

VVC 표준화 기술 동향: 변환 기술

□ 이범식 / 조선대학교

1. 서론

2013년 HEVC 비디오 코덱 제정 이후, 4K, 8K를 비디오 영상을 이용한 실감 영상 및 가상 현실 서비스 등이 확대됨에 따라 HEVC 대비 2배 이상 성능 개선을 목표로 하는 차세대 비디오 코덱인 Versatile Video Coding(VVC) 표준화 작업을 착수하게 되었고 현재는 표준화 작업이 활발히 진행 중이다. VVC는 비디오 부호화 표준화 그룹인 ISO/ICE MPEG(Moving Picture Experts Group)과 ITU-T VCEG(Video Coding Experts Group)이 공동으로 구성한 JVET(Joint Video Exploration Team)에서 HEVC 대비 2배 이상의 부호화 압축 성능 개선을 목표로 개발 중이다. VVC 표준화는 2018년 1월 121차 광주 MPEG, 9차 JVET 회의에서 제안 요청

서(Call for proposal)를 발표하였고, 122차 샌디에고 MPEG 및 10차 JVET회의에서 총 23개의 기관들이 비디오 코덱 기술을 제안함으로써 본격적인 비디오 표준화가 시작되었다. 122차 MPEG 및 10차 JVET 회의에서는 각 기관으로부터 제안된 비디오 코덱 기술들에 대한 기술 검토와 객관적 압축 성능 및 주관적 화질 평가를 수행하였으며, 여러 기술들 중에서 일부를 채택하여 Working Draft(WD) 1.0과 비디오 참조 소프트웨어인 VTM(VVC Test Mode) 1.0을 발표하였다. VVC 표준은 2019년 7월 127차 MPEG 및 15차 JVET회의 종료 이후 Committee Draft(CD)가 완성되었으며, 2020년 10월 Final Draft International Standard(FDIS)를 제정하는 것을 목표로 표준화가 진행 중이다.

기존 HEVC에서 쿼드트리(Quadtree)로 계층적

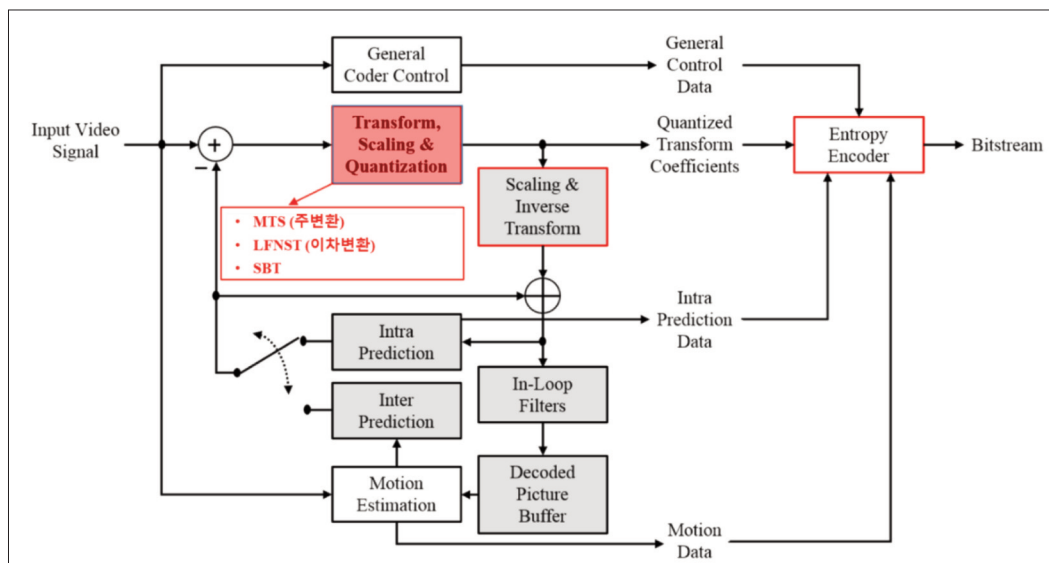
※ 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2017R1C1B1008646).

으로 분할하는 부호화 구조에서 VVC에서는 QTBT(QuadTree Binary Tree)와 TT(Ternary Tree)를 결합한 분할 블록 구조를 채택하였다. 이것은 HEVC 대비 보다 유연하게 예측 잔차신호를 발생하거나 처리할 수 있도록 하여, HEVC 대비 보다 향상된 압축 성능을 가져올 수 있었다. 이러한 기본 블록 구조 이외에, 적응적 루프 필터 기술(ALF, Adaptive Loop Filter), 움직임 예측 기술로서 Affine Motion Prediction(AMP), 디코더 움직임 보정 기술(DMVR, Decoder-side Motion Vector Refinement) 등 기존 코덱에서는 사용되지 않았던 새로운 기술들이 표준 기술로 채택되었다. 변환 및 양자화 기술로서는 기존 비디오 코덱에서 주로 사용되었던 변환 커널인 DCT-2가 계속적으로 이용되고 있으며 적용 블록의 크기가 더 큰 블록 크기까지 적용하는 것으로 변경되었다. 또한 기존 HEVC에서 4×4 와 같은 작은 변환 블록에 대하여 적용되어 왔던 DST-7 커널을 큰 변환 블록까지 확대되었

으며, 새로운 변환 커널인 DCT-8까지 변환 커널로 추가되었다. 또한 기존 비디오 코덱 대비 가장 큰 변화 중의 하나는 이차 변환의 채택이라고 할 수 있다. DCT-2, DST-7, DCT-8로 변환되는 변환 과정을 주변환(Primary Transform)이라고 하고 이 변환으로 얻는 변환 계수에 다시 한번 변환을 수행하는 것을 이차변환(Secundary Transform)이라고 한다. 이렇게 VVC에서는 기존 비디오 코덱에서 사용되어 왔던 기술을 사용한 부분도 있지만, 그 새롭게 도입되거나 구조를 크게 변경하는 기술도 있다. 본 논문에서는 비디오 코덱의 여러 기술 중에서 변환 및 양자화기술로서 VVC에서 채택되어 최종 표준 기술로 고려되고 있는 기술들에 대해서 기술한다.

II. 변환 기술

HEVC에서는 CU블록 내에서 변환 커널이 계층적



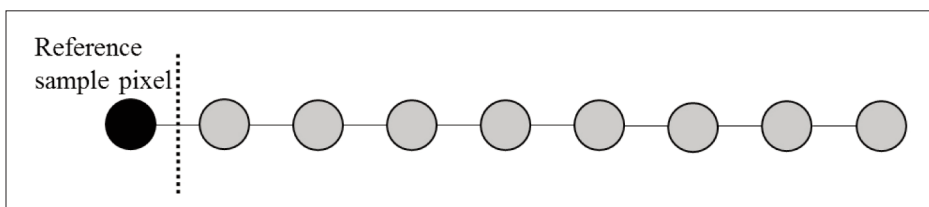
〈그림 1〉 VVC에서 사용된 변환 기술

쿼드트리 구조를 형성하면서 변환을 수행한다. 이러한 트리 구조를 이용하여 부호화하는 과정은 유사하나 사용되는 변환 커널의 종류가 증가한다는 점, 적용 변환 블록의 크기가 증가한다는 점과 이차 변환을 사용한다는 점에서 차이가 존재한다. <그림 1>은 VVC 비디오 코덱에서 달라진 변환 기술을 보여준다.

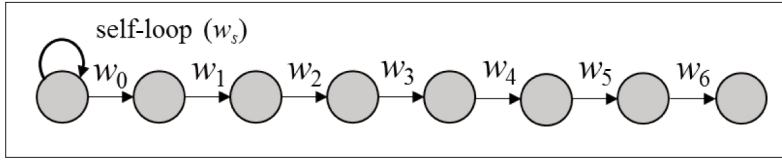
<그림 1>에서 보는 것처럼 변환 및 양자화 과정에서 여러 가지의 변환 커널을 사용하였을 뿐만 아니라, 이차 변환까지 수행하여 압축 효율을 증가시키는 것을 보여준다. 그리고 인터 예측 블록에 대해서는 블록의 일부만을 변환하는 SBT 변환도 채택되었다. HEVC에서는 인터 예측 부호화에서는 단일 종류의 DCT-2 변환 커널을 사용하고, 인트라 예측 잔차 신호 중 4×4 블록에 대해서는 단일 종류의 DST-7 커널을 사용하였다. 그러므로 어떤 변환 커널을 사용했는지에 대한 부가정보를 사용할 필요가 없었다. 그러나 VVC에서는 DCT-2, DST-7이외에 DCT-8 세 종류의 변환 커널이 사용되고, 어떤 변환 커널의 종류가 사용되는지를 시그널링하기 위하여 부가 정보 전송이 필요하다. 또한 주변환이 적용되고 그 변환 계수에 대하여 이차 변환을 수행하여 추가적인 압축 향상을 얻는 기술을 채택하였다. 다음 장에서는 주변환 커널의 종류와 적용방법, 이차 변환 방법과 SBT에 대하여 알아보기로 한다.

1. 주변환 기술(Primary transform)

주변환 기술은 일반적인 비디오 코덱과 같이 예측 잔차를 입력 신호로 받아 변환 커널을 적용하여 변환 계수를 얻는 것을 의미한다. VVC에서는 주변환과 이차변환, 2번의 변환을 수행하기 때문에 이차변환과 구분하기 위하여 주변환이라고 명명한다. 주변환 기술은 HEVC와 같은 일반적인 비디오 코덱에서 DCT-2를 이용하여 주로 예측 잔차 신호를 변환하여 신호에 대한 에너지 압축(energy compaction)을 얻는데 사용하였다. 여기서는 예측 잔차의 신호 성질이 픽셀 상에서 균일하다고 가정하며 DCT-2로 압축하였을 때 소수의 주파수만으로 신호 표현이 가능하면 보다 높은 압축 성능을 얻을 수 있다. 그러나 H.264/AVC 이후 예측 잔차를 얻는 과정이 보다 다양해지고, 복잡해짐에 따라 예측 잔차 신호의 특성이 균일한 특성을 갖지 못하는 경우가 많아지게 되었다. 특히 인트라 예측과 같은 경우는 주변 블록의 참조 픽셀로부터 방향성 예측을 하기 때문에, 잔차 신호가 에지 특성이 강한 특징을 갖고 있다. 이에 따라, 다양한 특성을 갖는 예측 잔차 신호에 적합한 다양한 변환 커널의 유도 또는 사용이 연구되었다. 참조 픽셀이 블록의 가장 자리에 위치할 때 방향성 인트라 예측을 통해 잔차 신호를 생성할 때, DST-7이 이러한 특성을 갖는 신호에 매우



<그림 2> 화면 내 예측의 예시



〈그림 3〉 화면 내 예측에 대한 그래프 모델링

적합한 것으로 알려져 있다[1]. 이는 그래프 이론을 통해 증명 가능하다. 〈그림 2〉는 좌측의 참조 픽셀로부터 화면 내 예측하는 신호의 예시를 보여준다.

〈그림 2〉에서 보는 것처럼, 그림의 점선은 블록의 경계를 의미하고 현재 블록의 좌측의 참조 픽셀 픽로부터 방향성 예측을 통해 잔차 신호를 생성하는 것을 보여준다. 〈그림 2〉의 그래프를 결합행렬, 차수 행렬로 표현할 수 있다. 블록 경계선 밖의 참조 샘플 픽셀은 그래프에는 포함되지 않으나 첫 번째 꼭지점에 셀프 루프(self-loop)로 정의하여 표현한다. 이것은 참조 픽셀이 첫 번째 픽셀에 가장 근접해 있기 때문에 자기 자신과 연결된 에지로 표현한 것과 같은 물리적 의미이다. 이를 종합하여 화면 내 예측을 그래프로 표현하면 〈그림 3〉과 같다[1].

〈그림 3〉에서 보는 것처럼 참조 픽셀은 self-loop 형태의 에지가 되고, 각 꼭지점은 가중치를 갖는 에지로 연결된 것을 볼 수 있다. 또한 제안 방법에서는 각 에지의 가중치를 1로 가정한다. 이 그래프의 차수행렬, 결합행렬은 식 (1), (2), (3)과 같이 표현할 수 있다[1].

이때 방향성을 갖는 self-loop로 인한 차수행렬의 차수가 1이 증가하게 되고 이것을 별도의 self-loop 행렬(S)로 표현할 수 있다. L, A, S를 이용하여 라플라시안 행렬을 구하면 식 (8)과 같이 얻을 수 있다.

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (2)$$

$$\mathbf{S} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

$$\mathbf{L} = \mathbf{D} - \mathbf{A} + \mathbf{S}$$

$$= \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \quad (4)$$

$$\mathbf{L} = \mathbf{D} - \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (5)$$

$$\text{diag}\{\mathbf{\Lambda}\} = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N \quad (6)$$

여기서 $\mathbf{U}$ 는 고유행렬, $\mathbf{\Lambda}$ 는 $\mathbf{L}$ 의 고유치($\lambda_1, \dots, \lambda_N$)이다. 라플라시안 행렬을 식 (5)와 같이 Eigen 분해할 경우에는 그래프에 해당하는 고유행렬과 고유치를 얻을 수 있고, 여기서 얻어진 고유행렬은 그래프가 신호라고 가정할 경우에 기저 벡터라고 볼 수 있고, 이 예시에서는 그 기저벡터가 DST-7 커널이 된다[1].

이 유도는 주변 블록의 경계 픽셀을 참조 픽셀로 하고 그 픽셀로부터 방향성 예측을 수행하여 잔차 신호를 생성하는 신호의 경우, DST-7이 가장 적합한 변환 커널임을 보여준다. 이러한 방법을 적용하여 DCT-8 커널도 유도할 수 있다. <그림 3>의 self-loop가 오른쪽에 존재할 때 동일한 방법으로 커널을 유도하면 DCT-8이 된다. DCT-8은 방향성 인트라가 예측되는 참조 픽셀이 블록의 오른쪽 가장자리에 존재할 때 가장 적합한 커널이 된다고 할 수 있다. DST-7과는 달리 실제 비디오 코덱에서 참조 픽셀이 현재 블록의 오른쪽에 존재하는 경우는 거의 존재하지 않기 때문에 인트라

예측 부호화 시 DCT-8의 압축 성능은 DST-7에 비해 성능 효과가 미미하다고 할 수 있다. 상기 설명하였듯이 DST-7과 DCT-8은 서로 쌍을 이루는 대칭적 구조를 가지고 있으며 식 (7)과 같은 관계를 가진다.

$$\mathbf{T}_{DST-VII} = \mathbf{S} \times \mathbf{T}_{DCT-VIII} \times \mathbf{R} \quad (7)$$

여기서 $\mathbf{T}_{DST-VII}$ 는 DST-7 커널, $\mathbf{T}_{DCT-VIII}$ 는 DCT-8 커널을 나타낸다. 또한 $\mathbf{S}$ 와 $\mathbf{R}$ 은 1과 0으로만 이루어진 순열 행렬이고 행렬의 원소는 식 (8), (9)와 같이 정의한다.

$$F_{m,n} = \begin{cases} 1, & \text{if } n = N - 1 - m, \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$S_{m,n} = \begin{cases} (-1)^m, & \text{if } n = m, \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

식 (7)의 유도과정에서 알 수 있듯이 DST-7과 DCT-8의 커널의 형태는 대칭적이며, 수식으로 비교하여도 그 형태가 유사하지만 대칭적인 특징을 가지고 있다. 위와 같은 방법으로 유도할 수 있는 DST-7, DCT-8과 DCT-2는 <표 1>과 같이 정의할 수 있다.

<표 1>에서 정의된 변환 커널은 블록 크기 $M \times N$ 에 대해서 분리 가능한 형태로 정의된다. 즉 가로 방향으로 변환 후 세로 방향으로 순차적으로 적용한다. 즉 <표 1>에서 N 은 가로 또는 세로 방향 적용 시, 적용되는 변의 픽셀 개수이다. 따라서 정의된 변환 커널은 N 에 따라 정수 형태로 정의되고, 변환과 양자화 시 비트 동적 영역과 정확도 등을 고려하여 8비트로 표현할 수 있도록 정의한다.

〈표 1〉 픽셀 개수가 N-point일 때 적용하는 변환 커널의 종류 [2][3]

변환 커널	Basis function $T_i(j), i, j = 0, 1, \dots, N-1$
DCT-2	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ $\text{where, } \omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-8	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-7	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

〈표 2〉에서 보는 것처럼 각 커널의 원소는 8 비트로 표현하도록 정수로 정의된다. VVC에 정의된 커널은 4-point DCT-2, 8-point DCT-2, 16-point DCT-2, 32-point DCT-2와 4-point DST-7은 HEVC와 동일한 커널을 사용한다. 그리고 새로 정의된 64-point DCT-2, 4-point DCT-8, 8-point, 16-point, 32-point DST-7과 DCT-8는 HEVC와 동일하게 8 비트 정수로 표현된다.

본격적인 VVC 표준화 논의가 진행되기 이전, JVET에서는 위 3개의 커널 이외에 보다 다양한 커널 적용하는 방법이 제안[5-10]되었다. 즉 DST-1, DST-3, DCT-4, DCT-5 등 미리 정의된 커널 뿐만 아니라, 잔차 신호를 오프라인에서 학습하여 유도한 KLT 커널 등을 적용하는 방법을 고려하였으

〈표 2〉 VVC 표준으로 정의한 8비트 정수 변환 커널: 4-point DST-7과 DCT-8[4]

(a) 4-point DST-7	(b) 4-point DCT-8
$\text{transMatrix}[m][n] =$ $\{$ $\{ 84 \ 74 \ 55 \ 29 \}$ $\{ 74 \ 0 \ -74 \ -74 \}$ $\{ 55 \ -74 \ -29 \ 84 \}$ $\{ 29 \ -74 \ 84 \ -55 \}$ $\}$	$\text{transMatrix}[m][n] =$ $\{$ $\{ 29 \ 55 \ 74 \ 84 \}$ $\{ 74 \ 74 \ 0 \ -74 \}$ $\{ 84 \ -29 \ -74 \ 55 \}$ $\{ 55 \ -84 \ 74 \ -29 \}$ $\}$

나, CE 논의 끝에 〈표 1〉에서 정의한 3개의 커널만 사용하기로 결정하였고, 이 기술은 다중 변환 커널 선택(MTS, Multiple Transform Selection)이라고 명명하였다. MTS 커널이 적용되는 변환 블록의 크기는 〈표 3〉과 같이 정의한다.

〈표 3〉 블록 크기와 예측 타입에 따른 MTS 적용

Point - N	Luma		Chroma
	Intra	Inter	
$N \geq 128$	No MTS	No MTS	No MTS
$N \geq 64$	DCT-2	DCT-2	No MTS
$N \leq 32 \ \& \ N \geq 4$	DCT-2, DST-7, DCT-8	DCT-2, DST-7, DCT-8	DCT-2

〈표 3〉에서 보는 것처럼 MTS 적용 블록의 최대 가로 또는 세로 픽셀 크기는 64이다. 분리 가능한 변환을 수행하기 때문에 MTS가 적용되는 변환 블록의 크기는 〈표 4〉와 같다.

〈표 4〉 분리 가능한 MTS가 적용되는 변환 블록의 크기

4×4	8×4	16×4	32×4	64×4
4×8	8×8	16×8	32×8	64×8
4×16	8×16	16×16	32×16	64×16
4×32	8×32	16×32	32×32	64×32
4×64	8×64	16×64	32×64	64×64

〈표 3〉에서 보는 것처럼 총 3개의 주변환 커널을 사용하기 때문에 MTS 커널의 인덱스를 시그널링해 주어야 한다. 이 인덱스를 `tu_mts_idx`라고 하고 〈표 5〉와 같이 정의한다.

〈표 5〉 MTS 커널 인덱스 정의

<code>tu_mts_idx[x0][y0]</code>	0	1	2	3	4
<code>trTypeHor</code>	0	1	2	1	2
<code>trTypeVer</code>	0	1	1	2	2

`tu_mts_idx`는 MTS 커널 인덱스로 나타나고, 각 인덱스 값에 따라 가로 및 세로 방향의 커널을 정의한다. 〈표 5〉에서 `trTypeHor`, `trTypeVer`는 가로 및 세로 방향의 커널의 종류를 나타낸다. 0은 DCT-2, 1은 DST-7, 2는 DCT-8을 의미한다. 〈표 5〉는 명시적으로 MTS 인덱스를 시그널링하기 때문에 이것을 ExplicitMTS라고 한다. ExplicitMTS는 SPS에서 `mts_flag`가 1일 때만 시그널링한다. 반면, `tu_mts_idx`는 명시적으로 나타내지 않고 인코더 또는 디코더에서 암시적(implicit)으로 유도할

때 이것을 ImplicitMTS라고 하고 SPS에서 `mts_flag`가 0일 때 수행된다. ImplicitMTS는 `mts_flag`이고 ISP, MIP 등 인트라 예측 관련 기술의 조건에 따라서 수행된다. ImplicitMTS 사용 시 〈표 5〉의 `trTypeHor`, `trTypeVer`는 식 (10)과 같이 유도할 수 있다[11–14].

$$\begin{aligned} & \text{trTypeHor} \\ &= (\text{nTbW} \geq 4 \ \&\& \ \text{nTbW} \leq 16) ? \text{DST7} : \text{DCT2} \\ & \text{trTypeVer} \\ &= (\text{nTbH} \geq 4 \ \&\& \ \text{nTbH} \leq 16) ? \text{DST7} : \text{DCT2} \end{aligned} \quad (10)$$

적용하는 가로 또는 세로 방향의 블록의 크기가 4 이상, 16 이하인 경우에는 DST-7을 사용하고 그렇지 않은 경우 DCT-2를 사용한다. 즉, 블록의 크기가 작은 경우에는 예측 잔차 신호의 방향성이 크기 때문에 DST-7을 사용하고, 상대적으로 큰 블록인 경우 좀더 균일한 특성을 갖는 잔차 신호의 특성이 강하므로 DCT-2 커널을 사용할 수 있도록 하였다.

HEVC에서는 4-, 8-, 16-, 32-point DCT-2와 4-point DST-7을 표준 커널로 사용하였고, 복잡도 감소를 위해 변환 과정 시 부분적 버터플라이 구조를 사용하였다. 버터플라이 구조를 사용하는 것을 곱셈과 덧셈의 회수 등 절대적인 연산 회수 복잡도는 크게 감소시킬 수 있으나, 변환 연산을 여러 개의 연산 블록으로 분해하여 수행하는 과정(cascading multiple arithmetic)이 필요하므로, 분해 연산을 수행할 경우 각 분해 과정에서 연산 지연이 발생하는 것으로 보고되고 있고[15], VVC에서는 단순히 행렬 곱셈 연산만을 적용하는 것으로 결정되었다. 이것은 변환 과정의 지연을 방지하기 때문에 병렬 연산 수행 시 보다 유리한 것으로 알려져 있다. CE에서 DST-7, DCT-8의 복잡도 감소를 위한 연

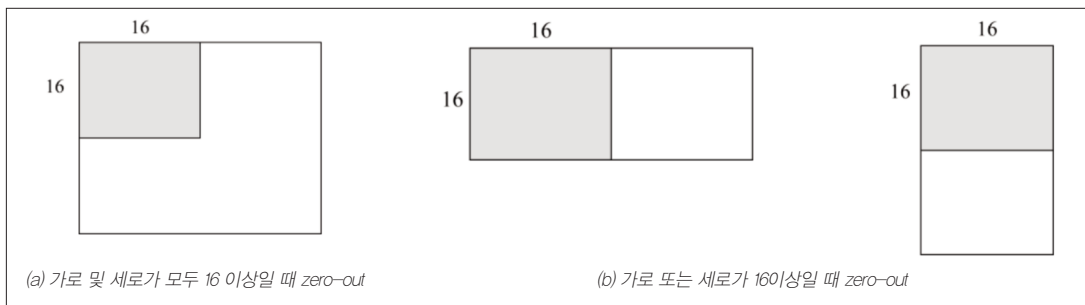
{	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	}
{	c	f	i	l	o	o	l	i	f	c	0	-c	-f	-i	-l	-o	}
{	e	j	o	m	h	c	-b	-g	-l	-p	-k	-f	-a	d	i	n	}
{	g	n	l	e	-b	-i	-p	-j	-c	d	k	o	h	a	-f	-m	}
{	i	o	f	-c	-l	-l	-c	f	o	i	0	-i	-o	-f	c	l	}
{	k	k	0	-k	-k	0	k	k	0	-k	-k	0	k	k	0	-k	}
{	m	g	-f	-n	-a	l	h	-e	-o	-b	k	i	-d	-p	-c	j	}
{	o	c	-l	-f	i	i	-f	-l	c	o	0	-o	-c	l	f	-i	}
{	p	-a	-o	b	n	-c	-m	d	l	-e	-k	f	j	-g	-i	h	}
{	n	-e	-i	j	d	-o	a	m	-f	-h	k	c	-p	b	l	-g	}
{	l	-i	-c	o	-f	-f	o	-c	-i	l	0	-l	i	c	-o	f	}
{	j	-m	c	g	-p	f	d	-n	i	a	-k	l	-b	-h	o	-e	}
{	h	-p	i	-a	-g	o	-j	b	f	-n	k	-c	-e	m	-l	d	}
{	f	-l	o	-i	c	c	-i	o	-l	f	0	-f	l	-o	i	-c	}
{	d	-h	l	-p	m	-i	e	-a	-c	g	-k	o	-n	j	-f	b	}
{	b	-d	f	-h	j	-l	n	-p	o	-m	k	-i	g	-e	c	-a	}

〈그림 4〉 DST-7 정수 커널의 정의

산 방법 등이 논의되었으나, 행렬 연산을 수행하는 것으로 결정되었다. 단 DST-7과 DCT-8 연산 수행 시, 커널의 특성 상 반복적으로 사용되는 커널 원소와 대칭성을 이용하여 행렬 연산 수행 시 복잡도를 감소시키는 방법[15]이 제안되어 채택되었다.

〈그림 4〉는 DST-7 정수 변환 커널의 정의식이다. 첫 번째 행 원소 a, b, c, ..., p 만을 정의하면 DST-7 커널을 생성할 수 있다. 이 때 DST-7 커널의 특성 상 $a+j=l$, $b+i=m$, $c+h=n$, $d+g=o$, $e+f=p$ 의 관계를 가진다. 이 성질을 이용한다면,

첫 번째 출력 원소를 출력할 때 $y_0 = a \cdot (x_0+x_{11}) + b \cdot (x_1+x_{12}) + \dots + j \cdot (x_9+x_{11}) + k \cdot x_{10}$ 을 이용하여 총 10번의 곱셈만으로 결과값을 얻을 수 있다. 단순 행렬 연산 16번에 비해 곱셈의 개수를 감소시켜 복잡도를 감소시키는 방법이 채택되었다. 또 다른 MTS 복잡도 개선 방법으로 저 주파수영역의 특정 계수만을 제외한 고주파 계수를 0으로 만드는 방법[17]이 채택되었다. DST-7, DCT-8의 경우 가로 또는 세로 방향의 변환 계수 중에서 16보다 큰 인덱스를 가지는 계수는 모두 0으로(zero-out) 만



〈그림 5〉 변환 블록에서 가로 및 세로 길이에 따른 zero-out 예시

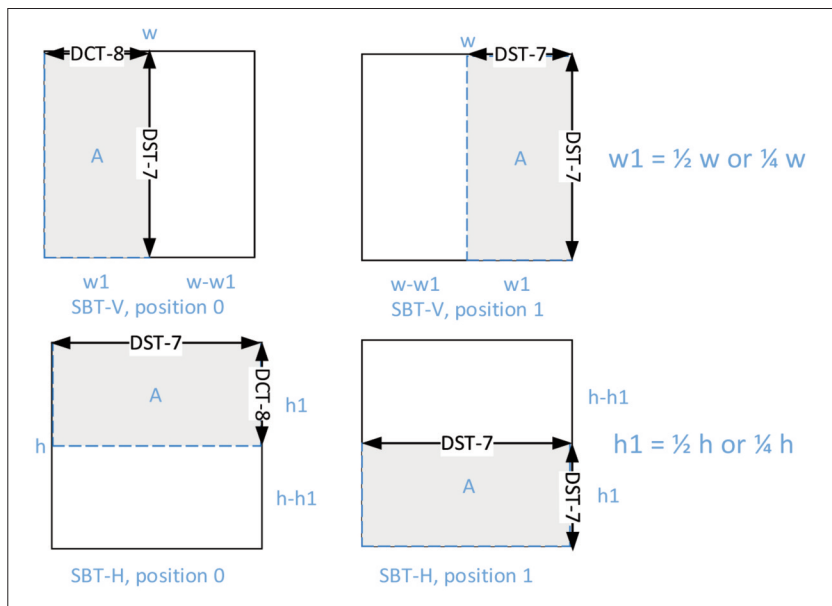
든다. <그림 5>는 이와 같은 zero-out의 예시를 보여준다. <그림 5>에서 진한색 이외의 부분은 변환 계수를 0으로 zero-out한다.

고주파수 영역의 계수를 zero-out하는 복잡도 감소 방법의 채택으로, DST-7과 DCT-8에서 32-point 커널은 주파수 영역 16~31까지 커널 원소는 저장할 필요가 없어 커널 저장에 따른 메모리를 감소시킬 수 있다.

2. 서브 블록 변환(SBT, Sub-Block Transform)

VVC에서는 인터 예측된 블록에 대해서 서브 블록 변환 기술(SBT)[18]을 채택하였다. SBT에는 잔차 예측 블록에 가로 또는 세로로 분할된 두 블록 중 하나만을 부호화하는 변환 방식을 말하고 인터 예측 블록에 대해서만 적용한다. 현재 CU가 인터 예측 부호화하는 블록이고, cu_cbf 가 1, 즉 양자화

계수의 개수가 1개 이상일 때, SBT로 부호화되는지 cb_sbt_flag 를 이용하여 시그널링한다. 또한 SBT가 사용한다면, SBT의 종류와 SBT가 적용되는 위치를 추가적으로 시그널링해야 한다. <그림 6>은 SBT의 분할 타입과 분할 타입에 따른 변환 커널의 종류를 나타낸다. <표 6>에서는 <그림 6>의 플래그 명과 변환 타입을 표로서 나타낸 것이다. <그림 6>과 <표 6>에서 보는 것처럼, 블록이 세로 방향으로 분할하면 $cb_sbt_horizontal_flag=0$, 가로 방향이면 $cb_sbt_horizontal_flag=1$ 로 시그널링된다. 또한 분할 블록에서 좌측 또는 위측에 위치한 블록이 부호화되는 경우 $cu_sbt_pos_flag=0$, 우측 또는 아래측에 위치한 블록이 부호화하는 경우는 $cu_sbt_pos_flag=1$ 로 부호화한다. $cb_sbt_horizontal_flag$ 와 $cu_sbt_pos_flag$ 에 따라서 가로 및 세로 방향의 변환 커널이 <표 6>과 같이 정의되어 변환을 수행한다. 가로 및 세로 방향으로 분할될 때 분할 비율은 CU의



<그림 6> SBT에서 SBT의 위치와 변환 종류의 시그널링

〈표 6〉 SBT의 플래그와 변환 타입

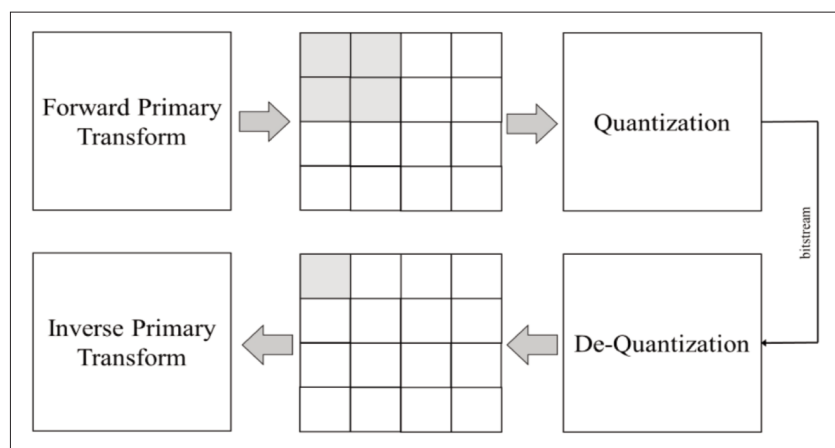
cu_sbt_horizontal_flag	cu_sbt_pos_flag	trTypeHor	trTypeVer
0	0	DCT-8	DST-7
0	1	DST-7	DST-7
1	0	DST-7	DCT-8
1	1	DST-7	DST-7

가로 또는 세로 길이의 절반 크기로 분할되거나 비대칭 이진 트리의 경우 (Asy-metric Binary Tree, ABT) 1:3/3:1비율로 분할될 수 있다. SBT에서 변환 블록의 한쪽 크기가 32 이상이면, 가로 및 세로 방향에 대해 DCT-2 커널을 적용한다. SBT에 적용되는 변환 커널의 최대 크기(max-SbtSize)는 SPS에서 시그널링되며 HD, 4K영상에 대해서는 64, 해상도가 작은 영상에 대해서는 32로 설정된다. 또한 SBT는 인터 및 인트라 혼합 예측 모드(CIIP)[19]와 TPM 모드[20]에 대해서는 적용하지 않는다.

3. 이차변환 기술(Secondary transform)

이미 이차 변환[21]은 HEVC 표준화 과정에서 표

준화 기술로서 논의된 바 있다. [22]에서는 1차 변환 커널에 대하여 ROT(Rotational Transform)라고 명명한 이차 변환 커널을 적용하여 압축 성능을 위한 기술이 제안되었으나, HEVC의 표준 기술로 채택되지 않았다. ROT 이차 변환 기술은 1차 변환 이후 생성된 가로와 세로의 성분에 대하여 회전 변환 행렬을 적용하였으며, 총 6개의 각도를 이용하여 여러 가지 이차 변환 조합을 구성하였다. 그러나 변환 커널에 대한 이론적 배경이 설명되지 않았으며 그 복잡도에 비해 압축 성능이 미미한 단점이 있다. [23]에서는 분리 가능하지 않은 이차 변환 커널을 제안하였고 높은 성능에도 불구하고 그 복잡도가 매우 크게 증가하는 단점이 존재하였다.



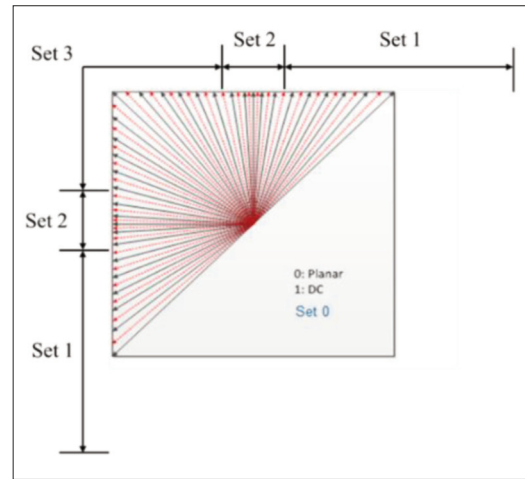
〈그림 7〉 이차 변환(LFNST) 블록도

〈그림 7〉은 VVC의 이차 변환의 개념도를 보여준다. 〈그림 7〉에서 보는 것처럼 먼저 주변환 커널인 DCT-2, DST-7, DCT-8을 이용하여 주변환을 수행한다. 이렇게 생성한 주변환계수에 대하여 좌측 상단의 저주파 영역의 계수에 대해서만 이차변환을 수행한다. 이차변환이 적용되는 저주파 영역의 크기는 주변환 블록의 크기에 따라 달라지며, 또한 그 영역에 적용하는 이차변환 커널의 종류 역시 달라지게 된다.

〈그림 7〉과 같이 주변환 계수의 저주파 영역의 계수에 대해서만 이차변환을 수행하고, 주변환과 달리 가로 방향과 세로 방향의 분리 가능한 커널이 아닌 비분리 커널을 사용하므로 VVC에서 이차변환을 저주파비분리변환(LFNST, Low Frequency Non-Separable Transform, NFNST)이라고 명명하였다. LFNST는 인트라 예측 부호에 대해서만 변환을 수행하고, 적용 커널은 인트라 예측 모드에 따라 다르게 구성하여 정의한다. 〈표 7〉과 〈그림 8〉은 인트라 예측 모드에 따른 LFNST 커널의 적용을 보여준다.

〈표 7〉과 〈그림 8〉에서 알 수 있듯이, 인트라 예측 모드에 따라서 총 4종류의 LFNST 커널 세트가 매핑된다. $\text{predModeIntra} < 0$, $56 \leq \text{predModeIntra} \leq 80$

은 광각 인트라 예측 모드(WAIP, Wide Angle Intra Prediction)이고 LFNST 커널 세트 1을 사용한다. $80 \leq \text{predModeIntra} \leq 83$ 은 INTRA_LT_CCLM, INTRA_T_CCLM or INTRA_L_CCLM 모드로서 LFNST 커널 세트 0을 사용한다[24].



〈그림 8〉 인트라 예측 모드에 따른 LFNST 커널 인덱스 세트 매핑

각 커널 세트(lfnstTrSetIdx)에 대해 총 2종류의 커널이 정의되어 있고, 2종류의 커널 중에서 어떤 커널을 선택할지는 lfnst_idx 인덱스를 이용하여 시

〈표 7〉 인트라 예측 모드에 따른 LFNST 커널 세트 매핑 테이블

predModeIntra	LFNST커널 세트 인덱스 (lfnstTrSetIdx)
predModeIntra < 0	1
0 ≤ predModeIntra ≤ 1	0
2 ≤ predModeIntra ≤ 12	1
13 ≤ predModeIntra ≤ 23	2
24 ≤ predModeIntra ≤ 44	3
45 ≤ predModeIntra ≤ 55	2
56 ≤ predModeIntra ≤ 80	1
81 ≤ predModeIntra ≤ 83	0

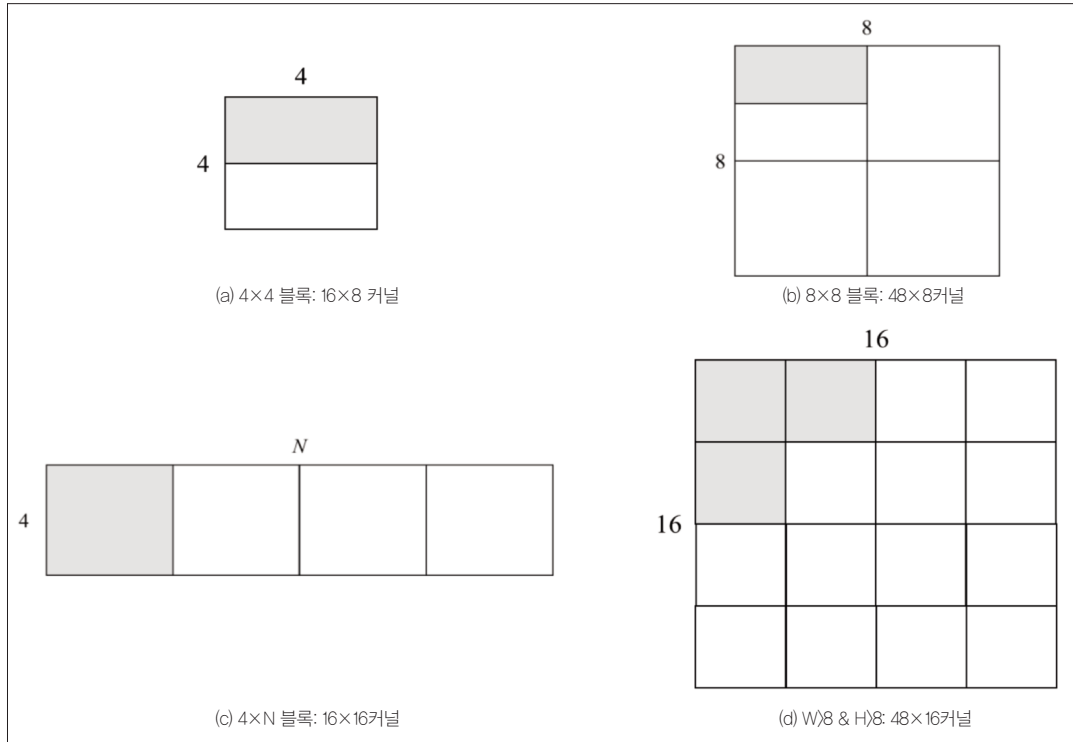
그넨링한다. $lfnst_idx$ 가 0일 때는 LFNST가 수행되지 않음을 의미하고, $lfnst_idx$ 가 1 또는 2에 따라 서로 다른 LFNST 커널이 적용된다. 블록의 크기에 따라서 이와 같은 동일한 세트가 하나 더 존재하므로 총 $4 \times 2 \times 2 = 16$ 종류의 LFNST 커널이 존재한다. LFNST는 커널의 크기가 16×16 과 16×48 크기의 정의되어 있고 블록 크기에 따라 <표 8>과 같이 적용된다.

<표 8> 블록의 크기와 LFNST 커널의 크기

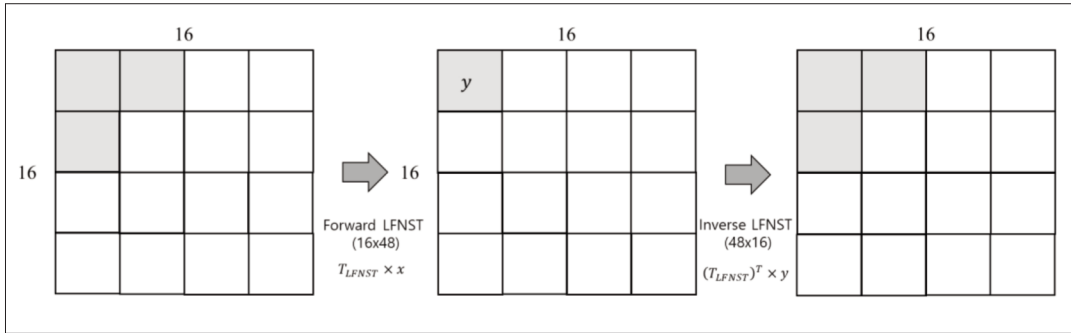
블록의 크기	커널의 크기
4×4	16×8
$4 \times N, N \times 4 (N \geq 8)$	16×16
8×8	48×8
$W \geq 8 \ \& \ H \geq 8$	48×16

<그림 9>는 블록 크기와 유형에 따른 서로 다른 커널의 적용을 나타낸다. 그림에서 그레이 색으로 표시한 부분은 역 LFNST 변환 커널 적용 후의 변환 계수가 존재하는 부분이고 흰 색으로 표시된 부분은 모든 변환 계수가 0이 되는 부분이다.

<그림 10>은 16×16 변환 블록에 대하여 LFNST 적용하는 예시를 보여준다. <표 8>에 의하여 16×16 변환 블록에 대해서는 16×48 LFNST 커널이 <그림 10> 좌측 상단의 그레이색 변환 계수에 대하여 적용된다. 16×48 LFNST 커널이 적용되면 출력 16개의 변환 개수 y 를 얻을 수 있다. 이 계수는 양자화 및 엔트로피 과정을 거치게 되고, 디코더에서는 48×16 , 즉 역 LFNST 변환 커널을 이용하여 다시 48개의 변환 개수를 얻는다. 이것을 역 주변환을



<그림 9> 변환 블록 크기에 따른 커널의 적용 및 결과



〈그림 10〉 16×16 변환 블록에 대하여 LFNST 적용하는 예시

수행하여 픽셀영역의 잔차 신호를 구할 수 있다.

LFNST에서 4×4와 8×8 블록 크기에 적용된 커널은 각 16×8, 48×16 커널에서 고주파 부분을 제외한 저주파 부분의 절반 커널을 나타낸다[25]. 주 변환에서는 DCT-2, DST-7, DCT-8 등 12개의 DCT와 DST 커널 시리즈 중에서 3개를 채택하여 주변환 커널로 사용하였으나, 이차변환에서는 각 인트라 예측 모드 별로 오프라인에서 학습하여 유도한다. 이차변환도 주변환과 마찬가지로 8비트 정수 원소로 표현된다.

4. 변환 기술(Secondary transform)과 다른 툴 간의 관계

이차변환기술은 변환계수를 다시 한 번 변환하여 압축 성능을 향상시킨다는 측면에서 기존의 비디오 압축 기술에서는 사용하지 않았던, VVC에서 가장 변화가 큰 기술 중의 하나이다. 일반적으로 픽셀영역의 신호를 변환을 거쳐 에너지 압축을 하기 때문에 이 압축된 변환계수를 다시 변환하였을 때 압축 효과를 얻는 것이 기존의 압축 이론과는 다소 상이하기 때문이다. 이것은 변환 계수를 오프 라인에서

학습하여 변환 계수에 최적화된 변환 커널을 유도하였기 때문에 고성능의 압축 효율을 달성할 수가 있었다. 그러나 VVC에서는 특정한 조건을 만족하지 않는 입력 신호가 입력되었을 때는 만족할만한 압축 효율을 얻을 수 없다는 것을 많은 실험을 통하여 알 수 있었다. 따라서 LFNST와 같이 사용하였을 때 압축 효율이 높지 않은 툴에 대해서는 LSNFT를 적용하지 않도록 하는 기술들이 많이 제안되었다.

먼저 ExplicitMTS에서 DCT-2 이외의 커널로부터 생성된 변환 계수에 대해 LFNST를 적용하였을 때 압축 성능이 높지 않으므로 DCT-2 커널 쌍으로부터 얻은 변환 계수에 대해서만 LFNST를 적용할 수 있도록 결정하였다[26–27]. 그리고 ImplicitMTS의 경우, LFNST가 사용될 경우 DCT-2 이외의 다른 커널이 사용될 경우 압축 성능이 저하되는 것을 실험을 통해 확인하였다. 따라서 ImplicitMTS의 경우에는 DCT-2 변환 커널 쌍만을 사용할 수 있도록 결정하였다[28–29].

MIP(Matrix-based Intra Prediction) 인트라 예측[30]의 경우에도 CU에서 시그널링되는 mip_flag가 1일 때는, 주변환 커널을 DCT-2 변환 쌍만을

사용한다[28, 32].

III. 마무리

VVC는 현재 HEVC 대비 약 2배의 압축 성능을 달성하기 위하여, 표준화 작업이 한창 진행 중에 있으며, 2020년 10월 FDIS 공개를 목표로 하고 있다. 2013년 HEVC의 표준화 제정 이후, 미디어 환경은 급격히 변화되고 있고, 4K를 넘어 8K 초고해상도 실감 영상과, 가상현실 서비스를 받고자 하는 사용자의 욕구가 크게 증가하고 있다. 이에 발 맞추어 VVC의 표준화 제정은 초고해상도에 대한 고압축 성능을 얻고자 하는 목표에 맞게 시기 적절하다고 할 수 있다. 이러한 목표를 달성하기 위하여, VVC에서는 기존에 등장하지 않았거나, 소개가 되었고 하더라도 큰 복잡도 때문에 채택되지 않았던 기술들을 많이 적용하고 있다. 이러한 기술들은 특히

4K 이상의 초고해상도 영상의 압축 성능 향상에 크게 기여하고 있어, VVC 비디오 코덱의 필수 기술들이라고 할 수 있다.

주변환기술에서 확장된 DCT-2 변환 커널과, DST-7의 확대 적용과, DCT-8 커널의 새로운 도입은 기존의 코덱과 비교하여 크게 변경된 사항이라고 할 수 있다. 또한 VVC에서 새롭게 채택된 이차변환인 LFNST는 작은 복잡도 증가만으로 압축 효율을 크게 상승시키는 핵심 톨로 채택되었다. SBT나 ISP 등 인터 또는 인트라 예측 기술과 결합한 변환 기술도 채택되어 비디오 압축 효율 향상시키고 있다.

그러나 VVC는 HEVC에 비해 인코더 복잡도가 10배 이상 매우 크게 증가하여 이를 상용화하기에는 아직도 가야할 길이 멀다. 4K 이상 비디오에 대해 초당 30에서 60프레임 이상을 실시간으로 부/복호화하기 위해서는 압축 성능 저하 없이 복잡도 감소 및 고속 알고리즘 기술에 대한 연구도 요구되는 실정이다.

참고 문헌

- [1] 이범식, "비디오 코덱을 위한 2차 변환기 설계", 차세대컴퓨팅학회 논문지, vol.13, no. 5, pp.113-123, 2017년 12월
- [2] J. Chen, Y. Ye and S. Kim, "Algorithm description for Versatile Video Coding and Test Model 6 (VTM 6)", JVET-O2002, 15-th JVET meeting, Gothenborg, Jul. 2019.
- [3] W. Park, B. Lee and M. Kim, "Fast computation of integer DCT-V, DCT-VIII and DST-VII for video coding" IEEE Transactions on Image Processing, vol. 28, no. 12, pp. 5839-5851, Dec. 2019.
- [4] B. Bross, J. Chen and S. Liu "Versatile Video Coding (Draft 6)", JVET-O2001, 15-th JVET meeting, Gothenborg, Jul. 2019.
- [5] X. Zhao, X. Li and S. Liu, CE6: "CE6: Compound Orthonormal Transform (Test 6.1.1 a/b/c/d)", JVET-M0496, 13-th JVET meeting, Marrakech, Jan. 2019
- [6] K. Abe, T. Toma, M. Ikeda, T. Tsukuba, K. Naser, F. L. Leanne and E. Francois, "CE6: JVET-L0262: Replacing all DST-7 / DCT-8 by DST-4 / DCT-4 used in MTS (test 6.1.1e)", JVET-M0084, 13-th JVET meeting, Marrakech, Jan. 2019
- [7] Y. Lin, J. Zheng, Q. Yu, N. Zhang and C. Zhu, "CE6 : MTS using DST-4 and transposed DCT-2 (test 6-1.3)", JVET-M0244, 13-th JVET meeting, Marrakech, Jan. 2019
- [8] A. Said, H.E. Egilmez, Y.-H. Chao, V. Seregin and M. Karczewicz, "CE6: Efficient Implementations of MTS with Transform Adjustments (tests 1.4a-d)", JVET-M0538, 13-th JVET meeting, Marrakech, Jan. 2019

- [9] H. Egilmez, V. Seregin, A. Said, T. Hsieh and M. Karczewicz, "CE6: Replacement of 4-point DST7/DCT8 with DST4/DCT4 in MTS (Test 6.1.6)", JVET-M0521, 13-th JVET meeting, Marrakech, Jan. 2019
- [10] M. Koo, M. Salehifar, J. Lim, S. Kim, "CE6: Fast DST-7/DCT-8 based on DFT (test 6.2.1)", JVET-M0288, 13-th JVET meeting, Marrakech, Jan. 2019
- [11] H. Gao, S. Esenlik, B. Wang, A. M. Kotra and J. Chen, "Non-CE6: Unification of Implicit Transform Core Selection", JVET-N0172, 14-th JVET meeting, Geneva, Mar. 2019
- [12] S.-C. Lim, J. Kang, J. Lee, H. Lee, H. Y. Kim, "Non-CE6: Simplification on implicit transform selection in ISP mode", JVET-N0375, 14-th JVET meeting, Geneva, Mar. 2019
- [13] H. E. Egilmez, A. Said, N. Hu, C.-H. Hung, V. Seregin, M. Karczewicz, "on-CE6: A Simplification of Implicit MTS", JVET-N0419, 14-th JVET meeting, Geneva, Mar. 2019
- [14] H. E. Egilmez, A. K. Ramasubramanian, A. Said, N. Hu, V. Seregin, G. Van der Auwera, M. Karczewicz, "Non-CE6: A Simplification of Implicit Transform Selection in Intra-subblock Partitioning", JVET-N0420, 14-th JVET meeting, Geneva, Mar. 2019
- [15] G. J. Sullivan and J.-R. Ohm, "Meeting Report of the 13th JVET Meeting", JVET-M1000, 13-th JVET meeting, Marrakech, Jan. 2019
- [16] X. Zhao, X. Li, Y. Luo, S. Liu, "CE6: Fast DST-7/DCT-8 with dual implementation support (Test 6.2.3)", JVET-M0497, 13-th JVET meeting, Marrakech, Jan. 2019
- [17] M. Koo, M. Salehifar, J. Lim, S. Kim, "CE6-related: 32 point MTS based on skipping high frequency coefficients", JVET-M0297, 13-th JVET meeting, Marrakech, Jan. 2019 \
- [18] Y. Zhao, H. Gao, H. Yang, J. Chen, "CE6: Sub-block transform for inter blocks (Test 6.4.1)", JVET-M0140, , 13-th JVET meeting, Marrakech, Jan. 2019
- [19] M.-S. Chiang, C.-W. Hsu, Y.-W. Huang, S.-M. Lei, "CE10.1.1: Multi-hypothesis prediction for improving AMVP mode, skip or merge mode, and intra mode", JVET-L0100, 12-th JVET meeting, Macal, Oct. 2018
- [20] N. Zhang, H. Liu, L. Zhang, K. Zhang, Y. Wang, "Non-CE4: Simplified motion field storage for TPM", JVET-O0265, 15-th JVET meeting, Gothenburg, Jul. 2019
- [21] M. Koo, M. Salehifar, J. Lim, S. Kim, "CE6: Reduced Secondary Transform (RST) (CE6-3.1)", JVET-N0193, 4-th JVET meeting, Geneva, Mar. 2019
- [22] K. McCann, W.-J. Han, I.-K. Kim, J.-H. Min, E. Alshina, T. Lee, J. Chen, V. Seregin, S. Lee, Y.-M. Hong, M.-S. Cheon, N. Shlyakhov, "Samsung's response to the call for proposals on video compression technology," JCTVC-A124, April, 2010.
- [23] X. Zhao, J. Chen, M. Karczewicz, "Mode-dependent non-separable secondary transform", ITU-T SG16/Q6 Doc. COM16 C1044, Oct. 2015.
- [24] Z. Zhang, R. Sj berg, R. Yu R. Sj berg, R. Yu, "Non-CE6: On LFNST transform set selection for a CCLM coded block", JVET-O0219, 15-th JVET meeting, Gothenburg, Jul. 2019
- [25] M. Siekmann, H. Schwarz, D. Marpe, T. Wiegand, "CE6-2.1: Simplification of Low Frequency Non-Separable Transform", JVET-O0094, 15-th JVET meeting, Gothenburg, Jul. 2019
- [26] H. E. Egilmez, A. K. Ramasubramanian, A. Said, V. Seregin, M. Karczewicz, "Non-CE6: An MTS-based Restriction for LFNST beyond Transform Skip", JVET-O0368, 15-th JVET meeting, Gothenburg, Jul. 2019
- [27] F. Le L annec, F. Urban, K. Naser, T. Poirier, "Non-CE6 : interaction between MTS and LFNST", JVET-O0466, 15-th JVET meeting, Gothenburg, Jul. 2019
- [28] K. Naser, F. Le L annec, T. Poirier, "Non-CE6 Interaction between Implicit MTS and LFNST and MIP", JVET-O0529, 15-th JVET meeting, Gothenburg, Jul. 2019
- [29] T. Ikai, "Non-CE6: Harmonization of implicit MTS and LFNST", JVET-O0214, 15-th JVET meeting, Gothenburg, Jul. 2019
- [30] JVET-N0217, 14-th JVET meeting, Geneva, Mar. 2019
- [31] X. Zhao, X. Li, S. Liu, "Non-CE6: Harmonization of LFNST, MIP and implicit MTS", JVET-O0540, 15-th JVET meeting, Gothenburg, Jul. 2019

필자소개



이범식

- 2000년 : 고려대학교 전기전자전파공학부 학사
- 2012년 : KAIST 정보통신공학과 박사
- 2012년 ~ 2013년 : UCSD, Postdoctoral Scholar
- 2014년 : KAIST 연구교수
- 2015년 ~ 2016년 : LG전자 차세대표준연구소 책임연구원
- 2016년 ~ 현재 : 조선대학교 정보통신공학과 조교수
- 주관심분야 : 비디오신호처리, 비디오압축, 바이오영상처리, 멀티미디어보안