HEVC 영상 압축 기술 소개

ㅁ 한우진 / 가천대학교 소프트웨어설계경영학과

요약

현재 ITU-T Video Coding Experts Group(VCEG)과 ISO/IEC Moving Picture Experts Group(MPEG)에서는 고효율 영상 압축 기술(High-Efficiency Video Coding; HEVC) 표준화에 대한 막바지 작업을 진행하고 있다. 본 원고에서는 기존 H.264/AVC 대비 동일 주관적 화질 대비 압축률 2배를 목표로 하고 있는 HEVC 영상 압축 기술에 대해, 그 배경 및 주요 구성 요소 기술에 대해 설명하고, 표준화 진행 과정에 대해서 간략하게 소개하고자 한다.

Ⅰ. 배경

주관적 화질 손실을 최소화하면서 막대한 영상 데이터의 양을 크게 감소시키기 위해 사용되는 영상 압축 기술은 주로 ITU-T와 ISO/IEC로 대표되는 국제 표준화 기구들을 통해 제정된 표준들에 의해 기술 발전이 이루어져왔다. ITU-T에서는 화상 통화를 위한 영상 압축 기술인 H.261[1], H.263[2] 등을 발표하였으며, MPEG에서는 광기기에 저장하기 위한 영상 압축 기술인 MPEG-1[3], MPEG-4 Visual[4]를 발표하였다. 또한, 두 기구에서 공동으로 작업하여 발표한 표준으로, H.262/MPEG-2 Video[5] 및 H.264/MPEG-4 AVC[6]가 있다.

최신 영상 압축 표준인 H.264/MPEG-4 AVC는 ITU-T와 ISO/IEC에서 공동으로 설립한 단체인 Joint Video Team(JVT)을 통해 1999년부터 2003년까지 표준화가 진행되었으며, 그 후 신축적 영상 압축 기술(scalable video coding), 다시점 영상 압축 기술(multi-view video coding)을 지원하도록 2003년에서 2009년까지 그 확장 작업이 이루어졌다. H.264/MPEG-4 AVC는 HD video streaming, digital camcorders, internet network video, mobile video, real-time video conversion 등 H.262/MPEG-2 Video의 압축 성능 한계로 인해 만

족스럽게 구현될 수 없었던 다양한 응용 분야를 현실화 시킴으로써 그 사용 폭이 크게 확대되고 있다.

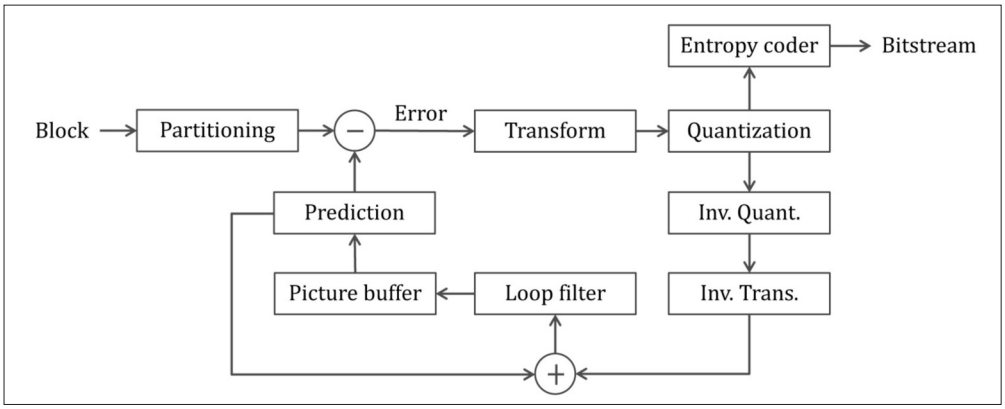
하지만, 최근 H.264/MPEG-4 AVC의 압축 효율을 더욱 향상시켜야만 할 필요성이 여러 측면에서 관측되고 있다. 스마트폰, 태블릿 등 모바일 기기에서도 HD 1080p 이상의 HD 영상에 대한 재생 및 녹화를 지원하는 등 HD 영상이 날로 보편화되고 있는 것은 물론, 이를 넘어 4k×2k, 8k×4k 등 HD 1080p 이상의 초고해상도를 지원하는 디스플레이 장치 및 영상 획득 장치들이 속속 등장하고 있다. 또한, 3D 고화질 영상 서비스 및 다시점 영상 서비스와 같이 영상 데이터의 양을 크게 증가시키는 서비스들이 점차 일반화될 것으로 예측된다. 한편 세계 최대의 영상 공유 사이트인 YouTube는 매분당 약 48시간의 영상 데이터가 업로드되고 있으며, 이와 관련하여 Cisco에서 2011년 예측한 자료에 의하면 이미 전세계 인터넷 트래픽의 50% 이상이 영상 서비스에 의한 것이며, 이 수치는 향후 3년 이내에 90%에 달할 것이라고 한다.

HEVC는 이러한 요구 사항에 맞추어, H.264/MPEG-4 AVC에 비해 그 압축 효율을 비약적으로 향상시키는 것을 목표로 만들어진 영상 압축 기술이다. H.262/MPEG-2 Video나 H.264/MPEG-4 AVC와 같이 ITU-T VCEG과 ISO/IEC MPEG의 공동 프로젝트이며, 이를 위해 설립된 JCT-VC (Joint Collaborative Team on Video Coding)라는 공동 기구에서 표준화가 진행되고 있다. HEVC는 기존 H.264/MPEG-4 AVC가 적용되던 모든 분야에 적용 가능하지만, 특히 고해상도 영상에 대한 압축 효율 향상에 주안점을 두고 설계되었으며, 구현 측면에서도 최근 기술 발전에 맞추어 병렬화에 용이하도록 만들기 위한 기술들을 다수 포함하고 있다.

본 원고의 제 2장에서는 HEVC를 구성하는 주요 기술들의 특징들을 H.264/MPEG-4 AVC와 비교하여 설명하며, 제 3장에서는 HEVC의 압축 성능 분석 결과를 소개한다. 제 4장에서는 HEVC 표준화의 진행 과정을 간략하게 소개한 후, 제 5장에서 결론을 맺는다.

II. 주요 기술 구성

HEVC는 H.261부터 사용되어 오던 것과 같이,



〈그림 1〉 Block-based hybrid coding (블록 기반 하이브리드 코딩 구조)

영상의 각 picture를 블록 구조로 분할한 후, 화면 간/화면 내 예측 기술과 변환 기술을 함께 사용하는 〈그림 1〉의 블록 기반 하이브리드 코딩 구조를 그대로 사용하고 있지만, 그 각각의 요소 기술을 크게 발전시킴으로써 압축 효율을 향상시킨다. 제 2장에서는, 최신 압축 표준인 H.264/MPEG-4 AVC에 비해 HEVC에서 변화된 부분들을 중심으로 주요 요소 기술들을 소개한다. 현재 정의된 HEVC 표준안에 대한 완전한 설명은 [7]을 참고하면 된다.

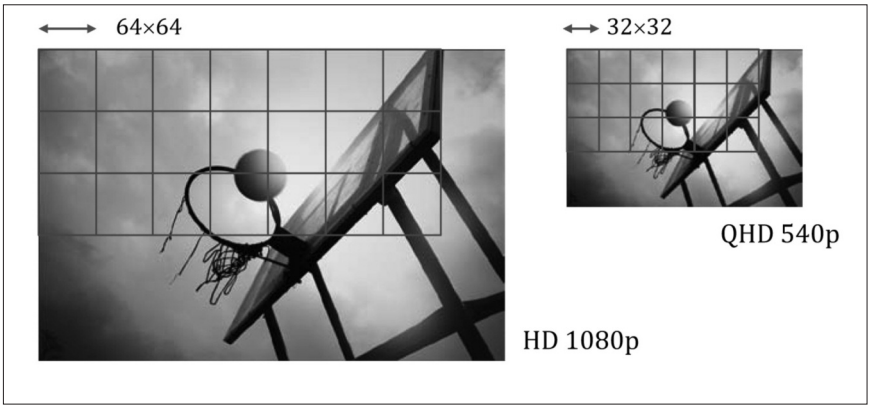
1. CTU (Coding Tree Unit) structure

H.264/MPEG-4 AVC 등 기존 영상 압축 표준들은 입력 영상의 각 picture를 매크로블록이라고 불리는 고정된 크기의 기본 블록 단위로 분할한 후 이후 과정을 수행한다. 이 매크로블록은 일반적으로 16×16 크기로 고정되어 있는데, HEVC는 이러한 고정 크기 분할 방법 대신, 다양한 크기의 분할 방법을 사용함으로써 그 유연성을 높이고 있다. 매크로블록에 대응되는 기본 블록 단위로서, HEVC에서는 CTU(Coding Tree Unit)라고 하는 단위를 사용하는데, CTU의 크기는 16×16 크기부터 64×64 크기까지 가변적으로 부호화기에서 선택 부호화할 수 있도록 한다. 〈그림 2〉는 HD 1080p와 QHD 540p의 두 가지 영상 해상도에 대해 각각 64×64와 32×32의 CTU 크기를 적용한 예이다.

일반적으로 고해상도 영상의 경우 큰 크기의 CTU가 압축 성능을 향상시키는데 유효한 것으로 알려져 있지만, 다양한 영상 해상도에 대해 압축 효율, 메모리 요구 사항, 부호화기/복호화기 총 지연 시간 등 여러 측면에서의 최적화가 가능하도록 하기 위해, HEVC에서는 고정된 크기의 CTU를 사용하는 대신 가변적인 크기의 CTU를 표준안에서 정의하고 있다.

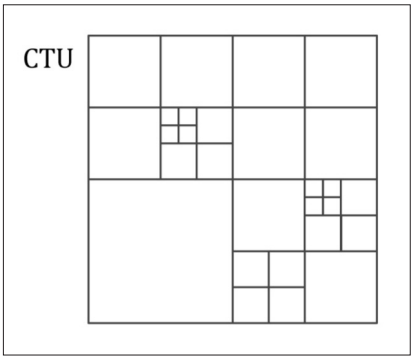
2. CU (Coding Unit) structure

기본 블록 단위인 CTU가 정의되고 나면, 하나의 CTU는 다시 예측 방법의 특성에 따라 다수의 CU(Coding Unit)들로 분할될 수 있다. 분할된 CU들 마다 화면 간 예측, 혹은 화면 내 예측 중 한 가지 방법이 선택되어 사용되는데, CTU의 크기가 매



〈그림 2〉 Various CTU sizes for different spatial resolution (영상 해상도에 따른 다양한 CTU 크기)

크로블록에 비해 일반적으로 훨씬 크며, 가변적인 크기를 갖는 것이 가능하므로 일반적인 쿼드트리 구조를 사용하여 CTU를 복수 개의 CU들로 분할함으로써 일관성 있는 비트스트림 구조를 갖도록 하고 있다. H.264/MPEG-4 AVC의 경우 화면 간 예측과 화면 내 예측 단위가 고정 크기를 갖는 매크로블록 별로 구별되는 데 비해, HEVC는 쿼드트리 형태로 분할된 CU 별로 이러한 구별이 이루어지는 특징이 있으며, CU는 8×8 크기에서 64×64 크기까지 가변 될 수 있어 더욱 정교한 예측 방법 구별을 가능하게 한다. 〈그림 3〉은 하나의 CTU를 쿼드트리 형태로 분할하여 복수 개의 CU들을 얻은 예이다.

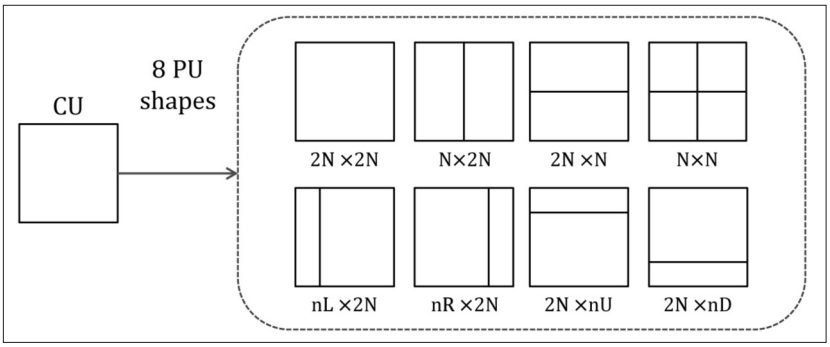


〈그림 3〉 CU partitioning from quadtree split of CTU (CTU의 쿼드트리 분할을 이용한 CU 분할)

3. PU (Prediction Unit) structure

화면 내 예측 혹은 화면 간 예측을 사용하는 하나의 블록에 있어, 그 압축 성능을 향상시키기 위해 가변 크기를 갖는 복수 개의 블록들로 재분할하는 기술이 일반적으로 사용된다. 하나의 CU로부터 재분할된 각 블록들은 화면 내 예측 혹은 화면 간 예측 중 한 가지 방법을 공유하지만 각각 독립된 예측 파라미터들을 갖는 특징이 있다. 구체적으로, 화면 간 예측의 경우, HEVC에서는 〈그림 4〉와 같이 H.264/MPEG-4 AVC에서 사용되는 4가지 분할 패턴을 확장함으로써 총 8개의 분할 패턴을 사용하고, 각각의 분할된 단위 마다 독립된 최적 움직임 벡터를 갖는 것을 허용함으로써 그 압축 효율을 향상시키고 있다. HEVC에서는 CU로부터 분할된 이러한 블록들을 PU(Prediction Unit)라고 정의한다.

한편, PU분할 방법 중 N×N 분할 방법은 CU를 한 단계 분할하는 것과 개념적으로 중복되기 때문에, CU가 최대한 분할되었을 경우에만 적용 가능하도록 한정하고 있으며, 부호화기의 복잡도를 줄이기 위해 비대칭 모양을 갖는 nL×2N, nR×2N, 2N×nU, 2N×nD 분할 방법은 CU가 8×8크기일 때는 적용되지 않는다. 또한, 요구 대역폭을 최소화



〈그림 4〉 PU shapes specified in HEVC (HEVC 표준에 정의된 PU 분할 모양들)

하기 위해, 화면 간 예측 방법에서는 4×4 PU를 허용하지 않으며, 4×8 PU 및 8×4 PU에서는 양방향 화면 간 예측 방법을 허용하지 않는다.

4. TU (Transform Unit) structure

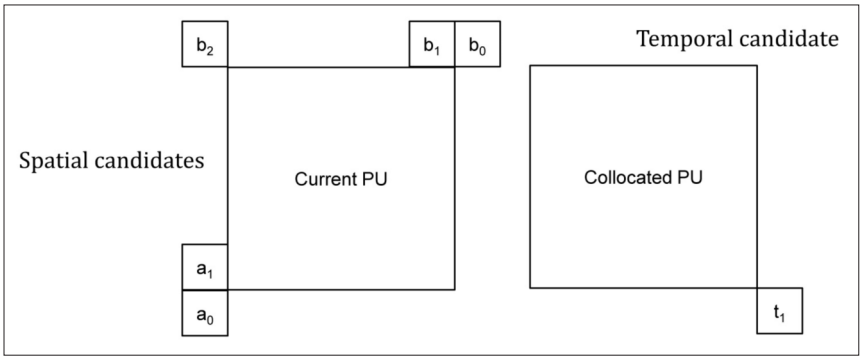
HEVC에서는 일반적인 hybrid coding 방식과 마찬가지로 예측 과정이 완료되고 난 후, 원 신호와 예측 신호간의 차이인 잔차 신호를 생성한 후, 이 신호에 2차원 변환을 적용한다. 하지만 H.264/MPEG-4 AVC 등 기존의 표준들이 일반적으로 예측 신호를 생성할 때의 블록 크기와 변환을 적용할 때의 블록 크기를 일치시키는 데 비해, HEVC는 주어진 크기의 잔차 신호 블록을 쿼드트리 형태로 재분할한 후, 각각의 분할된 영역에 대해 변환들을 적용할 수 있도록 허용함으로써, 압축 효율 및 주관적 화질을 향상시킨다.

HEVC에서는 변환이 적용되는 각 분할 영역들을 TU(Transform Unit)라고 정의하며, 최소 4×4에서 32×32까지 그 크기가 가변 될 수 있다. 이를 위해 HEVC에서는 H.264/MPEG-4 AVC에서 지원하는 4×4 변환, 8×8 변환 뿐만 아니라 16×16 변환과 32×32 변환을 추가적으로 정의한다. 이 때, 다양한 크기의 변환을 구현하기 위한 요구 메모리를 최소화하기 위해, H.264/MPEG-4 AVC의 4×4 변환, 8×8 변환을 재사용하는 대신, 32×32 변환을 정의하고, 이의 행렬 계수들로부터 각기 다른 샘플 크기를 갖는 모든 변환들의 행렬 계수를 복잡한 계산 없이 서브샘플링 만으로 얻을 수 있도록 설계하였다.

HEVC의 변환 기술은 이전 표준들과 마찬가지로 DCT에 기반한 형태를 갖고 있지만, 화면 내 예측이 사용되는 4×4크기의 블록에 대해서는 추가적으로 DST에 기반한 추가 변환 기술을 포함함으로써 압축 효율을 향상시키고 있다.

5. Motion information coding

움직임 벡터는 잔차 신호와 더불어 일반적으로 비트스트림에서 가장 많은 양을 차지하는 정보에 해당한다. HEVC는 H.264/MPEG-4 AVC와 마찬가지로 인접한 블록의 움직임 벡터로부터 현재 블록의 움직임 벡터를 예측함으로써 정보량을 줄이는 방법을 사용한다. 하지만 H.264/MPEG-4 AVC에



〈그림 5〉 Motion vector candidate positions in HEVC (HEVC에서 정의된 움직임 벡터 후보 위치들)

서 인접 움직임 벡터들의 중간 값만을 사용하여 예측을 수행하는데 비해, 사용 가능한 복수 개의 움직임 벡터들 중 하나를 골라 예측 값으로 사용하는 기술을 사용한다. 〈그림 5〉는 사용 가능한 움직임 벡터들을 얻기 위한 위치들을 표시한 것으로서, 동일 picture 내의 정보(spatial candidates) 뿐만 아니라 참조 picture 내의 정보(temporal candidate)들도 함께 사용하여 압축 효율을 향상시킨다.

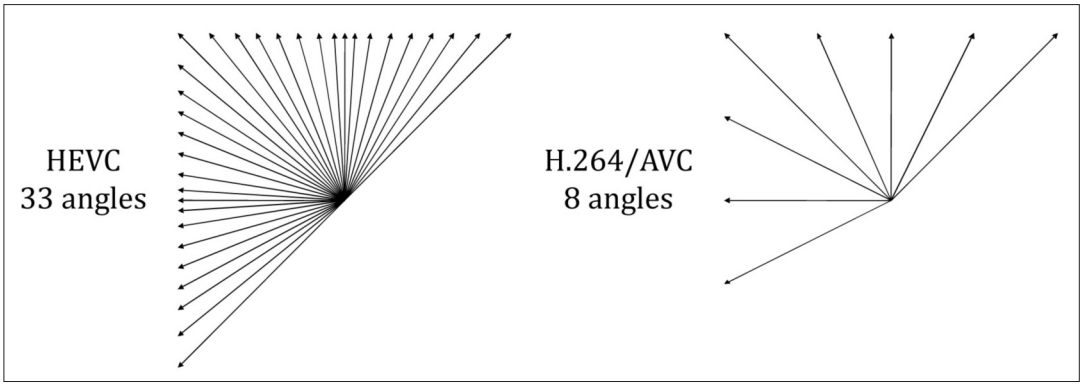
또한, 움직임 벡터 뿐 아니라 reference picture index 및 단일/양방향 예측 방향 등 화면 내 예측에서 사용되는 모든 정보들에 대한 추가적인 코딩 없이 예측된 값 만을 사용하는 블록 간 병합 모드(merge mode)를 지원하여 압축 효율을 최대화 한다.

6. Motion compensation

H.264/MPEG-4 AVC는 휘도 신호에 대해 1/4th 단위로 움직임 보상을 수행함으로써 이전 표준들에 비해서 압축 성능을 크게 향상시켰다. HEVC도 같은 정밀도를 갖는 움직임 보상 방법을 사용하지만, 몇 가지 측면에서 변화가 이루어졌다. 먼저, H.264/MPEG-4 AVC의 경우 6-tap의 길이를 갖는 필터를 1/2nd 위치의 보간 샘플을 얻는데 사용하고, bi-linear 필터를 이용하여 1/4th 위치의 보간 샘플을 얻는 2단계 과정을 거치는데 비해, HEVC의 경우 8-tap의 길이를 갖는 필터를 사용하여 1/2nd 위치의 보간 샘플을 얻으며, 7-tap의 길이를 갖는 필터를 사용하여 1/4th위치의 보간 샘플을 얻는다. HEVC는 H.264/MPEG-4 AVC에 비해 길어진 필터를 사용함으로써 고주파 성분을 갖는 특정 영상들에서의 압축 성능을 크게 향상시킬 수 있으며, 1/4th 위치의 보간 샘플을 얻는 과정에서도 1/2nd 위치의 보간 샘플과 동일하게 하나의 단계만을 거치므로 구현 상 이점을 갖는다. 한편, HEVC에서는 색차 신호에 대한 보간 방법 또한 H.264/MPEG-4 AVC에서의 bilinear filter 대신 4-tap 필터를 사용함으로써 압축 효율을 크게 향상시켰다.

7. Intra prediction

H.264/MPEG-4 AVC는 주파수 영역에서의 계수 값을 예측 값으로 사용하는 H.262/MPEG-2나 MPEG-4 Visual과는 달리 이미지 영역에서의 인



〈그림 6〉 Intra prediction modes of HEVC and H.264/MPEG-4 AVC (HEVC와 H.264/MPEG-4 AVC에서 정의된 화면 내 방향 예측 모드들)

접 샘플 값을 예측 값으로 사용함으로써 화면 내 예측의 성능을 크게 끌어올렸다. 특히, 총 8방향의 방향성을 고려할 수 있도록 함으로써 강한 방향성이 나타나는 형태에서 특히 높은 효율을 보일 수 있도록 설계되었다. HEVC도 거의 같은 방식을 사용하지만, 예측의 기준이 되는 블록의 크기가 최대 64×64까지 커짐에 따라, 〈그림 6〉과 같이 방향성의 개수를 33방향으로 확장하였다. 이처럼 늘어난 방향성의 개수에 대한 구현 부담을 최소화하기 위해서, 가로 방향과 세로 방향을 제외한 나머지 모든 방향들에 대해서 동일한 방법으로 예측 신호를 생성할 수 있도록 예측 알고리즘을 일반화 하였으며, 복잡한 계산 대신 미리 만들어진 테이블의 값을 참조함으로써 예측 샘플 값을 얻을 수 있도록 연산량을 최소화 하였다. 또한, 이 과정에서 인접 참조 샘플들의 값을 1/32 정밀도로 선형 보간하여 예측 신호의 정밀도 또한 크게 향상시켰다.

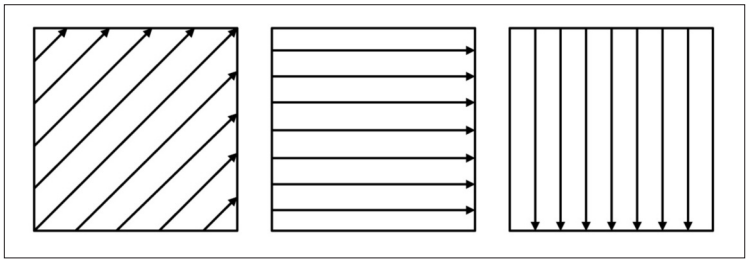
HEVC에서는 4가지의 예측 모드만을 사용하는 H.264/MPEG-4 AVC와는 달리 색차 신호에 대해 휘도 신호와 동일한 방향성을 적용할 수 있도록 허용함으로써 총 33 종류 중 하나의 방향성을 갖도록 할 수 있다. 또한, 특별히 방향성을 갖지 않는 영역에 대한 예측을 위해 H.264/MPEG-4 AVC와 유사하게 인접 샘플의 평균값을 사용하는 DC 모드와 점진적으로 부드럽게 변화하는 영역을 표현하는 planar 모드를 함께 정의하고 있다. 특히 H.264/MPEG-4 AVC의 경우 부드럽게 변화하는 영역을 표현하기 위한 plane 모드를 16×16 블록 크기에서만 허용하는데 비해, HEVC에서는 모든 블록 크기에서 planar 모드를 허용하며, 이는 H.264/MPEG-4 AVC에 비해 상대적으로 큰 크기의 예측 블록이 자주 사용되는 HEVC에서 압축 효율 및 주관적 화질을 향상시키는 역할을 한다.

8. Entropy coding

HEVC는 헤더 정보를 압축하기 위해서 가변 길이 부호화(variable length coding; VLC)를 활용하며, 나머지 비트스트림 정보는 H.264/MPEG-4 AVC에서 이미 사용되었던 엔트로피 코딩 기술인 CABAC을 사용한다. H.264/MPEG-4 AVC에서는 baseline 프로파일의 경우 연산량이 상대적으로 낮은 CAVLC를 사용하며, 나머지 프로파일에서 CABAC을 사용하는데 비해, HEVC에서는 한 가지의 엔트로피 코딩 기술을 사용함으로써 구현이 용이하도록 하였다.

비록 H.264/MPEG-4 AVC와 동일한 엔트로피 코딩 기술이 사용되지만, 다양한 측면에서 HEVC의 엔트로피 코딩 부분은 H.264/MPEG-4 AVC와 차이점을 보인다. 먼저, CABAC에서 심볼을 압축하기 위해 사용되는 문맥 정보를 유도함에 있어, 쿼드트리 구조를 활용하며, 총 문맥 정보의 수는 H.264/AVC에 비해 오히려 크게 감소되었다. 또한, 간단한 연산만을 요구하는 CABAC의 bypass mode가 H.264/MPEG-4 AVC에 비해 훨씬 폭넓게 사용됨으로써 엔트로피 코딩 부분에서의 throughput을 향상시킨다.

2차원 변환이 적용된 후 얻어진 변환 계수들을 1차원 형태의 데이터로 변환하기 위해 사용되는 계수 스캐닝 기술 또한 많은 변화가 이루어졌다. JPEG부터 사용되어 왔던 zigzag 스캐닝은 우상향 대각 방향 스캐닝으로 변경되어 구현을 좀 더 용이하게 하였고, 화면 내 예측의 경우 가로 방향, 세로 방향 스캐닝을 방향성에 맞게 적응적으로 사용함으로써 압축 효율을 향상시킨다. 〈그림 7〉은 HEVC에서 정의된 세 가지 스캐닝 방법을 나타낸 것이다.



〈그림 7〉 Three coefficient scanning patterns in HEVC (HEVC에서 정의된 세 가지 계수 스캐닝 방법들)

9. Deblocking filter

일반적으로, 블록 기반 영상 압축 기술들은 압축으로 인해 손실되는 정보들에 의해 블록 경계에서 왜곡이 나타나는 blocking artifact 현상을 겪게 된다. 이러한 현상을 감소시키기 위해 고안된 기술이 deblocking filter인데, H.264/MPEG-4 AVC는 deblocking filter가 적용된 복원 영상을 이후 picture에 대한 참조 영상으로 재활용하는 in-loop 형태를 갖는 것을 그 특징으로 하고 있다. HEVC도 마찬가지로 in-loop 형태의 deblocking filter를 사용하는데, 복잡도를 낮추기 위해서 4×4 블록 경계에 대해서는 deblocking filter를 적용하지 않으며, 세로 방향의 블록 경계에 대한 가로 방향 deblocking과 가로 방향의 블록 경계에 대한 세로 방향 deblocking을 picture 단위로 독립시킴으로써 병렬성을 향상시켰다. 즉, 모든 세로 방향의 블록 경계들에 대한 처리들은 각각 독립적으로 수행가능하며, 이후, 모든 가로 방향의 블록 경계들에 대한 처리들은 각각 독립적으로 수행 가능하다.

10. Sample adaptive offset

HEVC에서는 종전의 표준들에서 도입되지 않았던 새로운 형태의 in-loop filtering 기술인 sample adaptive offset(SAO) 기술이 포함되었다. SAO 기술은 복원된 영상의 각 샘플들을 특정 기준에 의해서 분류하고, 각 분류에 속한 샘플들의 값들을 일괄적으로 보상하는 방법으로서, 샘플 크기 별 분류 방법(band-offset)과 샘플 방향성 별 분류 방법(edge-offset)의 두 가지 분류 방법이 정의되어 있다. SAO는 특히 ringing artifact를 보상하는데 우수한 효과를 갖고 있으며, 이는 HEVC에서 도입된 큰 크기의 변환들에서 나타나는 ringing artifact를 제거하는 역할을 함으로써 주관적 화질을 크게 향상시킨다.

11. Parallelized picture partitioning

H.264/MPEG-4 AVC에서는 하나의 picture를 복수 개의 slice로 분할하고, 각각을 독립적으로 처리하는 방식을 지원한다. 하나의 slice는 복수 개의 매크로 블록으로 구성되는데, 각각의 slice들은 독립적으로 부호화/복호화가 가능하기 때문에 병렬처리에 활용될 수 있다. 한편, slice는 효율적인 네트워크 전송을 위한 패킷 크기를 한정하기 위한 목적으로도 사용되는데, HEVC에서는 이러한 목적과 병렬화를 위한 목적을 분리하기 위해, tile과

wavefront라는 두 가지 새로운 방식을 추가하였다.

HEVC에서의 slice가 연속된 CTU들로 이루어져 있는데 비해, tile은 직사각형 모양으로 나누어진 공간을 의미한다. 각 tile들은 헤더 정보를 공유하는 대신, 독립적으로 복호화가 가능한 특징을 갖는다. 따라서 복호화기에서의 병렬성을 손쉽게 증가시킬 수 있으며, 추가적으로 임의의 tile만을 복호화 하는 것을 가능하게 함으로써 전체 picture 중 일부만을 복호화 할 수 있는 기능을 제공한다.

Wavefront 기술은 하나의 picture를 CTU들의 열들로 분할하고, 각각의 열 들을 병렬화 하여 부호화하는 시나리오를 가상한다. 이를 slice를 이용하여 구현하는 경우, 각 열 들에 대한 병렬 처리는 가능하지만 각기 독립적으로 부호화되기 때문에 압축 효율이 하락하는 문제점이 있다. Wavefront 기술을 적용하는 경우, 하나의 열을 부호화 함에 있어, 이전 열의 일부분 부호화가 끝난 후 부호화를 시작하도록 한정하기 때문에, 이전 열의 부호화 결과를 일부분 재활용 할 수 있으며, 결과적으로 압축 효율 하락을 최소화 할 수 있다.

12. Special coding modes

HEVC는 특별한 목적을 위해 만들어진 세 가지 추가 코딩 모드인 I_PCM, lossless, transform skipping 모드를 포함하고 있다. I_PCM 모드는 H.264/MPEG-4 AVC에서도 포함되었던 기술로서, 예측, 변환, 양자화 및 엔트로피 코딩을 적용하지 않는다. 이러한 모드는 기존의 하이브리드 코딩 구조로 압축하기 어려운 잡음 신호 등을 처리하기 위해 고안되었다. Lossless coding 모드는 예측 단계는 포함하지만 변환과 양자화 과정을 생략함으로써 최소한의 변화 만으로 HEVC 표준안에서 무손실 압축을 할 수 있도록 한다. 마지막으로, transform skipping 모드는 변환 과정만을 생략하며, 이는 자연 영상이 아닌 컴퓨터 그래픽스로 만들어진 영상들에 대한 압축 성능을 크게 향상시킨다.

Ⅲ. 압축 성능

HEVC의 표준화가 현재 진행 중이며, 아직 최종적으로 완료되지 않았기 때문에 10년 가까이 부호화기 최적화가 이루어져 온 H.264/MPEG-4 AVC와의 정확한 압축 성능 비교는 아직 어렵다고 할 수 있다. 하지만, H.264/MPEG-4 AVC와 HEVC의 참조 소프트웨어들을 이용한 테스트 결과들을 보면, 고해상도 영상의 경우 동일 PSNR(Peak Signal-to-Noise Ratio) 수준에서 40% 이상의 비트 절감을 얻을 수 있음을 확인할 수 있다. 〈표 1〉은 Microsoft에서 테스트 한 결과[8]를 요약한 것이며, 1초마다 인트라 프레임을 삽입하는 random access 테스트 조건의 경우 4K×2K, 1080p 해상도에서 H.264/MPEG-4 AVC 대비 각각 43%, 44% 비트 절감율을 얻었으며, 지연 시간에 대한 제한을 둔 low-delay 테스트 조건의 경우 1080p에서 48%의 비트 절감율을 얻었다.

〈표 1〉 H.264/MPEG-4 AVC 대비 HEVC를 사용한 경우 동일 PSNR 값에서의 비트 절감율[8]

Sequence resolution	Random access	Low-delay
4K×2K	43%	Not tested
1080p	44%	48%
832x480 (WVGA)	34%	41%
416x240 (WQVGA)	32%	38%
720p	Not tested	51%

〈표 2〉 H.264/MPEG-4 AVC와 HEVC 간 주관적 화질 선호도 평가[9]

Sequence	Prefer H.264/MPEG-4 AVC	Prefer HEVC (half size)	No preference
Kimono (1080p)	14%	82%	5%
ParkScene (1080p)	18%	45%	36%
Vidyo3 (720p)	38%	31%	31%
Vidyo1 (720p)	54%	15%	31%
Raven (720p)	18%	32%	50%
Average	28%	41%	31%

한편, 표준화 기구를 통한 공식 테스트 결과는 아니지만, [9]에서는 주관적 화질 측면에서의 비교를 수행하였으며 〈표 2〉는 이 결과를 요약한 것이다. 이 결과에 따르면, 총 5개의 1080p 및 720p 해상도 영상들에 대해서, H.264/MPEG-4 AVC 대비 절반 크기를 갖도록 압축된 HEVC의 결과 영상을 선호하거나, H.264/MPEG-4 AVC와 대등한 화질로 판단한 경우가 전체 중 72%에 해당한다는 결과를 얻음으로써, HEVC가 많은 경우에 있어서 H.264/MPEG-4 AVC 대비 50% 압축률 향상을 이미 달성하고 있음을 증명하였다.

또한, 정지 영상 압축에 HEVC 기술을 적용한 결과[10]도 공개되었는데, JPEG-2000 대비 평균 23%의 비트 절감율을 얻음으로써 새로운 정지 영상 압축 방법으로서의 가능성도 보여주었다.

Ⅳ. 표준화 진행 과정

2004년 중반, H.264/MPEG-4 AVC의 High Profile에 대한 표준화가 완료된 이후, ITU-T VCEG과 ISO/IEC MPEG에서는 각각 차세대 영상 압축 기술에 대한 연구를 진행하여 왔다. ITU-T VCEG의 경우 2005년부터 H.264/MPEG-4 AVC의 참조 소프트웨어인 JM을 확장한 Key Technology Area(KTA) software를 공통 소프트웨어로 정하고, 다양한 신규 기술들의 성능을 평가하였으며, ISO/IEC MPEG의 경우도 2005년부터 2008년까지 관련된 다양한 기고문들이 제안되었다.

2009년 4월, MPEG에서는 H.264/MPEG-4 AVC에 비해 월등한 성능 향상을 보이는 기술들이 있는지 확인하기 위한 단계로서, Call-for-Evidence[11]가 공지되었다. 그 결과, 현재의 기술에 비해 압축 성능을 크게 증가시킬 수 있는 기술들이 존재함을 확인하고, ITU-T VCEG과 공동으로 JCT-VC라는 공동 기구를 통해 HEVC 표준화를 진행하게 되었다. 2010년 1월에는 HEVC 표준화의 초기 기술 집합을 확정하기 위한 Call-for-Proposal[12]이 공지되었으며, 기고된 제안서들 중 우수한 압축 효율을 보인 기술들[13]을 하나로 합하여 2010년 4월, Test Model under Consideration (TMuC)[14]이라고 불리는 초기 테스트 모델이 만들어졌다.

TMuC은 H.264/MPEG-4 AVC 대비 월등한 압축 효율을 보였지만, 여러 기고문들에서 각각 제안된 기술들을 하나로 합친 형태였기 때문에 수 많은

중복된 기술들이 존재했고, 이는 매우 높은 부호화/복호화 복잡도로 이어졌다. 따라서 TMuC을 구성하는 많은 기술 들 중 반드시 필요한 기술들만을 선택하기 위한 요소 기술 별 테스트 계획이 2010년 7월에 만들어졌으며, 이 테스트 결과를 통해 2010년 10월에 TMuC과 유사한 압축 효율을 가지면서도 훨씬 낮은 복잡도를 갖는 HEVC 테스트 모델 버전 1.0[15]이 공개되었다.

이후 수많은 기고문들을 통해 현재 HEVC 테스트 모델 버전 8.0[7]이 공개된 상태이며, 특히 HEVC의 첫 번째 프로파일인 HEVC Main Profile이 정의되었다. HEVC의 첫 번째 국제 표준 규격(Final Draft International Standard; FDIS)은 2013년 1월 공개를 목표로 현재 JCT-VC에서 표준화 작업이 진행 중이며, 신축적 영상 압축 기술 및 다시점 영상 압축 기술 등 HEVC 표준에 기반한 확장 기술 표준화도 이미 진행되고 있다.

V. 결 론

HEVC는 ITU-T VCEG과 ISO/IEC MPEG이 공동으로 진행하고 있는 차세대 영상 압축 표준으로서, H.264/MPEG-4 AVC 대비 동일 주관적 화질에서 50%의 압축률 향상을 제공할 수 있도록 설계되었다. HEVC는 기존 압축 기술과 유사한 block-based hybrid coding 방법에 기초하고 있지만, 각 요소 기술 측면에서 많은 발전이 이루어져, 뛰어난 압축 효율을 얻을 수 있다. 한편, 최신 멀티 프로세서 기술을 최대로 활용할 수 있도록, 병렬화를 위한 고려가 설계 단계에서 반영됨으로써 HEVC의 복잡도는 H.264/MPEG-4 AVC 대비 크게 증가하지 않을 것으로 예측되고 있어, 향후 UHDTV, HD 모바일 스트리밍, Full-HD 3D 영상 서비스 등 다양한 분야에 적용될 것으로 판단된다.

참 고 문 헌

[1] ITU-T, "Video Codec for Audiovisual Services at px64 kbit/s," ITU-T Recommendation H.261, Version 1: Nov. 1990; Version 2: Mar. 1993.

[2] ITU-T, "Video coding for low bit rate communication," ITU-T Recommendation H.263, Nov. 1995 (and subsequent editions).

[3] ISO/IEC JTC 1, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbit/s - Part 2: Video," ISO/IEC 11172-2 (MPEG-1), 1993.

[4] ISO/IEC JTC1, "Coding of audio-visual objects - Part 2: Visual," ISO/IEC 14496-2 (MPEG-4 Visual version 1), April 1999 (and subsequent editions).

[5] ITU-T and ISO/IEC JTC 1, "Generic coding of moving pictures and associated audio information - Part 2: Video," ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG 2 Video), Nov. 1994.

[6] ITU-T and ISO/IEC JTC 1, "Advanced Video Coding for generic audio-visual services," ITU T Rec. H.264 and ISO/IEC 14496-10 (AVC), May 2003 (and subsequent editions).

[7] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 8," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-J1003, July 2012.

[8] B. Li, G. J. Sullivan and J. Xu, "Compression performance of high efficiency video coding (HEVC) working draft 4," Proc. Of ISCAS, Seoul, Korea, pp. 20-23, May 2012.

[9] F. Kossentini, N. Mahdi, H. Guermazi, M. Horowitz, S. Xu, B. Li, G. J. Sullivan and J. Xu, "Informal subjective quality comparison of compression performance of HEVC working draft 5 with AVC high profile," JCTVC-H0562, 8th JCT-VC meeting, San Jose, USA, Feb. 2012.
[10] T. Nguyen and D. Marpe, "Performance comparision of HM 6.0 with existing still image compression schemes using a test set of popular still images," JCTVC-I0595, 9th JCT-VC meeting, Geneva, CH, Apr. 2012.
[11] ISO/IEC JTC 1/SC 29/WG 11, "Call for evidence on high-performance video coding (HVC)," ISO/IEC MPEG document N10553, Apr. 2009.
[12] ITU-T and ISO/IEC JTC1, "Joint Call for Proposals on Video Compression Technology," ITU-T VCEG-AM91 and ISO/IEC MPEG document N11113, Jan. 2010.
[13] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, and K. Ugur, "Special Section on the Joint Call for Proposals on High Efficiency Video Coding (HEVC) Standardization," IEEE Trans. Circuits Syst. Video Technol., vol. 20, pp. 1661-1666, Dec. 2010.
[14] ITU-T and ISO/IEC JTC1, "Test Model under Consideration," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-A205, Apr. 2010.
[15] T. Wiegand, W.-J. Han, B. Bross, J.-R. Ohm, and G. J. Sullivan, "WD1: working draft 1 of high-efficiency video coding," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-C402, Oct. 2010.

필 자 소 개

한 우 진

– 1995년 2월 : KAIST 전산학과 공학사
– 1997년 2월 : KAIST 전산학과 공학석사
– 2002년 2월 : KAIST 전산학과 공학박사
– 2002년 3월 ~ 2003년 3월 : SL2 연구소장
– 2003년 4월 ~ 2011년 8월 : 삼성전자 DMC 연구소 수석연구원
– 2010년 10월 ~ 현재 : HEVC Working Draft Editor
– 2011년 9월 ~ 현재 : 가천대학교 소프트웨어설계경영학과 조교수
– 주관심분야 : 영상압축, 영상이해, 멀티미디어통신