

차세대 비디오 부호화를 위한 스크린콘텐츠 부호화 기술

□ 박지윤, 전병우 / 성균관대학교

I. 서론

MPEG HEVC(High Efficiency Video Coding)/ITU-T H.265 국제 표준화가 2013년 완료된 이후, HEVC 대비 2 배의 압축률 향상을 목표로 VVC(Versatile Video Coding)로 불리는 차세대 비디오 부호화 표준화가 ISO/IEC JTC1/SC29/WG11 MPEG과 ITU-T SG16/Q6 VCEG간의 공동 비디오 부호화 전문가그룹(Joint Video Experts Team: 이하 JVET)에 의하여 진행되고 있다. VVC는 HD/UHD/8K 비디오, 10 비트를 갖는 High Dynamic Range(HDR) 비디오에 대하여 HEVC 대비 압축률 향상이라는 전통적인 목표 외에도, 스크린콘텐츠 영상부호화(Screen Content Coding: SCC), Adaptive Resolution Change, Independent

Sub-pictures와 같은 다양한 기능의 부호화 표준을 목표로 하고 있다. VVC 표준화는 2018년 4월 샌디에고 회의에 제출된 기술기초의 비교평가로부터 시작하여, 2019년 7월 위원회 표준안(Committee Draft: CD)을 만들었으며, 2020년 7월 FDIS(Final Draft IS)의 완성을 목표로 진행되고 있다.

스크린콘텐츠 영상(SCC)이란, 카메라에 의해 취득된 영상이 아닌, 컴퓨터나 스마트폰과 같은 전자 기기에 의해서 인위적으로 생성된 영상이나 동영상 을 의미한다. JVET는 스크린콘텐츠 영상의 특성을 파악하여 이러한 종류의 영상에 대한 부호화 성능 향상을 위한 다양한 압축기술을 개발하고 있는 바, 본 고에서는 참고문헌 [1][2]를 중심으로, 차세대 비디오 표준 VVC가 고려하고 있는 최신 스크린콘텐츠

※ 본 연구는 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구(No. 2016-0-00572, 초고실감 미디어 서비스 실현을 위해 HEVC/3DA 대비 2배 압축을 제공하는 5세대 비디오/오디오 표준 핵심 기술 개발 및 표준화)

트(SCC) 부호화 기술을 Intra Block Copy(IBC), 변환생략(Transform Skip: TS), Palette 부호화 기술을 중심으로 살펴본다.

II. Intra Block Copy(IBC) 기술

Intra Block Copy(IBC) 기술은 SCC를 위한 HEVC 확장(HEVC Extended Version) 표준에서도 이미 채택된 바 있는 기술로써, 스크린콘텐츠의 부호화 효율을 크게 개선시키는 것으로 잘 알려져 있다. VVC에서 표준화를 추진하고 있는 IBC 기술은 다음과 같다. 먼저, IBC 기술이 사용되는 블록은 IBC 모드로 신호되며 부호화 복잡도를 줄이기 위하여, 휘도신호 기준으로 너비와 높이가 모두 64 화소 이하인 CU에만 IBC 모드가 적용될 수 있도록 제한되었다. IBC 모드를 사용할 경우, 부호화기는 블록매칭(Block Matching: BM) 수행을 통하여 부호화되는 CU와 가장 유사한 블록을 찾은 후, 그 공간 위치를 가리키는 벡터(이를 블록벡터(Block Vector: BV)라고 한다)를 찾는다. 블록벡터는 이미 부호화되어 복호화기에서도 그대로 재현이 가능한 참조 화소(reference sample)들로 이루어진 참조블록들 중 현재 블록과 가장 유사한 블록까지의 공간변위를 나타낸다. IBC 기술에서 사용하는 BV는 휘도 및 색차신호 모두에 대하여 일반적으로 정수 정밀도를 사용한다(색차신호의 경우는 정수단위로 정수화하여 사용한다). 하지만, 가변의 벡터 정밀도를 사용하는 AMVR(Adaptive Motion Vector Resolution) 기술과 같이 사용되는 경우 IBC 모드는 1-pel과 4-pel 움직임벡터 정밀도 간에 절환되어 사용될 수 있다. IBC 모드를 갖는 CU는 화면내(Intra predicted) 또는 화면간 예측(Inter

Predicted) 모드가 아닌, 제3 예측 모드로 취급된다.

1. IBC를 위한 부호화 동작과 신호방법

VVC 부호화기는 BV를 탐색하기 위하여 Hash 기반 추정방법을 사용한다. 또한, CU의 폭(W) 또는 높이(H)가 휘도신호 기준으로 16보다 크지 않은 블록에 대해서는 윌왜곡(RD) 최적화를 수행한다. 비병합(non-merge) 모드의 경우, 먼저 Hash기반 추정을 사용하여 블록벡터를 탐색하는데, 만일 Hash기반 탐색을 통하여 유효한 BV후보를 찾지 못한 경우에는 블록매칭 기반의 지역 검색(block matching-based local search)을 수행한다. Hash 기반 BV 탐색 시, IBC가 허용되는 모든 블록 크기에 대하여, 현재 블록과 참조블록 사이의 Hash 키(32 비트 CRC) 매칭을 수행한다. 한편, Hash 키는 4x4 서브 블록 단위로 계산되며, 4x4보다 큰 블록의 Hash 키 매칭은 해당블록과 참조블록의 모든 4x4 서브 블록별로 Hash 키가 모두 일치하는지 여부를 확인한다. Hash 키가 매칭되는 참조블록이 하나 이상의 경우에는, 각 참조블록의 BV 벡터에 대한 코스트들을 비교하여 가장 최소 코스트에 해당하는 BV가 최종 결정된다. 한편, 블록매칭(BM)을 위한 탐색범위는 이전 CTU와 현재 CTU를 모두 포함하도록 설정된다. 또한, IBC 모드는 CU별로 다음과 같이 IBC AMVP 모드, 또는 IBC 스킵/병합 모드 중의 하나로 결정되며, 이 결정정보는 비트스트림 상의 정보로 신호된다.

- IBC 스킵/병합 모드 : IBC 모드로 부호화된 인접 후보 블록들의 BV 중 어느 것이 현재 CU의 BV로 사용되었는지를 병합(merge) 후보의 해

당 인덱스로 신호하는 모드이다. 이때, 병합 목록(merge list)은 공간, HMVP(History-based Motion Vector Prediction) 및 pairwise 후보로 구성된다.

- IBC AMVP 모드 : 이 모드에서는 BV벡터간의 차이를 움직임벡터 차이를 부호화하는 것과 같은 방식으로 부호화한다. 이때, BV 예측자로는 IBC로 부호화된 좌측 이웃 및 위쪽 이웃의 BV를 사용하며, 만일, IBC조건에 맞는 이웃이 없는 경우에는 Default BV벡터가 예측자로 사용된다. 선택된 BV벡터 예측자는 플래그 정보로 신호된다.

2. IBC 부호화를 위한 참조영역

메모리 소비 및 복호화기의 복잡성을 줄이기 위해 VTM6에 구현된 IBC기술은 현재 CTU 영역과 왼쪽 CTU의 일부 영역만이 포함될 수 있는, 가상 버퍼(virtual Buffer)라고 불리우는 미리 정의된 영역내에서만 IBC 참조블록을 가져오도록 규정하고 있다. 즉, BV가 지정하는 블록의 모든 화소들은 반드시 가상버퍼 내부에 존재하여야 한다는 것을 규정함으로써, IBC 모드가 사용하는 블록벡터가 가상 버퍼내의 유효한 블록을 가리키는지 여부를 확인하여 표준정합성 여부를 손쉽게 알 수 있도록 하고 있다. ibcBuf로 표시되는 가상버퍼는 CTU의 한 변의 길이를 ctbSize라고 할 경우, 너비(wIbcBuf)가 “ $128 * 128 / \text{ctbSize}$ ”이고 높이(hIbcBuf)는 ctbSize로 규정된다. 따라서, ibcBuf, 즉, 가상버퍼가 수용 가능한 총 화소수는 항상 128×128 로 설정되며, 주어진 CTU 크기에 따라, 가상버퍼의 높이는 CTU의 높이(즉, ctbSize)와 항상 동일하지만, 그 폭은 $wIbcBuf = 128 * 128 / \text{ctbSize}$ 가 된다. 예를

들어, CTU 크기가 128×128 인 경우 ibcBuf의 크기도 128×128 이며, 이때, 가상버퍼는 부호화하는 CTU만을 포함한다. 만일, CTU 크기가 64×64 인 경우에는, ibcBuf의 크기는 256×64 이며, 이 경우 현재 CTU 및 이전 3개의 CTU까지 포함할 수 있다. CTU 크기가 32×32 인 경우 ibcBuf의 크기는 512×32 이다. IBC를 위한 가상 버퍼 ibcBuf는 다음과 같이 유지된다. 여기서, 현재 픽처내 서로 겹치지 않는 단위로 정의되는 VPDU(Virtual Pipeline Data Unit)의 크기는 가로 세로 각각 $\min(\text{ctbSize}, 64)$ 이며, $W_v = \min(\text{ctbSize}, 64)$ 이다.

- 1) 각 CTU Row의 복호화 시작 시, -1의 값(유효하지 않은 값을 나타냄)으로 ibcBuf를 초기화한다.

- 2) 픽처의 좌측 상단기준 상대좌표 (xVPDU, yVPDU)에 해당하는 VPDU의 복호화 시작 시, ibcBuf는 다음과 같이 설정된다.

$$\begin{aligned} \bullet \text{ ibcBuf}[x][y] &= -1 \text{ where} \\ x &= xVPDU \% wIbcBuf, \dots, xVPDU \% wIbcBuf + W_v - 1; \\ y &= yVPDU \% \text{ctbSize}, \dots, yVPDU \% \text{ctbSize} + W_v - 1 \end{aligned}$$

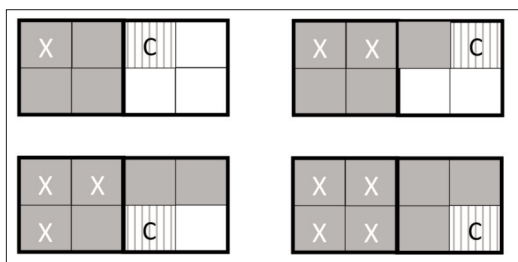
- 3) 픽처의 좌측 상단기준 상대좌표 (x,y)에 해당하는 CU복호화 후, ibcBuf는 다음과 같이 설정된다.

$$\begin{aligned} \bullet \text{ ibcBuf}[x \% wIbcBuf][y \% \text{ctbSize}] \\ = \text{recSample}[x][y] \end{aligned}$$

또한 좌표 (x, y)에 대응하는 블록의 블록벡터 $bv = (bv[0], bv[1])$ 에 대하여, 다음이 참이면 유효한 BV, 그렇지 않으면 유효하지 않은 BV인 것으로 판단한다.

* $\text{ibcBuf}[(x + \text{bv}[0]) \% \text{wIbcBuf}][(y + \text{bv}[1]) \% \text{ctbSize}]$ 는 -1과 같지 않다.

〈그림 1〉은 IBC 모드에서 사용할 수 있는 가상버퍼 영역을 보여준다. 여기서, 각 블록은 64x64 휘도 화소 단위를 나타낸다.



〈그림 1〉 현재 CTU 처리 순서와 현재 및 왼쪽 CTU에서 사용 가능한 IBC 참조블록

1) 현재 블록이 현재 CTU의 왼쪽 상단 64x64 블록에 속할 경우, 이 블록은 IBC부호화를 위하여 현재 CTU 내의 이미 부호화된 화소들에 추가하여 다음 위치 내의 블록들을 IBC 참조블록으로 사용할 수 있다.

- 왼쪽 CTU의 오른쪽 아래 64x64 블록
- 왼쪽 CTU의 왼쪽 아래 64x64 블록
- 왼쪽 CTU의 오른쪽 위 64x64 블록

2) 현재 블록이 현재 CTU의 오른쪽 상단 64x64 블록에 속할 경우, 휘도 위치 (0, 64)가 아직 부호화되지 않았다면, 이 블록은 IBC부호화를 위하여 현재 CTU 내의 이미 부호화된 화소들에 추가하여 다음 위치 내의 블록들을 IBC 참조블록으로 사용할 수 있다.

- 왼쪽 CTU의 오른쪽 아래 64x64 블록
- 왼쪽 CTU의 왼쪽 아래 64x64 블록

또한 그렇지 않았다면, 이 블록은 왼쪽 CTU의 오른쪽 아래 64x64 블록을 IBC 참조블록으로

사용할 수 있다.

3) 현재 블록이 현재 CTU의 왼쪽 하단 64x64 블록에 속할 경우, 휘도 위치 (0, 64)가 아직 부호화되지 않았다면, 이 블록은 IBC부호화를 위하여 현재 CTU 내의 이미 부호화된 화소들에 추가하여 다음 위치 내의 블록들을 IBC 참조블록으로 사용할 수 있다.

- 왼쪽 CTU의 오른쪽 위 64x64 블록

- 왼쪽 CTU의 오른쪽 아래 64x64 블록

또한 그렇지 않았다면, 이 블록은 왼쪽 CTU의 오른쪽 아래 64x64 블록을 IBC 참조블록으로 사용할 수 있다.

4) 현재 블록이 현재 CTU의 오른쪽 하단 64x64 블록에 속할 경우에는 IBC부호화를 위하여 현재 CTU내의 이미 부호화된 화소들만을 IBC 참조를 위하여 사용할 수 있다.

이러한 제한을 두는 이유는 Chip내에 존재하는 Local Memory만을 사용하여 IBC가 가능하도록 하여, 더욱 효율적인 하드웨어 구현을 할 수 있도록 하기 위함이다.

3. IBC와 다른 부호화 기술간의 상호작용

IBC 모드와 VTM6의 다른 인터 부호화 기술(예를 들어 pairwise merge candidate, 히스토리 기반 모션 벡터 예측(HMVP), 인트라/인터 병합예측 모드(CIIP: Combined Intra/Inter Prediction), MMVD(Merge mode with Motion Vector Difference), 삼각분할(Triangle Partition) 등)과의 상호작용은 다음과 같다.

– IBC는 pairwise merge candidate 및 HMVP

와 함께 사용할 수 있다. 새로운 pairwise IBC 병합 후보는 두 개의 IBC 병합 후보를 평균하여 생성할 수 있으며, HMVP 기술을 사용할 경우 향후 참조를 위해 IBC BV가 히스토리 버퍼에 삽입된다.

- IBC 기술은 Affine motion, CIIP, MMVD 및 삼각분할을 사용하는 화면간 예측 기술과는 같이 사용되지 않는다.
- DUAL_TREE 분할구조가 사용되는 색차블록에는 IBC 사용되지 않는다.

VVC의 IBC 기술은 HEVC 확장버전에서 사용하는 IBC 기술과는 다른 점들이 많이 있다. 예를 들어, VVC의 IBC 기술은 참조 픽처 리스트 0(List0)에 더 이상 현재 픽처를 포함시키지 않는다. 또한, VVC의 IBC기술은 BV 벡터 부호화를 수행할 때 화면 간 예측 모드로 부호화된 인접 블록들을 고려하지 않는다. 또한, 이 반대의 경우, 즉, 화면간 예측 모드로 부호화된 블록의 움직임벡터 부호화 시 인접 블록들의 BV 벡터들을 역시 고려하지 않는다. VVC의 IBC 기술은 또한 다음과 같이 설계되었다.

- IBC는 pairwise merge 후보 및 History 기반 모션 예측자를 포함한 일반 움직임벡터 병합과 동일한 프로세스를 공유하지만, TMVP 및 제로 벡터는 IBC모드와 함께 사용될 수 없기 때문에 허용하지 않는다.
- 별도의 HMVP 버퍼(각 후보 5명)는 기존 움직임 보상 및 IBC에 사용된다.
- BV벡터 제약은 비트스트림 적합성 제약의 형태로 구현된다. 따라서, 부호화기는 비트스트림에 유효하지 않은 벡터가 존재하지 않도록

하여야 하며, 병합후보가 유효하지 않은 경우(즉, 범위를 벗어났거나 0인 경우) 병합을 사용하지 않아야 한다. 이러한 비트스트림 적합성 제약은 이미 설명된 바와 같이 가상 버퍼에 대한 제한으로 설정되었다.

- IBC 모드로 부호화된 블록은 인터 모드와 같은 방식으로 디블록킹된다.
- 현재 블록이 IBC 모드로 부호화된 경우 AMVR은 1/4-pel을 사용하지 않는다. 대신, AMVR은 움직임벡터가 1-pel인지 또는 4-pel 인지를 나타내도록 신호된다.
- IBC 병합 후보의 수는 슬라이스 헤더에서 일반, 서브 블록 및 삼각병합 후보의 수와 별도로 신호될 수 있다.

III. 변환 생략(Transform Skip) 기술

변환생략기술은 HEVC 표준에도 반영된 기술로써, 특히 스크린콘텐츠 부호화에 효과가 큰 기술로 알려져 왔다. 변환생략기술이 적용되는(인터 또는 인트라) 잔차블록에 대하여는 DCT/DST 와 같은 변환이 적용되지 않아 화소공간에서의 잔차값이 부호화된다. 또한, 이 잔차화소값들은 일반적인 변환적용으로 얻어지는 변환계수와는 다른 방법을 사용하여 부호화된다. VVC를 위한 VTM6에서는 $MaxTsSize \times MaxTsSize$ 크기보다 크지 않은 휘도 블록에 변환생략기술을 사용할 수 있도록 설계되었다. $MaxTsSize$ 값은 변환생략기술이 사용될 수 있는 최대 블록크기를 나타내는 것으로써, PPS (Picture Parameter Set) 레벨에서 신호된다. VTM6의 경우 이 값은 32이다. 주어진 블록에 대하여 변환이 사용되는지 또는 생략되는지 여부는 변

환생략플래그를 신호된다.

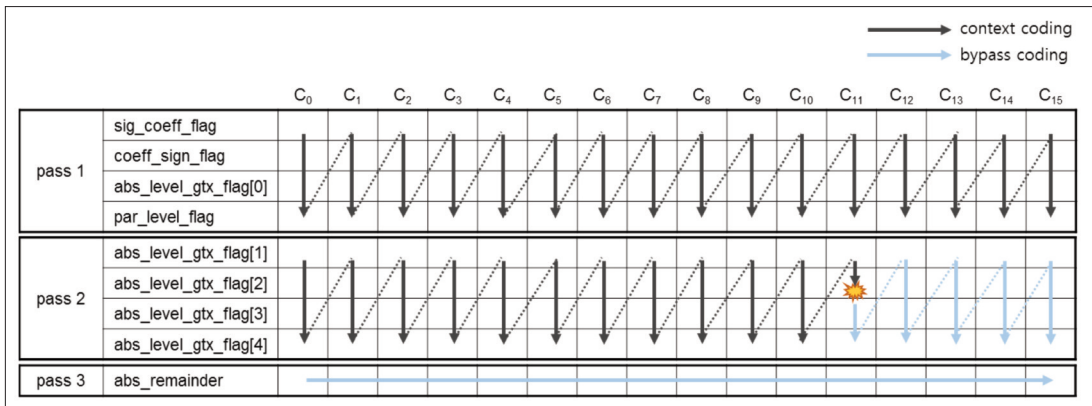
변환생략 모드가 적용되는 잔차신호 역시 변환 적용시와 마찬가지로, 4x4 크기의 상호중복되지 않은 서브 블록 단위로 변환생략 잔차 부호화 프로세스를 사용하여 양자화되고 부호화되는데, 이 프로세스는 더 나은 부호화 효율을 위해 일반적으로 사용하는 변환 계수 부호화 방법과는 다르게 설계되었다. 다음은 VTM6에서 사용하는 변환 생략 부호화와 정규 변환 잔차 부호화의 차이점을 요약한 것이다.

- ① 변환 블록내 4x4서브 블록들 및 4x4 서브블록 내의 화소위치들은 공히 순방향 스캐닝 순서에 따라 스캔된다.
- ② 마지막 화소에 대한 위치 (x,y) 신호는 없다.
- ③ coded_sub_block_flag는 모든 서브 블록에 대해 부호화된다. 단, 모든 이전 서브블록의 coded_sub_block_flag가 0일 때 마지막 서브블록에 대하여 coded_sub_block_flag는 전송되지 않는다.
- ④ sig_coeff_flag 컨텍스트 모델은 상단 및 왼쪽 이웃의 해당 값에 영향을 받으며, 축소된 템플릿을

릿을 사용한다.

- ⑤ abs_level_gt1 컨텍스트 모델 역시 상단 및 왼쪽 이웃의 sig_coeff_flag 값에 영향을 받으며, 축소된 템플릿을 사용한다.
- ⑥ par_level_flag는 하나의 컨텍스트 모델만을 사용한다.
- ⑦ abs_level_gt3, gt5, gt7, gt9 flag를 사용하여 계수레벨을 표시하며, gt3, gt5, gt7, gt9 flag들은 각각 한 개의 컨텍스트를 사용한다.
- ⑧ 나머지(remainder) 값의 이진화를 위해 수정된 rice 매개 변수를 도출한다.
- ⑨ coeff_sign_flag의 컨텍스트 모델은 왼쪽 및 상단의 해당 값을 기반으로 결정된다.
- ⑩ coeff_sign_flag 는 모든 컨텍스트 부호화된 bin을 함께 유지하기 위해 sig_coeff_flag 뒤에 파싱된다.

각 서브블록에 대해, coded_subblock_flag가 1인 경우(즉, 서브블록 내에 0이 아닌 양자화된 잔차신호가 적어도 하나 존재하는 경우), 양자화된 잔차레벨의 부호화는 <그림 2>에 보여진 3개의 스캔 패



<그림 2> 변환 생략 블록에 대한 잔차 부호화

스를 통하여 수행되며, 각 패스는 다음과 같다.

- 첫번째 스캔: 이 스캔을 통하여 `sig_coeff_flag`, `coeff_sign_flag`, 레벨의 절대값이 1보다 큰 값을 나타내는 `abs_level_gtx_flag[0]` 플래그 및 패리티(`par_level_flag`)가 부호화된다. 주어진 스캔 위치에 대해 `sig_coeff_flag`가 1과 같으면 `coeff_sign_flag`가 부호화된다. 그리고 그 다음 `abs_level_gtx_flag[0]`(레벨의 절대값이 1보다 큰지 여부를 지정함)이 부호화된다. `abs_level_gtx_flag[0]`이 1인 경우, `par_level_flag`가 추가로 부호화되어 레벨의 절대값의 패리티를 지정한다.
- Greater than x 스캔: 레벨의 절대값이 1보다 큰 각 스캔 위치에 대해, 주어진 위치에서의 절대 레벨이 각각 3, 5, 7 또는 9보다 큰지를 나타내기 위해 최대 4개의 `abs_level_gtx_flag[i]`가 부호화된다 여기서 $i = 1, \dots, 4$ 이다.
- Remainder 스캔: 레벨의 절대값에 대한 remainder 값은 `abs_level_gtx_flag[4]`가 1인 모든 스캔 위치에(즉, 레벨의 절대값이 9보다 큰 경우에 해당) 대해 부호화된다.

스캔 패스 # 1 및 # 2에서(첫 번째 스캔 패스 및 Greater than x 스캔 패스) 발생하는 bin은 한 TU당 허용하는 최대 context coded bin수가 다 소진될 때까지 컨텍스트 부호화된다. 하나의 TU당 허용하는 최대 context coded bin수는 $2x \text{ block_width} \times \text{block_height}$ 이며, 이는 평균적으로 한 화소 위치당 2개의 context coded bin에 해당한다. 마지막으로 수행되는 remainder 스캔 패스를 통하여 발생하는 bin들은 bypass 부호화된다.

또한, BDPCM 모드로 부호화되지 않은 블록이 변환생략 부호화될 경우 level mapping이 적용된 후, 변환 생략 잔차 부호화가 수행된다. level mapping은 시그널링 비용을 줄이기 위해 상단 및 좌측 인접 계수 레벨값을 참조하여 현재의 계수 레벨을 예측한다. 주어진 절대값이 취해진 잔차신호 계수 레벨에 대해, level mapping이 적용되기 전의 값을 `absCoeff`이라 하고, level mapping 후의 값을 `absCoeffMod`라고 하자. 또한, X_0 은 좌측 인접 위치의 절대 계수 레벨, 그리고 X_1 은 위쪽 인접 위치의 절대 계수 레벨을 나타낼 때, level mapping은 다음과 같이 수행된다.

```
pred = max(X0, X1);
if (absCoeff == pred)
    absCoeffMod = 1;
else
    absCoeffMod = (absCoeff < pred) ?
        absCoeff + 1 : absCoeff;
```

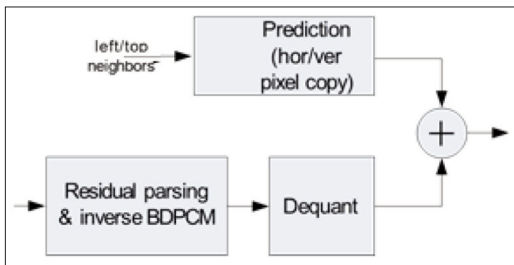
그런 다음, `absCoeffMod` 값은 전술한 바와 같이 부호화된다.

IV. Block-based Delta Pulse Code Modulation(BDPCM) 기술

VTM6은 스크린콘텐츠 부호화 기술의 하나로 BDPCM(Block Differential Pulse Coded Modulation) 기술을 지원한다. BDPCM 기술의 사용가능여부는 시퀀스 레벨인 SPS(Sequence Parameter Set)의 `BDPCM_enable_flag`에 의하여 지시되는데, 이 플래그는 SPS를 통하여 변환생략

(TS)모드가 활성화된 경우에만 신호된다. BDPCM이 활성화(즉, BDPCM_enable_flag=1)된 경우, CU 크기가 휘도기준으로 MaxTsSize(여기서, MaxTsSize는 변환생략 모드가 허용되는 최대 블록 크기이다)보다 크지 않은 경우, 인트라 부호화된 CU에 대하여 CU 레벨 플래그로 전송된다. 이 플래그는 주어진 CU가 인트라 부호화되었는지 또는 BDPCM으로 부호화되었는지를 지시하며, 또한, BDPCM이 사용되는 경우, BDPCM의 예측방향(즉, 수직 또는 수평방향)을 지시하는 플래그가 또한 전송된다.

BDPCM을 적용할 경우, 필터링되지 않은 참조 화소들을 사용하며, 수평(Horizontal) 또는 수직(Vertical) 방향 인트라 예측을 수행하여 예측잔차 신호를 계산하며, 이 예측잔차신호는 양자화된다. 또한, BDPCM된 현재 블록과 BDPCM된 이전블록의 양자화된 예측잔차신호 간의 차이가 부호화된다. 블록의 크기가 M(높이) × N(폭)이며, 예측잔차신호 블록을 R이라고 하자. 또한, 이 R 블록내 (i,j) 화소 위치 ($0 \leq i \leq M-1$, $0 \leq j \leq N-1$)에서의 예측잔차신호 값을 $r_{i,j}$ 이라고 하고, 이 잔차신호의 양자화된 값을 $Q(r_{i,j})$ 이라고 하자.



〈그림 3〉 BDPCM 적용방법

〈그림 3〉과 같이 양자화된 잔차 값에 BDPCM을 적용하여 얻어진 (i,j) 위치에서의 신호를 $\tilde{r}_{i,j}$ 라고 할

때, 이 신호 $\tilde{r}_{i,j}$ 는 다음과 같이 계산된다.

- 수직 BDPCM 예측의 경우

$0 \leq j \leq (N-1)$ 에 대하여, $\tilde{r}_{i,j}$ 는 다음과 같이 계산된다.

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & i = 0 \\ Q(r_{i,j}) - Q(r_{(i-1),j}), & 1 \leq i \leq (M-1) \end{cases} \quad (1)$$

- 수평 BDPCM 예측의 경우

$0 \leq i \leq (M-1)$ 에 대하여, $\tilde{r}_{i,j}$ 는 다음과 같이 계산된다.

$$\tilde{r}_{i,j} = \begin{cases} Q(r_{i,j}), & j = 0 \\ Q(r_{i,j}) - Q(r_{i,(j-1)}), & 1 \leq j \leq (N-1) \end{cases} \quad (2)$$

복호화기 측에서, 상기 프로세스는 다음과 같이 $Q(r_{i,j})$, $0 \leq i \leq M-1$, $0 \leq j \leq N-1$ 를 역으로 계산한다.

$$Q(r_{i,j}) = \sum_{k=0}^i \tilde{r}_{k,j}, \text{ 수직 BDPCM 사용 경우} \quad (3)$$

$$Q(r_{i,j}) = \sum_{k=0}^j \tilde{r}_{i,k}, \text{ 수평 BDPCM 사용 경우} \quad (4)$$

역 양자화된 잔차신호, $Q^{-1}(Q(r_{i,j}))$ 는 복원화소 값을 생성하기 위해 인트라 블록 예측 값에 더해진다. BDPCM을 적용하여 얻어진 신호 $\tilde{r}_{i,j}$ 은 변환 생략 모드 잔차 부호화에서와 동일한 잔차 부호화 프로세스를 사용하여 부호화된 후 복호화기로 전송된다. 또한, BDPCM에 사용한 수평 또는 수직 예측 모드는 그 이후의 인트라 모드 부호화를 위한 MPM(Most Probable Mode) 모드 결정에 사용할 수 있도록 하기 위하여 저장된다. 또한, 디블록킹 필터링을 적용할 경우, 블록 경계의 양쪽에 있는 두 블록 모두가 BDPCM 부호화되었다면, 이 두 블록 사이의 블록 경계는 디블록킹 필터링되지 않는다.

V. Palette부호화 기술

VTM6은 휘도 기준으로 64x64보다 작거나 같은 4:4:4의 컬러포맷을 지닌 CU의 화면내 예측부호화를 위하여 팔레트(Palette)부호화 기술을 사용할 수 있다. 이 부호화 기술은 IBC 및 변화생략기술과 함께 스크린콘텐츠 부호화에 매우 효과적이다. 팔레트 모드가 사용되는 경우 이를 알리는 플래그가 CU 레벨에서 전송된다. 팔레트부호화된 CU는 화면 내 예측, 화면 간 예측 및 인트라 블록 카피(IBC) 모드와는 다른 또 하나의 예측 모드로서 취급된다. 팔레트 모드 사용시 사용하는 팔레트는 현재 CU의 화소 값을 표현하기 위한 몇 개의 대표 색상 값 세트로 구성된다.

팔레트 색상에 가까운 값을 가진 화소는 원래의 화소 값 대신, 팔레트 인덱스 신호로 표시된다. 또한, 이스케이프(Esc) 기호와 양자화된 color component 값으로 구성된 별도의 신호 방법을 사용하여 팔레트에 포함되지 않은 화소값들을 표현할 수도 있다. 팔레트부호화의 개념이 <그림 4>에 설명되어 있다.

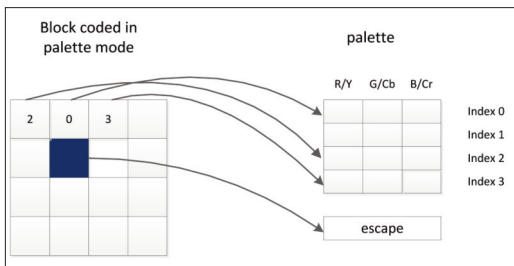
팔레트부호화에서도 역시 팔레트 예측자를 사용한다. 이 예측자는 구체적으로 다음과 같이 설정된다. wavefront 기술을 사용하지 않는 경우 각 슬라

이스의 시작 시, wavefront 기술을 사용하는 경우 각 CTU row의 시작 시, 이 예측자(predictor)는 0으로 초기화된다. 팔레트 예측자에 대하여, 이 예측자가 현재 CU에 대한 팔레트에 있는 것인지 여부가 reuse플래그로 신호된다. 또한, reuse플래그들은 run-length coding을 사용하여 부호화된다. 그 이후, 새로운 팔레트 엔트리의 수 및 새로운 팔레트 엔트리의 컬러 컴포넌트 값이 신호된다.

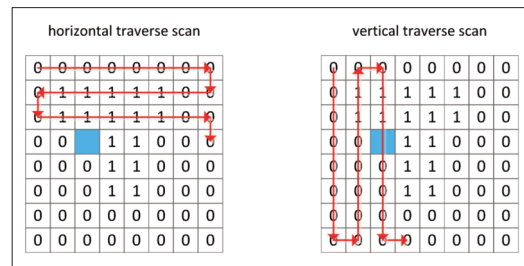
부호화기는 주어진 CU를 팔레트모드로 부호화한 후, 팔레트 예측자를 현재 팔레트를 사용하여 업데이트한다. 현재 팔레트에서 재사용되지 않는 이전 팔레트 예측자의 엔트리들은 허용된 최대 크기에도 달할 때까지 새 팔레트 예측자 다음 위치에 추가된다. ESC 플래그는 현재 CU에 ESC심볼이 존재하는지를 나타내기 위해 각 CU에 대해 escape flag가 신호된다. ESC심볼이 있으면 팔레트 테이블은 한 칸 증가된 후, 마지막 인덱스가 ESC경우로 지정된다.

CU 내 각 화소에 대한 팔레트인덱스들을 영상형태로 표시한 것을 팔레트 인덱스 맵이라고 부른다. 이 팔레트 인덱스 맵은 <그림 5>와 같이 가로 및 세로 트래버스(traverse) 스캔을 통하여 부호화된다. 이때, palette_transpose_flag를 사용하여 선택된 스캔 순서는 비트스트림에 표시한다.

팔레트 인덱스는 두 가지 주요 팔레트 화소 모드



<그림 4> 팔레트 모드로 부호화된 블록의 예



<그림 5> 수형 및 수직 트래버스(traverse) 스캔

인 “INDEX”와 “COPY_ABOVE”를 사용하여 부호화되며, 선택된 팔레트 화소모드는 플래그값으로 신호된다. 그러나, 수평 traverse 스캔을 사용할 경우, 맨 위 행, 수직 traverse 스캔을 사용할 경우, 첫 번째 열, 또는 이전 모드가 “COPY_ABOVE”인 경우, 이렇게 3가지 경우에는 그렇지 않다. “COPY_ABOVE” 모드의 경우에는 행의 위(ROW ABOVE)에 있는 화소의 팔레트 인덱스가 복사되어 사용되거나, “INDEX”모드에서의 팔레트 인덱스는 명시적(explicit)으로 신호된다. “INDEX” 및 “COPY_ABOVE” 모드 공히, 동일한 모드로 연속하여 부호화되는 화소수(즉, run) 값을 부호화한다.

인덱스 맵의 부호화 순서는 다음과 같다.

- ① “INDEX” 모드로 부호화되는 연속된 화소의 수를 알리는 index run 값에 해당되는 인덱스 값의 수를 알린다.
- ② 팔레트 모드(INDEX 또는 COPY_ABOVE)와 각 run에 해당하는 run length값이 순차교차(interleaved) 방식으로 신호된다.
- ③ 마지막으로, escape 모드에 해당하는 전체 CU에 대한 양자화된 컬러컴포넌트 값들이 그룹화된 후, Exponential Golomb 방식으로 부호화된다.

휘도와 색차 성분간의 CU partitioning이 상이한

Dual Tree를 사용하는 슬라이스의 경우 팔레트는 휘도(Y 구성 요소)와 색차(Cb 및 Cr 구성 요소)에 개별적으로 적용된다. 또한, 단일트리(Single Tree)의 경우 팔레트는 Y, Cb, Cr 구성 요소에 공히 함께 적용된다. 즉, 팔레트의 각 엔트리에는 Y, Cb, Cr 값이 함께 포함된다. 한편, 디블록킹 수행 시, 블록 경계의 한쪽 측면에 있는 팔레트 부호화된 블록에 대하여는 디블록킹 필터링을 적용하지 않는다.

VI. 결 론

본 고에서는 참고문헌 [1][2]를 중심으로 스크린콘텐츠 부호화에 효과적인 VVC 부호화 기술로 알려져 있는 IBC 기술, 변환생략부호화 기술, 팔레트부호화 기술에 대하여 알아보았다. 2019년 7월 위원회 표준안(Committee Draft: CD)이 만들어진 이후, 다소간의 기술변화나 정리가 아직은 추가로 더 있을 수는 있으나, 2020년 7월 FDIS(Final Draft IS)의 완성을 목표로 하고 있는 점으로 미루어 볼 때, VVC 표준을 위한 대부분의 기술들은 상당한 안정화에 이른 것으로 판단된다. 본 고에서의 기술소개가 이 분야의 연구개발자에게 많은 도움이 되기를 바라며, 본 고에서 미흡한 점은 참고문헌 [1][2]와, 향후 JVET표준화 회의의 관련된 문서들을 참조하기를 권한다.

참고문헌

- [1] B. Bross, J. Chen, S. Liu, "Versatile Video Coding (Draft 6)", JVET-O2001, Joint Video Experts Team(JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Gothenburg, SE, July 2019.
- [2] J. Chen, Y. Ye, S. Kim, "Algorithm description for Versatile Video Coding and Test Model 6 (VTM 6)", JVET-O2002, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, Gothenburg, SE, July 2019.

필자소개



박지윤

- 2016년 8월 : 국민대학교 정보통신대학 컴퓨터공학부 학사
- 2017년 9월 ~ 현재 : 성균관대학교 전자전기컴퓨터공학과 석박사통합과정
- 주관심분야 : 멀티미디어 영상압축



전병우

- 1985년 2월 : 서울대학교 전자공학과 학사
- 1987년 2월 : 서울대학교 전자공학과 석사
- 1992년 12월 : Purdue Univ. School of Electrical Eng., 공학박사
- 1993년 ~ 1997년 : 삼성전자 신호처리연구소 선임/수석연구원
- 1997년 9월 ~ 현재 : 성균관대학교 전자전기공학부 교수
- 주관심분야 : 영상신호처리, 패턴인식, 멀티미디어