹화는 8개의 Short window MDCT 계수를 같은 주파수 대역별로 묶는 과정이다. 기존 Huffman coding을 사용할 때, 8개 Short window의 128개 MDCT 계수를 그룹화 하여 bit rate에 맞게 복호화기에 전달한다. 그러나 기존 Short window에서 사용하는 그룹화 된 MDCT 계수를 arithmetic coding에 그대로 적용할 수 없는데, 그룹화 된 MDCT 계수는 window switching시 2-tuple의 주파수가 섞여 있으므로 context arithmetic coding을 할 수 없기 때문이다. 따라서 arithmetic coding 동작에서는 그룹화 하지 않은 Short window MDCT 계수를 사용한다.

### 3. 성능 평가

위에서 설명한 구조에 따라 USAC arithmetic coding을 enhanced aacPlus에 적용하여 encoder와 decoder를 구현하였고, 동작 검증과 성능 검증을 진행하였다. 이 때 사용한 오디오 입력 신호의 규격은 다음과 같다.

- 채널 : 모노, 스테레오
- 샘플링 주파수 : 48kHz
- 비트 해상도 : 16 bit/sample
- 길이 : 570 초

동작 검증을 위해 기존 encoder와 변경된 encoder가 주어진 비트율과 입력에 대하여 MDCT 계수 양자화까지 동일한 동작을 수행하도록 하고, 각각 Huffman coding과 arithmetic coding을 수행하고 bit-stream을 출력하도록 하였다. 다음, 기존 decoder와 변경된 decoder를 수행하여 각각 출력을 구하고, 두 출력이 동일한 것을 확인하였다. 이를 통하여 arithmetic coding을 포함한 encoder와 decoder가 정상 동작하는 것을 검증하였다.

그림 4는 동일한 입력에 대하여 기존 Huffman coding의 전체 bit count와 arithmetic coding의 전체 bit count를 비교하여, 비트수가 감소한 감소율(%)을 보여준다. Huffman coding의 bit count는 Huffman table index와 section 분리를 위한 bit를 포함한다. 모노 신호에서는 bit rate에 따라 최소 5.3%에서 최대 9.3%의 비트 감소율을 확인할 수 있고, 스테레오 신호에서는 최소 5.7%에서 최대 6.6%의 비트 감소율을 확인할 수 있었다. 이를 통하여 arithmetic coding이 Huffman coding보다 압축률이 높다는 것을 확인할 수 있다.

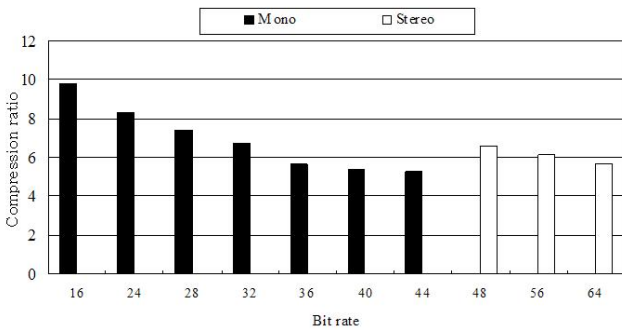


그림 4. Bit rate별 무손실 부호화의 비트 감소율.  
Fig. 4. Bit reduction ratio in lossless coding for each bit rate.

Arithmetic coding은 Huffman coding에 비해 많은 연산량을 요구한다. 표 2는 새로 적용한 arithmetic coding과 기존 Huffman coding을 사용하는 부호화기의 계산량을 각각 측정된 결과를 보여준다. 계산량 측정 실험은 Intel i5 @ 3.30GHz, 4GB RAM 환경에서 진행하였다. 표 2와 같이 45.56초에서 46.68초로 기존 대비 약 2.4%로 계산량이 증가하였지만, 오디오 부호화기의 활용에 큰 영향을 미치지 않는다.

표 2. 부호화기 Huffman coding과 Arithmetic coding의 계산량.  
Table 2. Computational complexity of Huffman coding and arithmetic coding

부호화	Huffman Coding	Arithmetic Coding
시간	45.56초	46.68초

#### 4. 결론

본 논문에서는 USAC arithmetic coding을 enhanced aacPlus에 적용하여 enhanced aacPlus의 비트 수를 감소시키는 실험을 진행하였다. USAC arithmetic coding의 동작을 분석하고, 기존 Huffman coding과의 차이점을 확인하고, 이를 해결하여 enhanced aacPlus에 적용하였다. 이를 통하여 arithmetic coding이 적용된 encoder와 decoder를 구현하였고, 동작 검증과 성능 검증을 진행하였다. 부호화 비트율에 따라 비트 감소율에서 성능 차이를 가지며, 모노 신호에 대하여 최대 9.3%의 비트 감소율을 얻고, 스테레오 신호에 대하여 최대 6.6%의 비트 감소율을 얻는 것을 확인하였다.

#### 감사의 글

본 연구는 미래창조과학부가 지원한 2013년 정보통신·방송(ICT) 연구개발사업의 연구결과로 수행되었습니다.

#### 참고문헌

- [1] ISO/IEC 14496-3, "Coding of Audio-Visual Objects - Part 3 : Audio, Subpart 4 : General Audio Coding - AAC, TwinVQ, BSAC," 2001.
- [2] 3GPP TS 26.401, "Enhanced aacPlus General Audio Codec : General Description," 2004.
- [3] ISO/IEC 14496-3, AMD. 1, "Bandwidth Extension," 2003.
- [4] ISO/IEC 14496-3, AMD. 2, "Parametric Coding of High Quality Audio," 2004.
- [5] ISO/IEC FDIS 23003-3:2011, "Information technology Part 3: Unified Speech and audio coding," International Standards Organization, 2011
- [6] V. Singla, R. Singla, and S. Gupta, "Data compression modeling: Huffman and Arithmetic," International Journal of the computer, the Internet and Management, vol. 16, no. 3, pp. 64-68, September-December, 2008
- [7] A. Shahbaharni, R. Bahrampour, M. Sabbaghi Rostami, and M. Ayoubi Mobarhan, "Evaluation of Huffman and Arithmetic Algorithms for Multimedia Compression Standards," International Journal of Computer Science, Engineering and Applications, vol 1, no. 4, August 2011.