

HEVC 코덱 기반 고속 병렬 UHD 디코더 개발

김영만, 한재일
국민대학교

요약

본고에서는 서버 단에서 동작하는 UHD급 코덱 기술에 대한 소개와 병렬 UHD 디코더 개발 현황을 소개한다. 먼저 UHD급 코덱의 국제 표준화 동향을 살펴보고 2013년 4월 표준으로 제정된 8K UHD급 코덱 표준인 HEVC의 핵심 요소 기술에 대하여 기술한다. 마지막으로 고속 UHD 디코더 개발 현황에 대하여 설명한다.

I. 서론

2003년에 표준이 제정된 H.264[1]는 비디오 압축 성능이 우수하여 수많은 미디어 서비스에서 애용되어 왔다. 한편 최근 4K(3840×2160), 8K(7680×4320) UHD 영상과 같은 초고품질의 영상미디어에 대한 요구사항에 맞추어 2009년부터 ISO/IEC Moving Picture Experts Group(MPEG)과 ITU-T Video Coding Experts Group(VCEG)의 두 미디어 표준화 그룹은 공동으로 High Efficiency Video Coding(HEVC)[2]이라는 명칭의 새로운 영상압축 표준을 개발하여 왔으며 2013년 4월에 이르러 두 표준 기관은 각각 ISO/IEC 23008-2 MPEG-H Part 2와 ITU-T H.265라는 명칭으로 HEVC 표준을 확정하였다. HEVC의 자세한 내용은 마지막 Draft(2013년 1월 Draft 10[3])가 공개되어 있어 누구나 자유롭게 열람할 수 있다.

성능평가에 관련된 연구에서 임의의 화면을 동일한 화질 수준으로 압축하는 경우 HEVC는 H.264[4][5][6]에 비하여 약 2배의 압축률을 갖고 있어 UHD 영상과 같이 대용량 데이터가 발생하는 미디어 스트림을 네트워크로 전송하고 목적지 단말을 통하여 재생하는데 있어 H.264 보다 더 적합하다.

UHD 미디어는 화면 크기가 기존 HDTV 영상(1280×720)에 비하여 한 프레임이 차지하는 픽셀 수가 최대 36배로 늘어나기 때문에 이를 실시간으로 압축, 재생하기 위해서 높은 계

산 능력을 가진 연산장치가 필요하다. 이와 같은 대용량 미디어를 처리하기 위하여 멀티코어 CPU와 매니코어 GPU를 갖춘 서버나 단말에서 병렬 프로그래밍 기법을 사용하는 것이 합리적인 방법이며 HEVC 표준에서도 병렬 프로그래밍이 수월하게 이루어질 수 있도록 3가지 프레임 구성 방식(slice, tile, wavefront(WPP))[7]을 지원한다.

HEVC는 미디어 데이터 흐름에서 병렬처리하기 어려운 H.264 알고리즘의 구조를 보다 손쉽게 프로그램 병렬화 할 수 있도록 개선하고 있으나 arithmetic coding 모듈을 통한 압축과 복원 구조가 갖는 직렬구조로 인하여 병렬 프로그래밍하기 어려우며 압축률을 높이기 위하여 화면 상황을 세분화하는 작업 때문에 if 문을 필요로 하는 프로그램 구현으로 인하여 GPU 사용 효율이 많이 떨어지게 된다.

본 고에서는 우선 8K UHD급 코덱 표준인 HEVC의 핵심 요소에 대해서 살펴보고, 이어서 고속 UHD 디코더 개발 현황에 대하여 기술한다.

II. HEVC 핵심 요소

본 장에서는 8K UHD급 코덱 표준인 HEVC의 핵심 요소에 대해서 살펴보고자 한다. <그림 1>은 HEVC 인코더의 내부 구조를 나타내고 있다.

<그림 1>에서 보듯이 왼쪽에서 들어오는 입력 비디오 신호에 대하여 프레임 별로 압축을 위한 데이터 처리를 시작한다. 먼저 현 프레임을 64×64 사이즈의 Code Tree Unit(CTU)들로 나누어 각각의 CTU에 대하여 필요하면 작은 사이즈의 블록으로 다시 분할한다. 이렇게 분할한 최종 블록은 목적에 따라서 Prediction Unit(PU) 혹은 Transform Unit(TU)로 일컫는데 PU가 TU와 같은 사이즈일 필요는 없다.

각각의 PU에 대하여 시간적으로 인접한 참조화면에서 현 PU와 가장 유사한 블록을 찾아서 PU와 이 블록 사이의 공간적인 오차를 모션 벡터로 부르고 <그림 1>의 오른쪽에 있는 CABAC

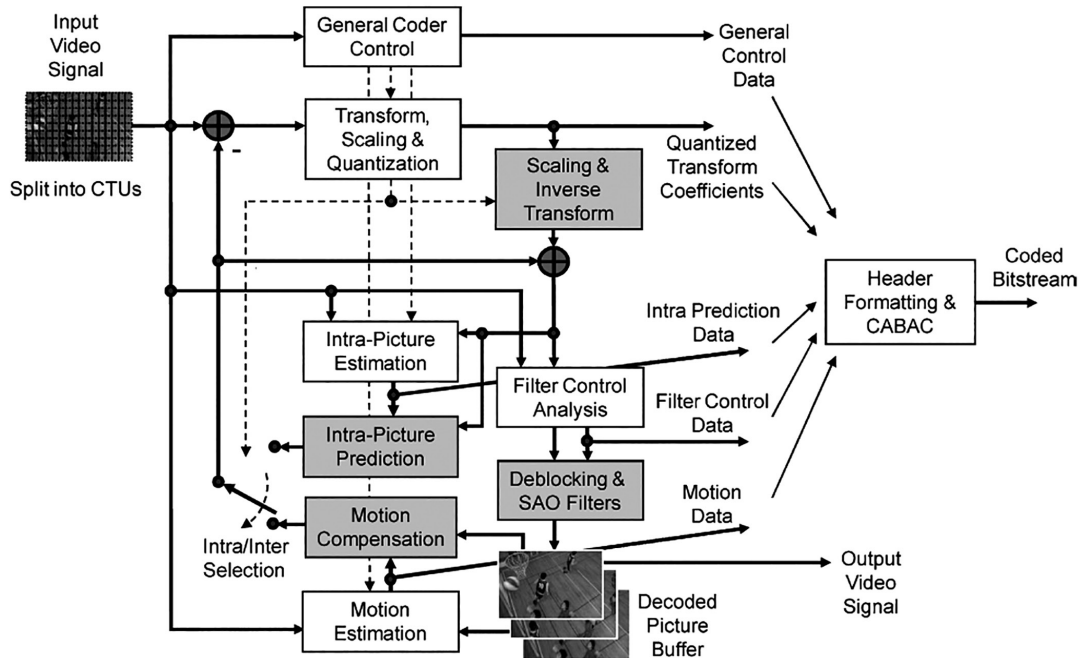


그림 1. HEVC 인코더 내부 구조도

모듈에 보내어 벡터 값을 코딩한 후에 비트 스트림으로 파일에 저장하거나 네트워크를 지나 디코더로 보내져서 재생된다. 한편 위에서 찾은 유사블록과 PU 간의 차이를 계산하여 잔여블록을 구한 후 변환 모듈(Transform, Scaling & Quantization)로 보내어 다음 처리를 하게 되는데 변환 모듈에 입력된 값이 0에 가까울수록 모듈에 의해 처리된 출력 데이터가 작은 사이즈로 줄어드는 효과가 있다. 따라서 위의 과정에서 PU와 흡사한 유사블록을 찾는 과정이 최적화 될 필요가 있으며 이 때 사용되는 과정을 Inter Prediction 혹은 Motion Compensation 방식이라고 부른다.

Inter Prediction 방식을 대체하는 방법으로 Intra Prediction 방법(Intra-Picture Prediction)이 있는데 PU와 유사한 블록을 동일 프레임 상의 다른 위치에서 찾기 때문에 Inter Prediction 방식에 비하여 두 블록 간의 유사성이 떨어지며 최종 압축률도 Inter Prediction 방식에 비해 많이 부족한 결과를 보여준다.

두가지 Prediction 방식에 의해 생성된 잔여 영상 블록이 변환 모듈을 통과하게 되면 Discrete Cosine Transform(DCT) 처리에 의해서 0에 가까운 값이 더 많이 발생하게 되며, 고주파 영역의 값들이 작은 경우 Quantization 처리를 통하여 삭제해도 복원 화면에 별로 영향을 주지 않는 성질을 이용하여 잔여 영상 블록의 일부를 데이터 처리 대상에서 제외함으로써 추가적인 압축효과를 달성한다.

변환 모듈에서 출력된 데이터는 마지막으로 CABAC[8] 모듈을 통하여 arithmetic coding 처리를 받게 되어 statistical

data redundancy 제거에 의한 압축을 보태어 HEVC 본연의 고압축률을 실현하게 된다.

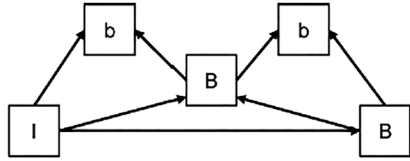
Inter Prediction은 현 프레임에 시간적으로 인접한 화면들을 참조화면으로 사용하는데, 변환 모듈에서 출력된 데이터를 역변환 모듈(Scaling & Inverse Transform)에 투입하여 잔여블록을 재현한 후 Inter/Intra Prediction 과정에서 선택한 유사블록을 더하여 PU를 복원하고 영상 버퍼(Decoded Picture Buffer(DPB))에 저장하여 참조화면으로 사용한다. 이 때 복원된 PU는 원 PU와 약간의 차이가 존재하며 이는 변환 모듈이 사용하는 DCT 알고리즘이 원 데이터 정보를 일부 유실하는 lossy compression 방식에 속하기 때문이다.

DPB에 저장된 프레임들은 디코더에서 실제로 영상재생에 사용되기 때문에 프레임에서 블록 경계에 존재하는 인위적인 결을 제거하여 재생 화면의 질을 높이는 필터링(Deblocking & SAO Filters)[9] 작업을 수행한 후에 저장하게 된다.

〈그림 1〉에서 회색으로 칠해진 모듈들은 프레임 복원을 위하여 디코더에서 사용된다. 다음은 HEVC 프레임 구성 요소와 각 모듈에 대한 상세한 설명이다.

1. Inter Prediction을 위한 프레임 참조

〈그림 2〉는 HEVC 인코더가 처리하는 프레임 순서의 일례를 보여주고 있다. Picture Order Count(POC)는 프레임이 캡처되는 순서를 나타내는 값으로 그림에서는 5장의 프레임



POC	0	1	2	3	4
Decoding order	1	4	3	5	2
RPS	-	[0,2]	[0,4]	[2,4]	[0]

그림 2. HEVC 프레임 참조 관계

이 왼쪽에서 오른쪽으로 가면서 순차적으로 찍힌 것을 나타낸다. 한편 HEVC 인코더에서 프레임이 처리되는 순서는 촬영 순서와 다르게 진행되는데 <그림 2>에서는 0, 4, 2, 1, 3 번 프레임의 순서로 처리된다. 0번 프레임은 I-Frame으로서 Intra Prediction 방식으로만 처리되므로 주변 참조 프레임셋(Reference Picture Set(RPS))을 사용하지 않는다.

4번 프레임은 두 번째로 처리되는데 0번 만을 참조 프레임으로 사용하여 Inter Prediction 방식으로 예측하기 때문에 P-Frame으로 불리워지며 RPS 값은[0]을 갖고 있다. 2번 프레임은 세 번째로 처리되며 0번, 4번 프레임, 즉 2개의 프레임을 참조로 하여 예측하므로 B-Frame으로 분류되고 있으며 RPS는[0,4]이다.

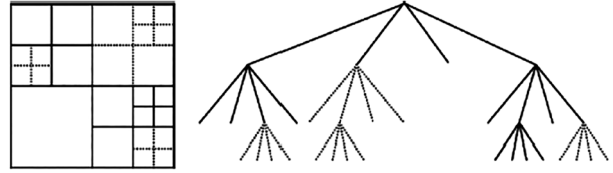


그림 3. CTU, PU, TU 블록 구조

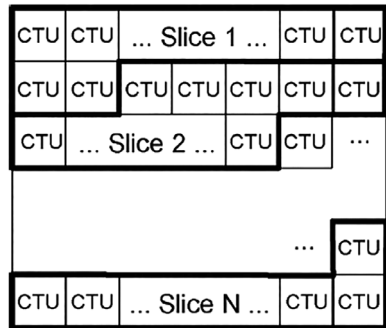
2. CTU, PU, TU 블록 구조

CTU는 프레임을 구성하는 최상위층의 정사각형 블록으로서 <그림 3>과 같이 내부가 분할되며 각 소형 블록이 예측의 기본 단위로 사용되면 PU(Prediction Unit), 변환의 단위로 사용되면 TU(Transform Unit)로 불리워진다. 각 블록은 트리 구조로 관리되고 있다.

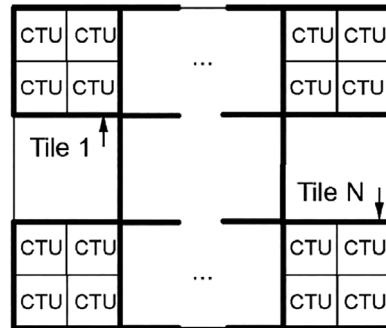
프레임을 이루는 각 픽셀은 3가지 성분으로 이루어진다. 예를 들면 YUV 포맷으로 이루어진 프레임의 경우, 픽셀은 Y(밝기), U(색깔1), V(색깔2) 성분으로 구성된다. 이 때 각 색깔을 대표하는 CTU, PU, TU의 해당 유닛을 CTB, PB, TB라고 일컫는다.

3. 프레임 분할 구조

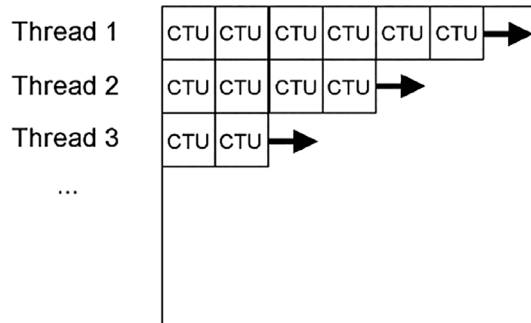
<그림 4>는 하나의 프레임이 분할되는 3가지 구조를 보여준다. (a)는 slice 구조로서 전체 화면을 slice 단위로 분할한 결과를 보여준다. 이 때, 각 slice의 최대 크기를 CTU 블록 갯수, 바



(a)



(b)



(c)

그림 4. 프레임 분할 구조

한 비디오 영상에 대한 실시간 처리 기능을 검증할 예정이다.

Acknowledgement

본 연구는 미래창조과학부 ‘범부처 Giga KOREA 사업’의 일환으로 수행하였음. [GK13P0100, Giga Media 기반 Tele-Experience 서비스 SW플랫폼 기술 개발]

참고 문헌

- [1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard and A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 560–576, 2003
- [2] “ISO/IEC 23008–2:2013,” International Organization for Standardization, 2013–11–25, Retrieved 2013–11–29.
- [3] B. Bross, W. J. Han, G. J. Sullivan, J. R. Ohm and T. Wiegand, “High Efficiency Video Coding (HEVC) Text Specification Draft 10,” JCTVC–L1003, ITU–T/ISO/IEC Joint Collaborative Team on Video Coding (JCT–VC), Jan. 2013
- [4] S. Wenger, “H.264/AVC over IP,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 645–656, 2003
- [5] T. Stockhammer, M. M. Hannuksela and T. Wiegand, “H.264/AVC in wireless environments,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 657–673, 2003
- [6] H. Schwarz, D. Marpe and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 17, pp. 1103–1120, 2007
- [7] G. J. Sullivan, J. R. Ohm, Woo–Jin Han, T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) Standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1649–1668, 2012
- [8] J. Sole, et al, “Transform Coefficient Coding in HEVC,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1765–1777, 2012
- [9] Chih–Ming Fu, et al, “Sample Adaptive Offset in the HEVC Standard,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1755–1764, 2012
- [10] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan and T. Wiegand, “Comparison of the coding efficiency of video coding standards – Including High Efficiency Video Coding(HEVC),” IEEE Trans. Circuits Syst. Video Technol., vol. 22, pp. 1668–1683, 2012
- [11] F. Bossen, B. Bross, K. Sühring and D. Flynn, “HEVC complexity and implementation analysis,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, pp. 1684–1695, 2012
- [12] F. Bossen, D. Flynn, K. Sühring, “HEVC HM 10 Reference Software,” JCTVC–L1010, ITU–T/ISO/IEC Joint Collaborative Team on Video Coding (JCT–VC), Jan. 2013
- [13] FFmpeg, <http://ffmpeg.org/>

약 력



김 영 만

1980년 서울대학교 기계공학과 졸업
 1982년 KAIST 기계공학과 석사
 1983년~1986년 LG전자 선임연구원
 1987년~1992년 The Ohio State University, 전산학, Ph.D.
 1996년~현재 국민대학교 컴퓨터공학부 교수
 관심 분야: 에너지 절감 운영체제, HEVC 병렬처리, 컴퓨터 네트워크, 미들웨어



한 재 일

1980년 연세대학교 수학과 졸업
 1986년 Syracuse University 컴퓨터과학 석사
 1992년 Syracuse University 컴퓨터과학 박사
 1993년~1994년 한국전자통신연구원 선임연구원
 1995년~현재 국민대학교 컴퓨터공학부 교수
 관심분야: 분산처리, 클라우드 시스템, 미들웨어, 객체지향 시스템, 지능형 시스템, 빅데이터, 서비스 컴퓨팅