

# 차세대 비디오 압축 부호화 - HEVC 표준 동향

김재일, 김문철  
한국과학기술원

## 요약

본고에서는 최근 ISO/IEC와 ITU-T에서 공동으로 표준을 완성한 차세대 비디오 부호화 표준인 High Efficiency Video Coding(HEVC) 기술에 대해 알아본다.

## I. 서론

HEVC는 ISO/IEC의 MPEG 그룹과 ITU-T VCEG 그룹에 의해 결성된 Joint Collaborative Team on Video Coding (JCT-VC)에 의해 기존 H.264/MPEG-4 AVC 대비 약 2배의 압축 부호화 성능을 목표로 표준화가 진행되어 2013년 1월 제12차 JCT-VC 제네바 회의에서 최종 표준안이 발표된 차세대 비디오 압축 부호화 규격이다. 향후, 고성능 스마트폰에서의 고화질 비디오 서비스, IPTV, 감시카메라, 디지털 캠코더, UHDTV 방송, 3D방송 등 많은 분야에 빠른 속도로 적용될 때 우 중요한 비디오 압축 표준이다. 본고에서는 이러한 HEVC의 특징과 성능에 대해 기존 H.264/MPEG-4 AVC 표준 규격과 대비하여 설명한다.

## II. HEVC 표준화 현황

2003년 H.264/MPEG-4 AVC 표준이 완성된 이후, 우리나라가 이를 DMB에 적용함으로써 세계최초로 상용화를 시작한 이래로 많은 분야에 H.264/MPEG-4 AVC 표준 기술 채택이 확산되었다. 그 이후, ISO/IEC MPEG과 ITU-T VCEG은 서로 H.264/MPEG-4 AVC 표준 기술보다 부호화 성능을 더 향상된 새로운 기술 표준을 마련하기 위해 요구사항 및 사전 기술 탐색을 모색해 왔다. 그러던 중 2010년 1월 MPEG과 VCEG는 차세대 비디오 압축 부호화 규격 개발을 위해 JCT-VC를 구성하고 기술제안요청서를 발표하였다. 2010년 4월 독일 드레스덴에서

개최된 제1차 JCT-VC 회의에 27개의 기술이 제안되었고, 이를 평가하여 표준 개발을 위한 Test Model Under Consideration이라는 HEVC의 전신이 되는 기본 플랫폼을 결정하였다. 2010년 10월 중국 광저우에서 개최된 제3차 JCT-VC 회의에서 HM (HEVC Test Model) 1.0이 완성되었으며, 그 이후 많은 기술 기고와 평가를 통해 2013년 1월 HEVC 최종 표준안(Draft Final International Standard, DFIS)가 완성되었다. 현재는 전문가용 컬러포맷(4:2:2, 4:4:4) 및 높은 비트 샘플(12 bit 이상) 부호화를 위해 HEVC Range Extensions 표준화를 진행하고 있으며 2014년 1월 표준완성을 목표로 하고 있다.

## III. HEVC 비디오 부호화 기술

HEVC는 기존의 H.264/MPEG-4 AVC 비디오 압축 기술과 비슷한 블록 기반 하이브리드 비디오 부호화 방법이다. 하지만 기존 H.264/MPEG-4 AVC가 16×16 화소 블록 크기의 매크로블록 단위로 입력 비디오 화면을 처리한 것에 비해 HEVC는 이를 최대 64×64 화소 블록 크기로 확장하여 입력 영상의 특징에 따라 다양한 작은 크기의 화소 블록으로 분할하여 부호화를 수행할 수 있도록 설계하였다. HEVC는 블록 크기의 확장을 포함하여 각 요소기술들의 확장 및 새로운 요소기술 개발을 통하여 H.264/MPEG-4 AVC 대비 평균 50% 정도의 부호화 성능을 향상 하였다 [1]. 본 장에서는 HEVC의 성능 향상을 위해 확장 및 새롭게 제안된 기술에 대해 소개한다.

### 1. 확장 블록 구조

HEVC에 채택된 기술 중 확장 블록 구조(large block structure)는 기본 부호화 단위 블록인 CTU (Coding Tree Unit)와 최대 64×64에서 최소 8×8 크기의 트리 구조 CU(Coding Unit), 최대 32×32에서 최소 4×4 크기의 트리 구조 TU(Transform Unit), 보다 정확한 예측 부호화를 위한 확장된 블록 분할을 지원하는 PU(Prediction Unit)로 구성되어

있다 [1]. 이러한 확장 블록 구조는 압축 향상에 크게 기여한 기술로서 울-왜곡 비용 최소화 측면에서 다양한 블록 모드 및 변환 크기를 지원한다. HEVC의 다양한 블록 모드 및 변환 크기는 영상의 텍스처나 움직임 벡터의 변화에 따라 압축을 수행하기 때문에 부호화 성능을 크게 개선 하는데 기여하였다.

HEVC의 기본 압축 단위인 CU는 각 크기마다 이전 CU의 가로, 세로 1/2인 크기인 quadtree 형태를 지원한다. 따라서 CTU의 크기가 64×64인 경우 가능한 CU의 크기는 64×64, 32×32, 16×16, 8×8이다. HM의 CU는 화면간/화면내 예측 모드 또는 PCM 모드를 위해 다양한 크기의 예측 모드 블록으로 나뉠 수 있으며, H.264/MPEG-4 AVC의 방식과 비슷하다. 화면간/화면내 예측을 위한 블록을 HEVC에서는 PU라 하며 CU가 2N×2N일 때 2N×2N SKIP, 2N×2N Inter, 2N×N Inter, N×2N Inter, 2N×2N Intra, N×N Intra, PCM 를 지원한다(가장 작은 CU의 경우에만 N×N Intra 모드를 지원한다). 뿐만 아니라 위 직사각 형태의 모드 외에 2N×nU, 2N×nD, nL×2N, nR×2N 형태의 모드도 지원을 하며[2], 화면간 모드에서 가능한 정방향/비정방향 예측은 <그림 1>과 같다. 예를 들어 32×32 크기인 CU는 32×32 SKIP, 32×32 Inter, 32×16 Inter, 16×32 Inter, 32×8 Inter, 32×24 Inter, 8×32 Inter, 24×32 Inter, 32×32 Intra, 32×32 PCM PU를 지원한다.

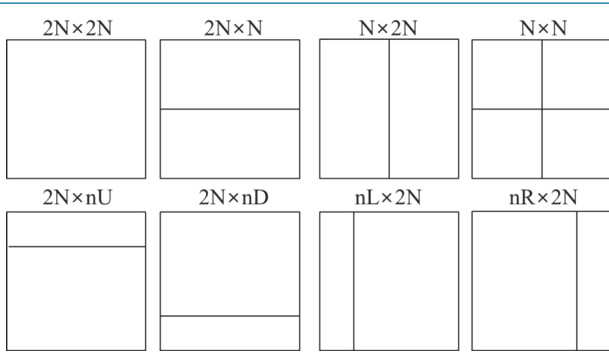


그림 1. HM에서 지원하는 PU 구조

CU의 여러 예측 블록인 PU를 통해 잔차 신호가 얻어지면 변환 과정을 수행한다. HEVC에 정의된 확장 블록 구조의 가장 큰 특징은 PU와 TU를 독립적으로 수행하는 것으로 각 PU를 수행해 얻어진 CU 크기의 잔차 신호가 quadtree 형태로 다양한 크기의 변환 과정을 수행한다. HEVC에서 변환 과정은 예측 블록의 크기를 넘어선 영역에 적용되며 예측 블록의 경계를 포함하여 수행한다 [1]. <그림 2>는 실제 Racehorse 영상에서 CU, PU, TU가 선택된 최종 모드를 도식화한 그림이다. CU와 PU, TU의 최적화된 조합으로 인해 화면간 움직임 예측 성능이 혁신적으로 향상되었다. 따라서 예측을 통해 얻어진 영상의 잔

차 신호의 양이 상대적으로 작아지게 되었다.

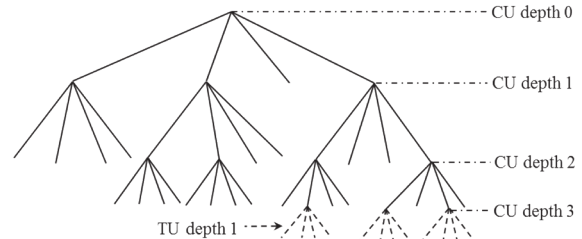
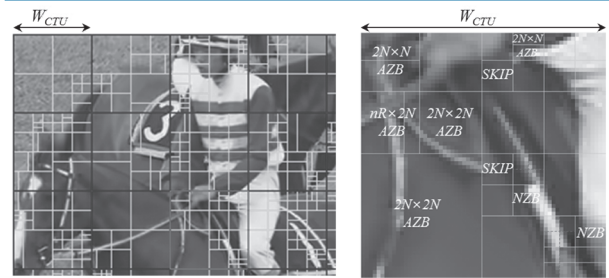


그림 2. CTU내 CU, PU, TU의 부호화 선택 예

## 2. 화면내 예측 모드

H.264/MPEG-4 AVC에서 화면내 예측 방법은 16×16, 8×8, 4×4 블록 크기에 대해 16×16은 4개, 8×8과 4×4는 9개의 방향성 모드를 통해 예측을 한다 [3]. HEVC는 화면내 방향성 예측 부호화 성능을 향상하기 위해 4×4, 8×8, 16×16, 32×32, 64×64 크기에 대해 33개의 방향성 예측을 수행하여 더 정교한 방향성 예측 부호화를 수행한다. 이때 방향성 데이터를 생성하기 위해 1/32 픽셀 단위의 선형 보간 방법을 이용한다 [4].

<그림 3>은 H.264/MPEG-4 AVC와 HEVC의 화면내 예측 모드이다. HEVC는 모드 2번의 왼쪽 대각선 방향부터 모드 34번의 오른쪽 대각선 방향까지 총 33개의 방향성 모드와 모드 0인 planar 예측 방법, 모드 1인 DC 예측 방법을 지원한다. HEVC의 휘도 신호 블록에서 총 34개의 부호화 모드를 지원하기 때문에 부호화 모드 비트의 비중이 기존 H.264/AVC에 비해 커지게 된다. 따라서 화면내 모드의 모드 정보를 부호화하기 위하여 주변 블록의 모드정보 및 선택 비율이 높은 모드정보를 통해 얻어진 MPM (Most Probable Mode) 정보와 H.264/MPEG-4 AVC 부터 사용했던 MPM을 제외한 rem\_intra\_pred\_mode 모드들을 잘 조합하여 방향성 모드 정보를 부호화한다 [5].

HEVC에서 화면내 예측 모드인 planar 모드는 H.264/MPEG-4 AVC에서 화면내 텍스처가 부드럽게 변화하는 영역을 예측하기 위한 16×16 plane 모드와 비슷한 동작을 하는 모드로, CTU내 모든 CU 크기 및 색차 신호에도 planar 모드를

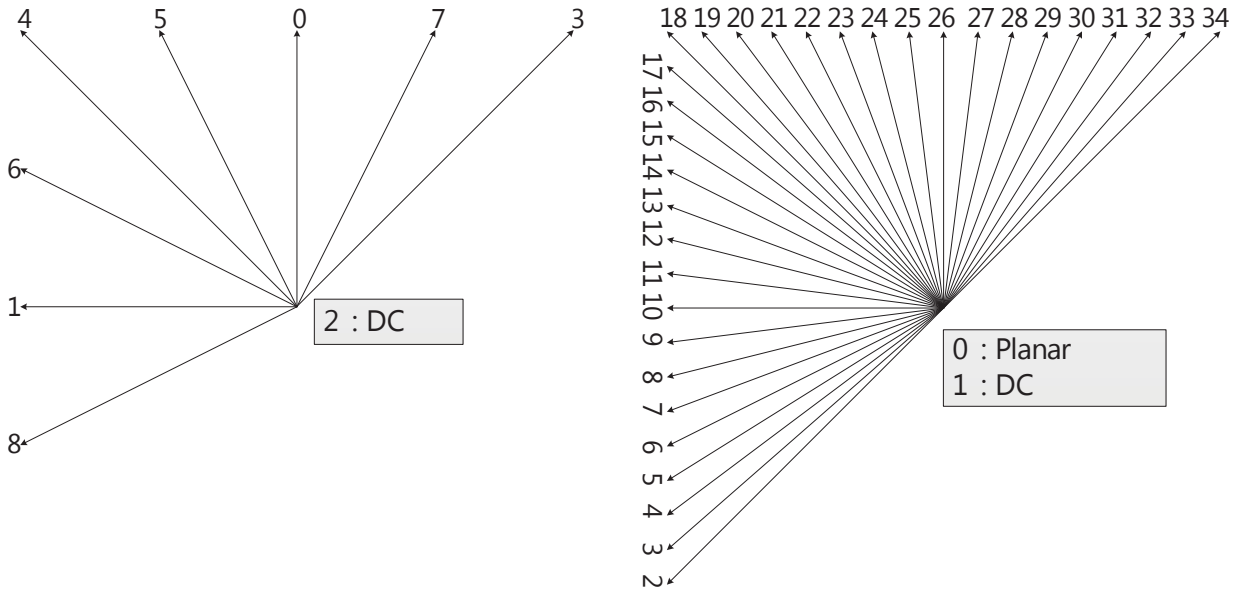


그림 3. H.264/MPEG-4 AVC 및 HEVC의 화면내 예측 모드 [3][5]

지원을 하며 이는 텍스처가 점차적으로 변하는 단순한 영역에서의 압축 성능 및 주관적 화질을 향상하는데 매우 중요한 부호화 톨로서 사용된다.

HEVC의 색차 신호 블록의 화면내 예측에서는 planar (모드 0), vertical (모드 26), horizontal (모드 10), DC (모드 1) 모드 이외에 휘도 신호 블록에서 최적의 방향성 모드로 선택되었던 모드를 재 사용하는 모드를 사용하여 화면내 예측 부호화를 수행한다. 색차 신호는 휘도 신호에 비하여 복잡하지 않고, 휘도의 방향성과 비슷한 경우가 많기 때문에 위와 같은 다섯 가지 예측 모드만을 사용해 색차 신호의 화면내 방향성 모드를 결정해 부호화를 수행한다 [5].

### 3. 화면간 예측 모드

화면간 예측 모드 방법은 기존의 H.264/MPEG-4 AVC와 구조적인 측면에서 크게 차이가 나지 않는다. 단지 기존의 구조를 좀더 효율적으로 개선하여 부호화 성능을 향상하였다. 특별히 HEVC의 확장 블록 구조는 SKIP 모드를 포함한 다양한 크기의 예측을 가능하게 하여 화면간 예측 성능을 획기적으로 개선하였다 [1]. 예측 성능의 개선뿐만 아니라 움직임 벡터 및 블록 구조를 저장하는 신택스 비트량을 감소하기 위한 방향으로 여러 연구가 수행 되었다.

그 중 움직임 벡터를 부호화 성능을 향상하기 위하여 기존의 H.264/MPEG-4 AVC는 주변 블록의 중간 값 움직임 벡터를 현재 블록의 움직임 벡터를 부호화 하는데 사용하였지만 HEVC에서는 공간적인 주위 블록 및 시간적인 동일 위치의 움직임 벡

터로 여러 개의 움직임 벡터 예측 후보를 얻어낸 후 움직임 벡터를 최적으로 부호화하는 방법을 사용한다 [1]. 이렇게 함으로써 어떤 움직임 벡터가 최적인지를 올-왜곡 관점에서 찾아 움직임 벡터의 부호화 효율을 개선 하였다. <그림 4>에서 보듯이 H.264/MPEG-4 AVC는 부호화기가 현재 블록의 세 개의 주변 블록의 움직임 벡터 중 중간 값을 일괄적으로 선택하여 부호화를 수행하고 추가적인 예측 움직임 벡터(PMV)의 전송 없이 복호화기도 동일한 작업을 수행하도록 하였으나 HEVC는 부호화기가 5개의 주변 블록들과 한 개의 이전 프레임의 동일 위치 (collocated) 블록 중 현재블록의 움직임과 가장 유사한 움직임을 갖는 블록의 움직임 벡터를 선택하여 해당 블록의 인덱스를 복호화기로 전달하는 방법을 사용하여 비록 움직임 벡터를 전송하는 정보량은 증가하였으나 움직임 예측 성능 자체를 개선 시키므로 전체적인 부호화기의 성능을 향상시키는 방법을 사용하고 있다.

또한, 현재 블록의 움직임 벡터를 포함한 움직임 벡터를 표현하

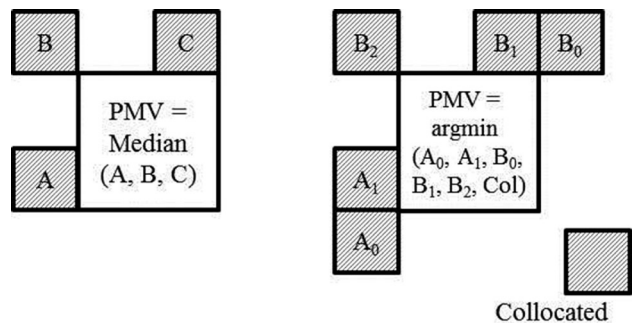


그림 4. H.264/MPEG-4 AVC와 HEVC의 MV 예측 비교

기 위해 사용하는 모든 정보는 주위 블록과 유사할 가능성이 높다. 따라서 HEVC에서는 주변의 움직임 정보가 같은지를 간단하게 알려줘 주변의 움직임 정보를 통합 할 수 있는 merge 모드를 사용한다 [6]. HEVC는 각 CU의  $2N \times 2N$  PU에 대하여 움직임 예측에 이용된 것과 동일한 후보 블록들을 이용하여 움직임 벡터로 사용될 예측 블록을 선택하고 예측 블록의 움직임 벡터와 동일한 움직임 벡터를 사용하여 움직임 예측을 수행한다. 따라서 merge 모드의 경우 참조 프레임 정보와 리스트 정보, 차분 움직임 벡터(Motion Vector Difference)를 전송하지 않고 예측에 사용된 블록의 인덱스 정보만 전송하므로 움직임벡터 전송에 사용되는 정보량을 줄일 수 있다. 그리고 merge모드는 merge skip으로 명명되는 한가지 움직임 방법을 더 이용하고 있다. Merge skip 움직임 예측은 merge mode처럼 움직임 벡터를 주변 블록의 것을 동일하게 사용하여 움직임을 예측하고 예측된 블록의 화소 정보를 그대로 현재 블록의 화소 값으로 사용하는 예측 방법이다. 잔차 신호를 이용하지 않으므로 변환 및 양자화 과정이 생략되며 잔차 신호에 사용되는 정보량을 줄일 수 있다. 움직임 예측 모드, merge 모드 그리고 merge skip의 차이를 표로 정리하면 <표 1>과 같다.

표 1. 각 움직임 예측 방법 비교

	움직임 예측 모드	Merge 모드	Merge skip
예측 움직임 벡터 (PMV)	O	O	O
차분 움직임 벡터 (MVD)	O	X	X
잔차 신호	O	O	X

움직임 보상을 위해 기존의 H.264/MPEG-4 AVC에서는 1/2 화소 위치의 보간 정보를 얻기 위해 6-tap의 필터를, 1/4 화소 위치의 보간 정보를 얻기 위해 이중선형(bilinear) 보간 필터를 두 단계로 적용하였다 [3]. 하지만 HEVC의 경우 좀 더 긴 탭을 사용하는데 1/2 화소 위치의 보간 정보를 얻기 위하여 8-tap의 필터를, 1/4 화소 위치의 보간 정보를 얻기 위해 7-tap 필터를 한번에 적용한다 [1]. 따라서 긴 탭 필터의 사용으로 인해 복잡한 영상에서 압축 성능을 향상 하였으며 한번에 1/2 및 1/4 화소 위치의 보간을 동시에 수행하기 때문에 구현상의 이점을 갖는다. <그림 5>는 보간 영상의 예시를 보여준다. 음영이 있는 화소는 정수화소이고 나머지 부분은 정수화소를 이용해 보간된 화소들이다. a, b, c, d, h, n으로 표시된 화소들은 1차원 필터를 사용하여 보간되고 나머지 화소들은 2차원 필터를 사용하여 보간 된다.

A <sub>1,1</sub>					A <sub>0,1</sub>	a <sub>0,1</sub>	b <sub>0,1</sub>	c <sub>0,1</sub>	A <sub>1,1</sub>					A <sub>2,1</sub>
A <sub>1,0</sub>					A <sub>0,0</sub>	a <sub>0,0</sub>	b <sub>0,0</sub>	c <sub>0,0</sub>	A <sub>1,0</sub>					A <sub>2,0</sub>
d <sub>1,0</sub>					d <sub>0,0</sub>	e <sub>0,0</sub>	f <sub>0,0</sub>	g <sub>0,0</sub>	d <sub>1,0</sub>					d <sub>2,0</sub>
h <sub>1,0</sub>					h <sub>0,0</sub>	i <sub>0,0</sub>	j <sub>0,0</sub>	k <sub>0,0</sub>	h <sub>1,0</sub>					h <sub>2,0</sub>
n <sub>1,0</sub>					n <sub>0,0</sub>	p <sub>0,0</sub>	q <sub>0,0</sub>	r <sub>0,0</sub>	n <sub>1,0</sub>					n <sub>2,0</sub>
A <sub>1,1</sub>					A <sub>0,1</sub>	a <sub>0,1</sub>	b <sub>0,1</sub>	c <sub>0,1</sub>	A <sub>1,1</sub>					A <sub>2,1</sub>
A <sub>1,2</sub>					A <sub>0,2</sub>	a <sub>0,2</sub>	b <sub>0,2</sub>	c <sub>0,2</sub>	A <sub>1,2</sub>					A <sub>2,2</sub>

그림 5. HEVC의 보간 화소 [5]

#### 4. 변환 및 양자화

HEVC에서는 기존의 하이브리드 비디오 부호화 방식과 유사하게 예측이 수행된 이후 예측과 원 영상 간의 잔차신호에 대해 변환 및 양자화 과정을 수행하여 부호화를 진행한다. 이때 각 변환을 수행하는 분할 영역을 TU 하며  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$  크기의 변환을 지원한다. 이러한 변환을 설계하기 위하여 처음에는 초기 HEVC에서는  $4 \times 4$ ,  $8 \times 8$  크기 변환 커널은 H.264/MPEG-4 AVC의 정수 변환(integer transform)을,  $16 \times 16$ ,  $32 \times 32$  크기 변환은 Chen의 고속 DCT 커널을 사용하였다. 하지만 각 변환의 크기에 일관성을 갖는 변환 방법도 모든 크기 ( $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ )의 변환 커널이 최종적으로 채택되었으며, 작은 크기의 변환 커널 계수를 따로 저장할 필요가 없이 큰 변환 커널 계수만을 이용해서 작은 커널의 계수를 얻어낼 수 있도록 커널의 크기에 따른 norm을 조절하여 설계하였다 [5]. 또한, H.264/MPEG-4 AVC와는 다르게 DCT를 단순 근사화 하여 단일 norm을 사용하도록 설계하였기 때문에 모든 위치에 QP에 따라 동일한 양자화 스텝만을 적용한다. 따라서 변환 및 양자화 테이블을 따로 저장할 필요가 없어 양자화 및 역양자화 과정이 간단하다. 뿐만 아니라 모든 연산이 16비트 내에서 수행 되도록 절삭 연산(clipping operation)을 수행하여 추가적인 하드웨어 복잡도 유발을 피하도록 설계하였다.

HEVC의 변환 기술은 DCT(Discrete Cosine Transform) 기반의 변환 방법뿐만 아니라 화면내 예측 모드의  $4 \times 4$  블록에 대해서는 DST(Discrete Sine Transform) 기반의 예측을 수행한다 [5]. 이는  $4 \times 4$  화면내 예측의 경우 예측 블록내 화소간의



상관도가 높기 때문에 DST를 적용함으로써 부호화 성능을 개선하였다 [4]. 또한, 화면내 또는 화면간 모드 4×4 블록에서 변환을 수행하지 않는 모드를 지원한다 [1]. 텍스트 영상과 같이 블록 내 텍스처가 복잡한 신호의 경우 변환을 수행하지 않는 것이 부호화 성능을 향상하게 된다. 따라서 이런 특별한 경우 (HEVC에서는 클래스 F 영상) 부호화 성능 및 주관적 화질을 향상하기 위하여 변환 스킵 모드 (Transform skip mode)가 채택 되었다.

### 5. 엔트로피 부호화 (Entropy coding)

H.264/MPEG-4 AVC 는 복잡도가 상대적으로 낮은 baseline 프로파일에서 CAVLC(Context-Adaptive Variable Length Coding)를 사용하고, 상대적으로 복잡도가 높은 나머지 프로파일에 대해 CABAC (Context-Adaptive Binary Arithmetic Coding)를 사용하여 엔트로피 부호화를 수행한다. 하지만 HEVC의 경우 SPS(Sequence Parameter Set), PPS(Picture Parameter Set), Slice 헤더 등과 같은 헤더 정보를 압축하기 위해서만 VLC를 사용하며 나머지 CTU 단위의 부호화 정보를 포함한 잔차 신호에 대한 부호화는 CABAC만을 사용하여 엔트로피 부호화를 수행한다 [1]. HEVC의 CABAC는 H.264/MPEG-4 AVC의 기본 개념인 심볼을 압축하기 위하여 문맥 정보를 얻어내어 엔트로피 부호화를 수행하는 방식은 동일하지만 총 문맥 정보 수는 감소하였으며 엔트로피 부호화에서 문제가 되는 처리량(throughput)을 개선하기 위하여 다양한 방식의 기술이 채택 되었다 [7].

HEVC에서 변환 및 양자화 된 신호를 엔트로피 부호화 하기 위한 스캐닝 방법이 크게 개선 및 확장 되었다. 먼저 JPEG 및 H.264/MPEG-4 AVC와 같은 기존의 비디오 부호화 방법에서 사용했던 Zigzag 스캐닝 방법에서 <그림 6>과 같은 4×4 블록 단위의 대각선 방향의 스캐닝으로 구현의 용이성을 위하여 개선 되었다. 이러한 개선은 각 블록을 스캐닝 및 엔트로피 부호화 할 때 소프트웨어의 SIMD 또는 하드웨어의 구현 복잡도가

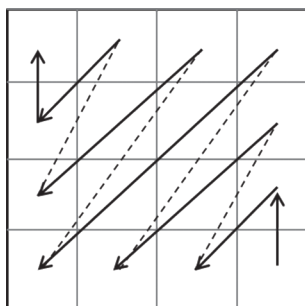


그림 6. HEVC에서 4×4 TU에 대한 대각선 (diagonal) 스캐닝 방법 [8]

높아지는 것을 막아 준다. 뿐만 아니라 4×4 및 8×8 화면내 예측에서 방향성 모드에 따라 <그림 7>과 같이 스캐닝을 다르게 적용하도록 하였으며 vertical 방향 예측의 경우 horizontal 방향의 스캐닝으로, horizontal 방향 예측의 경우 vertical 방향의 스캐닝으로, 나머지 모드에 대해서는 <그림 6>과 같은 대각선 방향 스캐닝을 수행하도록 한다. 이는 각 화면내 예측의 방향성에 따라 변환 이후 잔차신호의 배열이 vertical 방향 예측은 horizontal 방향으로, horizontal 방향의 경우 vertical 방향으로 데이터가 집중되는 특성을 이용해 부호화를 효율적으로 수행하기 위해 각 모드에 따른 다른 방향성 스캐닝 방법이 채택 되었다.

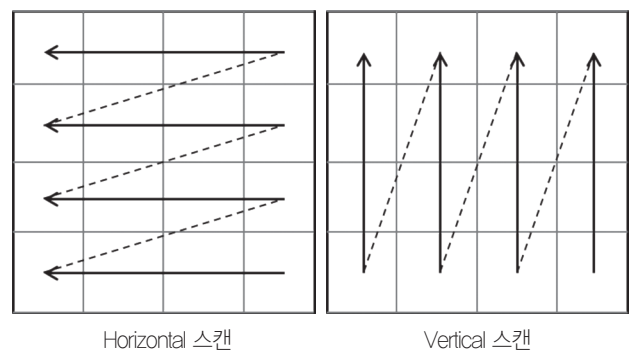


그림 7. HEVC에서 화면내 예측의 방향성 스캐닝 방법 [8]

### 6. 루프내 필터 (In-loop filtering)

루프내 필터는 부호화 효율의 향상은 물론 주관적 화질 개선에 기여하는 기술로서, HEVC에는 H.264/MPEG-4 AVC와 마찬가지로 루프내 필터가 표준에 포함되어 있다. HEVC에는 블록화 현상 (blocking artifact)을 완화하기 위해 H.264/MPEG-4 AVC에서도 도입되었던 디블록킹 필터(de-blocking)가 복잡도 측면에서 개선된 형태[9]로 포함되어 있고 그에 더하여 복원 영상의 화소 값에 적응적으로 오프셋(offset)을 적용하여 원본 영상으로부터의 왜곡을 줄이는[10] SAO (sample adaptive offset) 가 새롭게 추가되었다. SAO는 deranging 필터로서 양자화 과정에서 고주파 성분이 많이 제거 됨으로써 객체의 경계영역에서 생기는 여러 겹의 경계왜곡 현상을 완화 시켜 주관적 화질을 향상 시키는데 효과적이다. H.264/MPEG-4 AVC에서는 1개의 루프내 필터를 포함하는데 비해, HEVC는 2개의 루프내 필터를 포함하고 있다.

#### 6.1 디블록킹 필터 (deblocking filter)

HEVC는 블록 기반으로 예측과 변환 및 양자화를 수행하기

때문에 복원영상에서 블록 경계를 따라 화소 값에서 불연속점이 발생할 수 있다 [9]. 이 같이 불연속점이 나타나는 것을 블록화 현상 (blocking artifact)이라고 하는데 이 블록화 현상은 주관적인 화질에 영향을 주는 것은 물론 부호화 효율에도 영향을 미친다. 블록화 현상을 완화하기 위하여 H.264/MPEG-4 AVC에서와 마찬가지로 HEVC에서도 복원영상에 대하여 디블록킹 필터를 적용한다. 블록화 현상의 주요 원인은 블록 기반으로 수행되는 예측과 변환 및 양자화이므로 블록화 현상은 PU(prediction unit)와 TU(transform unit)의 경계에서 나타난다. 따라서 디블록킹 필터는 PU와 TU의 경계에 조건적으로 적용된다. 각 경계에 대해 경계강도(boundary strength, BS)를 계산하여 BS값과 경계 주변의 화소 값을 이용하여 필터를 적용할지 여부와 필터의 종류를 결정한다. <그림 8>은 디블록킹 필터의 전반적인 실행 순서를 나타낸다.

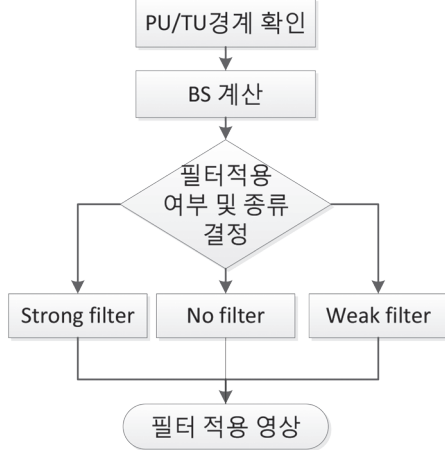


그림 8. HEVC 디블록킹 필터 실행 순서

H.264/MPEG-4 AVC와 비교하여 HEVC의 디블록킹 필터는 복잡도를 개선함은 물론 병렬 설계에 효과적인 방향으로 설계되었다. 우선 H.264/MPEG-4 AVC에서 4×4 크기의 블록에 적용되던 것을 휘도와 색차 신호 블록에서 8×8 크기 이상의 블록에 적용하도록 하여 복잡도를 완화하였다. 또한 HEVC의 디블록킹 필터는 H.264/MPEG-4 AVC에서 BS가 0-4의 값을 가졌던 것과는 다르게 0-2의 값만을 갖는 것으로 간소화되었다. 또한 HEVC의 디블록킹 필터는 각 경계에서의 필터링 간의 의존성을 없애도록 설계되어 한 프레임 내에서 모든 수직경계 필터링이 독립적으로 적용될 수 있고 수직경계 필터링이 끝난 후에는 모든 수평경계 필터링이 독립적으로 적용될 수 있다.

### 6.2 SAO (Sample Adaptive Offset)

SAO는 H.264/MPEG-4 AVC에서는 존재하지 않았던 기술로서 HEVC에 새롭게 추가된 기술이다. SAO는 각 화소를 4개

의 카테고리로 분류한 후 카테고리 별로 오프셋을 적용하여 원본 영상으로부터의 왜곡을 줄이는 기술이다. SAO에서 적용되는 오프셋에는 크게 주변 픽셀과의 관계로부터 화소를 분류하는 EO (edge offset, SAO type 1-4)과 화소 값 자체 이용하여 화소를 분류하는 BO (band offset, SAO type 5)이 있다. Band offset에서는 화소 값의 범위를 32구간의 등간격으로 분류한다. 그 다음 연속되는 4구간에 구간별로 오프셋을 적용하여 원본 영상과의 왜곡을 줄인다. Edge offset에서는 화소를 분류하기 위해 주변 화소와의 관계를 이용하는데 주변화소를 정의하는데 4종류의 패턴이 있다. <그림 9>는 주변 화소를 정의하는 4패턴(SAO type 1-4)을 나타낸다.

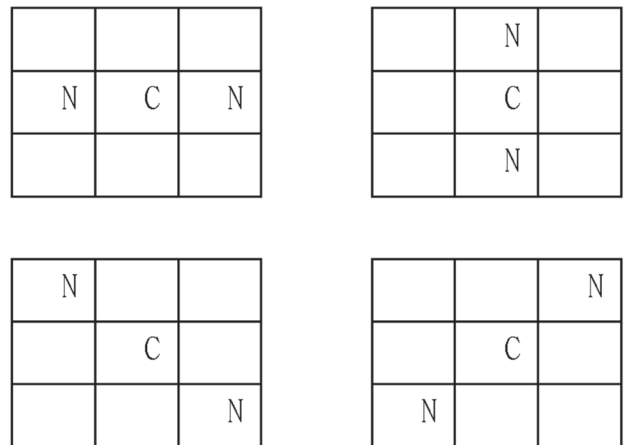


그림 9. 주변 화소를 정의하는 4 패턴

<그림 9>와 같이 현재 화소 C를 분류하기 위해 주변 화소 N이 이용되는데 세 화소의 크기의 관계에 따라 현재 화소 C의 카테고리가 정의된다. <표 2>는 각 카테고리별 화소값의 관계를 나타낸다. 각 화소를 <표 2>에 따라 4개의 카테고리로 분류하여 오프셋을 적용한다.

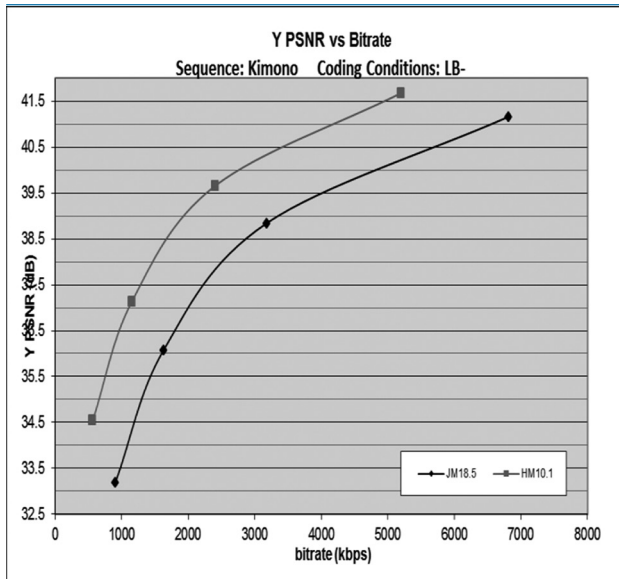
표 2. 화소 값의 관계에 따른 카테고리 분류

카테고리	화소 값의 관계
0	아래에 해당 없음
1	C < 2개의 N
2	C < 1개의 N and C = 1개의 N
3	C > 1개의 N and C = 1개의 N
4	C > 2개의 N

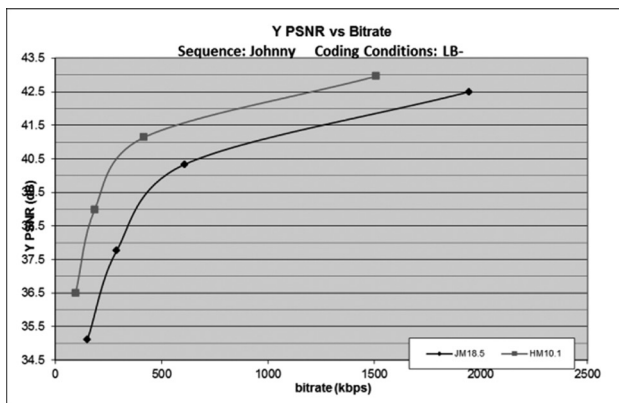
SAO type은 각 CTU당 하나씩으로 결정되고 왼쪽 CTU의 SAO정보나 위쪽 CTU의 SAO정보를 merge하여 SAO에 할당되는 비트를 줄일 수도 있다.

## IV. HEVC 부호화 성능

HEVC와 H.264/AVC의 성능을 비교하기 위하여 다음과 같은 실험을 수행 하였다. 부호화 성능 비교는 H.264/AVC 참조 모델인 JM 18.5[11]와 HEVC의 최신 참조 모델인 HM10.1[12]을 이용해 실험 하였다. 실험 조건은 HM의 Low delay 모드를 이용해 실험 하였으며 JM의 경우 JM 18.5 소스코드 안에 HM



Kimono (1080p)



Johnny (720p)

그림 10. Full HD 영상에 대한 JM 18.5와 HM 10.1의 압축 성능 비교

과 동일한 조건으로 테스트가 가능하도록 제공된 설정 파일을 이용하였다. 실험은 1080p 영상인 Kimono 와 720p 영상인 Johnny 영상에 QP가 22, 27, 32, 37에 대해 실험 하였다. 실험 결과는 <그림 10>과 같다.

각 실험 에서 BDBR 결과는 H.264/MPEG-4 AVC와 비교했을 때 복잡한 Kimono 영상의 경우 44%, 정적인 Johnny 영상의 경우 53%의 PSNR 대비 부호화 향상을 가져왔다.

HEVC 표준화 회의의 실험 결과에 따르면 주관적 화질 대비 50% 이상의 부호화 효율을 달성 하였으며, 해상도가 높은 영상일수록 부호화 효율이 증가 하였다 [13].

## V. 결론

본고에서는 최근 표준화가 완성된 차세대 비디오 압축 부호화 규격인 HEVC의 특징 및 성능에 대해 소개 하였다. HEVC는 기존의 비디오 압축 코덱들과 비교하여 압축 성능을 크게 향상 시켰으며 향후 도래하는 UHD TV 방송이나 3D 비디오 부호화 응용에 널리 활용될 것으로 예상된다. 뿐만 아니라, 스마트폰 등 모바일 응용분야와 IPTV 등 스트리밍 분야에도 넓은 응용 분야를 확보할 것으로 예상된다.

## 참고 문헌

- [1] Sullivan, G. J., et al., "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1649–1668, Dec. 2012.
- [2] Kim, I. K., et al., "Block Partitioning Structure in the HEVC Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1697–1706, Dec. 2012.
- [3] Wiegand, T., et al., "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, No. 7, pp. 560–576, Jun. 2003
- [4] Lainema, J., et al., "Intra Coding of the HEVC Standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1792–1801, Dec. 2012.
- [5] JCT-VC, "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Last Call)," JCTVC-L1003\_v34, Geneva, Switzerland, Jan. 2013
- [6] Helle, P., et al., "Block Merging for Quadtree-Based Partitioning in HEVC," IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1720–1731, Dec. 2012.
- [7] Sze, V. and M. Budagavi, "High Throughput CABAC Entropy Coding in HEVC," IEEE Trans. Circuits Syst.

- Video Technol., vol. 22, No. 12, pp. 1778–1791, Dec. 2012.
- [8] Sole, J., et al., “Transform Coefficient Coding in HEVC,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1765–1777, Dec. 2012.
- [9] Andrey N., et al., “HEVC Deblocking Filter,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1746–1754, Dec. 2012.
- [10] Chih-Ming F., et al., “Sample Adaptive Offset in the HEVC Standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1755–1764, Dec. 2012.
- [11] <http://iphome.hhi.de/suehring/tml/download>
- [12] [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware)
- [13] Ohm, J. R., et al., “Comparison of the Coding Efficiency of Video Coding Standards – Including High Efficiency Video Coding (HEVC),” IEEE Trans. Circuits Syst. Video Technol., vol. 22, No. 12, pp. 1669–1684, Dec. 2012.

## 약 력



김재일

2006년 한국기술교육대학교 정보통신공학과 전자공학 학사  
 2006년~현재 한국과학기술원 정보통신공학과 석박사 통합과정  
 관심분야: 비디오 부호화, 주관적 화질 기반 변환 및 양자화 기술, 패턴 인식, 방송시스템



김문철

1989년 경북대학교 공학사  
 1992년 University of Florida, Gainesville, USA 공학석사  
 1996년 University of Florida, Gainesville, USA 공학박사  
 1997년~2001년 한국전자통신연구원 방송미디어 연구부 선임연구원  
 2001년~2009년 한국정보통신대학교 공학부 조교수/부교수  
 2009년~현재 한국과학기술원 전기및전자공학과 부교수  
 관심분야: 2D/3D Video Coding, Video Analysis and Understanding, Statistical Machine Learning, Patterning Recognition