일반논문 (Regular Paper)
방송공학회논문지 제28권 제1호, 2023년 1월 (JBE Vol.28, No.1, January 2023)
https://doi.org/10.5909/JBE.2023.28.1.100
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

곱셈 연산을 고려한 고속 역변환 방법

송 현 주[a)], 이 영 렬[a)‡]

Fast Inverse Transform Considering Multiplications

Hyeonju Song[a)] and Yung-Lyul Lee[a)‡]

요 약

하이브리드 블록 기반 비디오 압축에서 변환 부호화는 공간 영역의 잔차 신호를 주파수 영역으로 변환하여 낮은 주파수 대역에 에너지를 집중시켜 이후 엔트로피 코딩 과정에서 높은 압축률을 달성할 수 있게 한다. 최신 비디오 압축 표준인 VVC(Versatile Video Coding)는 DCT-2(Discrete Cosine Transform type 2), DST-7(Discrete Sine Transform type 7), DCT-8(Discrete Cosine Transform type 8)를 사용하여 주변환을 수행한다. 본 논문에서는 DCT-2, DST-7, DCT-8이 모두 선형 변환임을 고려하여, 선형 변환의 선형성을 이용하여 역변환 시 곱셈 연산량을 줄이는 역변환 방법을 제안한다. 제안하는 역변환 방법은 VVC의 참조 소프트웨어인 VVC Test Model-8.2 (VTM-8.2) 대비 비트율의 증가 없이 부호화 시간과 복호화 시간이 AI(All Intra)에서 평균 26%, 15%, RA(Randon Access)에서 평균 4%, 10% 감소하였다.

Abstract

In hybrid block-based video coding, transform coding converts spatial domain residual signals into frequency domain data and concentrates energy in a low frequency band to achieve a high compression efficiency in entropy coding. The state-of-the-art video coding standard, VVC(Versatile Video Coding), uses DCT-2(Discrete Cosine Transform type 2), DST-7(Discrete Sine Transform type 7), and DCT-8(Discrete Cosine Transform type 8) for primary transform. In this paper, considering that DCT-2, DST-7, and DCT-8 are all linear transformations, we propose an inverse transform that reduces the number of multiplications in the inverse transform by using the linearity of the linear transform. The proposed inverse transform method reduced encoding time and decoding time by an average 26%, 15% in AI and 4%, 10% in RA without the increase of bitrate compared to VTM-8.2.

Keyword : Linear transform, Computation complexity, Versatile Video Coding, fast inverse transform

a) 세종대학교 컴퓨터공학과(Department of Computer Engineering, Sejong University)
‡ Corresponding Author : 이영렬(Yung-Lyul Lee)
E-mail: yllee@sejong.ac.kr
Tel: +82-2-3408-3753
ORCID: https://orcid.org/0000-0003-2709-8282
※ 본 논문은 (과제명: 5G 연계 초고실감 미디어를 위한 VVC (Vesatile Video Codec) 디코더 IP 개발, 과제고유번호: 1415172976, 과제번호: 20010342) 산업통상자원부에서 지원한 과제로 일부 수행되었다.
· Manuscript October 26, 2022; Revised December 9, 2022; Accepted December 9, 2022.

Ⅰ. 서 론

일반적인 하이브리드 블록 기반 비디오 압축은 입력된 2차원 영상 신호를 여러 개의 블록으로 나누어 각 블록에 화면 내 예측 또는 화면 간 예측, 변환 및 양자화, 엔트로피 코딩 등의 과정을 적용하여 압축을 진행한다. 이중 변환 부호화는 예측 단계로부터 생성되는 잔차 신호에 적용되며, 공간 영역의 잔차 신호를 주파수 영역으로 변환함으로써 저주파 영역으로 에너지를 압축하여 이후 엔트로피 코딩 과정에서 더 효율적으로 압축할 수 있게 한다.

최근 4K 영상, UHD(Ultra High Definition) 영상과 같은 고화질, 고해상도 영상의 수요와 공급이 증가하고, 화상 회의, 비디오 스트리밍과 같은 서비스 시장의 성장으로 더 높은 압축 효율을 가지는 비디오 압축 기술의 필요성이 대두되고 있다. 최신 비디오 코딩 표준인 VVC(Versatile Video Coding)[1]는 ITU-T VCEG(Video Coding Experts Group)과 ISO/IEC MPEG(Moving Picture Experts Group)이 공동으로 창설한 JVET(Joint Video Exploration Team)에서 이전 표준인 HEVC(High Efficiency Video Coding)[2] 대비 2배의 압축 성능을 목표로 개발하였고, 2020년 7월 표준화가 완료되었다. VVC는 UHD 영상, HDR(High Dynamic Range) 영상, WCG(Wide Color Gamut) 영상, 스크린 콘텐츠 영상, 360º 영상 등 다양한 영상을 대해 지원하며, 압축 성능을 높이기 위해 기존의 비디오 코딩에서 사용되지 않았던 여러 새로운 기술이 도입되었다. HEVC에서 사용된 QT(Quaternary Tree) 방식에 BT(Binary Tree)와 TT(Ternary Tree)를 더한 MTT(Multi-Type Tree)와 휘도와 색차 성분의 분할을 서로 다르게 가지는 dual tree[3], 화면 내 예측에서 3개의 참조 샘플 라인 중 하나를 선택하여 사용하는 MRL(Multiple Reference Lines)[4], BM(Bi-lateral Matching) 기반 움직임 벡터 탐색 과정을 통해 merge mode의 예측 정확도를 높이는 DMVR(Decoder side Motion Vector Refinement)[5], 주변환 이후 변환 계수에 한 번 더 변환을 적용하는 LFNST(Low Frequency Non-Separable Transform)[6], 원본 영상과 복원 영상의 오차를 최소화하는 최적의 필터 계수를 유도하여 필터링하는 ALF(Adaptive Loop-Filter)[7] 등 다양한 분야에서 새로운 기술이 채택되었다. 그러나 이러한 새로운 기술의 도입으로 VVC의 부호화기와 복호화기의 복잡도가 상당히 증가하였으며 이에 따라 부호화 성능을 유지하면서 복잡도 감소를 위한 방법이 필요하다.

본 논문에서는 부호화 과정 및 복호화 과정에서 연산량을 줄이기 위해 선형 변환의 선형성을 이용한 고속 역변환 방법을 제안한다. 제안하는 방법은 역양자화 및 역이차변환 이후 역주변환 과정에서 곱셈 연산량에 따라 제안하는 방법과 기존의 역변환 방법을 선택적으로 사용한다.

본 논문은 다음과 같이 구성한다. 2장에서는 변환 부호화에 관해 설명하고, 3장에서는 제안하는 고속 역변환 방법을 설명한다. 그리고 4장에서는 실험 결과를 분석하고, 5장에서 결론을 맺는다.

Ⅱ. 변환 부호화

변환 부호화는 공간 영역의 잔차 신호를 주파수 영역으로 변환함으로써 저주파 영역에 에너지를 집중시켜 압축하는 방법으로, 적은 계수에 더 많은 에너지를 집중시킬수록 효율적인 변환이다. 일반적으로 자연 영상에서 입력된 신호는 주변 화소들과 높은 상관관계를 가지기 때문에 변환을 통해 변환 계수들 간의 상관관계를 낮추는 방식으로 압축을 수행한다. 비상관화 측면에서 가장 최적의 변환은 KLT(Karhunen-Loève transform)라고 알려져 있다. 하지만 KLT는 변환 기저가 입력 신호에 의존적이고, 복잡도가 높아서 실제 비디오 압축 표준에서는 잘 쓰이지 않는다. DCT(Discrete Cosine Transform)가 KLT를 잘 근사하기 때문에 대부분의 비디오 코딩 표준에서는 DCT-2(Discrete Cosine Transform type 2)[8]와 같은 고정 기저를 사용한다. 이처럼 고정 기저를 사용하는 경우, 변환 기저 값이 모두 고정된 상숫값이기 때문에 변환 기저에 대한 정보를 따로 전송할 필요가 없다. DCT-2는 입력 신호가 균질한 경우 좋은 성능을 보여주지만, 자연 영상의 다양한 특성 때문에 DCT-2가 잘 동작하지 않는 경우도 존재한다. 이를 보완하기 위해 여러 변환 기저를 사용하는 방법들이 제안되었다. HEVC에서는 화면 내 예측으로 생성된 4×4블록에 한해 DST-7(Discrete Sine Transform type 7)을 사용한다[9].

VVC의 변환 부호화 관련 기술[10]은 주변환, 이차변환,

변환 스킵 등이 있다. 주변환은 기존의 비디오 압축에서 사용되었던 기술로 예측 후 잔차 신호에 변환 기저를 적용하여 변환 계수를 얻는 기술이다. VVC의 주변환은 직전 비디오 압축 표준인 HEVC와 비교하였을 때, 최대 CTU(Coding Tree Unit)의 크기가 128로 증가함에 따라 최대 변환 크기도 32에서 64로 증가하였다. DCT-7의 적용 범위가 확대되었으며, DCT-8 (Discrete Cosine Transform type 8)이 새로운 변환 기저로 추가되었다. DCT-2는 최대 64-point까지 적용할 수 있고, DST-7과 DCT-8은 최대 32-point까지 적용할 수 있다. 변환 시 가로 방향과 세로 방향의 변환 기저를 서로 다른 변환 기저를 선택할 수 있다. VVC의 주변환은 MTS(Multiple Transform Selection)라고도 불린다. MTS는 휘도 성분에만 적용 가능하며, 색차 성분은 기존과 같이 DCT-2를 사용하여 변환을 수행한다. 수식 (1)은 1차원 N-point 변환을 나타내고, 수식 (2)는 1차원 N-point 역변환을 나타낸다.

$$F(u) = \sum_{x=0}^{N-1} p(x) v_{u,x} \qquad u = 0,1,2,...,N-1 \quad (1)$$

$$p(x) = \sum_{u=0}^{N-1} F(u) v_{u,x} \qquad x = 0,1,2,...,N-1 \quad (2)$$

여기서 $F(u)$는 변환된 신호를 의미하고, $p(x)$는 입력되는 신호를 의미하고, $v_{u,x}$는 변환 기저를 의미한다. 수식 (3), (4)는 1차원 N-point DCT-2, (5), (6)은 각각 1차원 N-point DST-7, 1차원 N-point DCT-8의 변환 기저를 나타낸다.

$$v_{u,x} = \alpha(u) \cos\left(\frac{u(2x+1)\pi}{2N}\right) \quad (3)$$

$$\alpha(u) = \begin{cases} \sqrt{1/N}, & u = 0 \\ \sqrt{2/N}, & u = 1,2,\cdots,N-1 \end{cases} \quad (4)$$

$$v_{u,x} = \sqrt{\frac{4}{2N+1}}\, sin\left(\frac{(2u+1)(x+1)\pi}{2N+1}\right) \quad (5)$$

$$v_{u,x} = \sqrt{\frac{4}{2N+1}}\, cos\left(\frac{(2u+1)(x+1)\pi}{4N+2}\right) \quad (6)$$

변환의 크기가 커짐에 따라 고해상도 영상 부호화의 효율이 높아졌지만, 계산 복잡도가 증가하였다. 이를 해결하기 위해 고주파수 영역에 있는 변환 계수를 0으로 만드는 (zero-out) 방법을 채택하여 계산 복잡도를 낮췄다. Zero-out은 64-point DCT-2와 32-point DST-7, DCT-8에만 적용된다. 64-point DCT-2의 경우 인덱스가 32보다 작은 경우에만 계수가 유지되고 나머지 계수들은 모두 0이 된다. 32-point DST-7, DCT-8의 경우 인덱스가 16보다 작은 경우에만 계수가 유지되고 나머지 계수들은 모두 0이 된다. MTS는 explicit MTS와 implicit MTS로 구분할 수 있다. Explicit MTS는 선택된 변환 기저를 MTS index를 통해 명시적으로 시그널링하고, implicit MTS는 부호화된 정보를 기반으로 변환 기저를 선택한다.

이차변환은 주변환을 통해 얻은 변환 계수에 다시 변환을 적용하여 추가로 중복성을 제거하는 기술이다. 이차변환은 잔차 신호의 왼쪽 위의 저주파 영역에 있는 계수에만 적용되며, 주변환과 달리 비분리 변환을 사용하여 LFNST라고 명명되었다. 이차변환은 화면 내 예측으로 생성되고, 가로, 세로 방향의 변환 기저가 모두 DCT-2인 경우에만 적용할 수 있다. 변환 블록의 크기에 따라 이차변환이 적용되는 영역이 달라지며, 변환 기저도 달라진다.

Ⅲ. 제안하는 고속 역변환 방법

변환 및 역변환은 행렬 곱셈 또는 고속 변환 방법을 통해 구현될 수 있다. 대부분의 비디오 압축 표준에서는 연산량을 줄이기 위해 2차원 변환을 가로, 세로 방향의 1차원 변환으로 나눠서 수행한다. 행렬 곱셈을 이용하는 경우 1차원 N-point 변환에 대해 N^2번의 곱셈 연산이 필요하다. 고속 변환 방법은 주로 변환 기저의 특성을 이용하여 연산 횟수를 줄이는 방법을 이용한다. DCT-2의 경우 짝수 번째 기저 벡터는 대칭성을 지니고, 홀수 번째 기저 벡터는 반 대칭성을 가지는 특성을 이용하여 고속 변환 방법을 구현할 수 있다[9]. DST-7과 DCT-8의 경우도 마찬가지로 변환 기저의 특성을 이용하여 고속 변환 방법을 사용할 수 있다[11]. 이러한 변환 기저의 특성을 이용한 고속 변환 방법은 해당 기저에만 적용할 수 있다는 단점이 있다.

본 논문에서는 선형 변환의 선형성을 이용하여 역변환

시 계산량을 줄이는 고속 역변환 방법을 제안한다. 제안하는 방법은 주변환에 적용되며, 변환 기저 종류와 관계없이 적용할 수 있다. 현재 VVC의 주변환에서 사용하는 DCT-2, DST-7, DCT-8은 모두 정규직교 기저로 역변환은 변환의 전치 행렬을 통해 수행할 수 있으며, 분리 가능한 기저이기 때문에 2차원 변환을 가로, 세로 방향으로 1차원 기저를 적용하여 수행할 수 있다.

2차원 변환을 두 번의 1차원 변환으로 분리하여 수행하는 경우, 가로의 길이가 m, 세로의 길이가 n인 블록 Y에 대해, 세로 방향 기저가 A, 가로 방향 기저가 B인 경우 역변환 과정을 수식 (7)과 같이 표현할 수 있다.

$$X' = A^T Y B \tag{7}$$

수식 (7)에서 X'는 역변환된 블록을 의미한다. 블록 Y가 0이 아닌 계수 N개로 이루어진 경우, Y는 수식 (8)과 같이 Y와 같은 크기의 단 한 개의 0이 아닌 계수를 가지는 N개의 서브 블록의 합으로 표현할 수 있다.

$$Y = y_1 + y_2 + \ldots + y_N \tag{8}$$

그림 1은 3개의 0이 아닌 계수를 가지는 4×4블록을 서브 블록의 합으로 표현하는 예시를 보여준다. 현재 VVC에서 주변환 기저로 사용하는 DCT-2, DST-7, DCT-8은 모두 선형 변환이기 때문에 다음과 같은 성질을 가진다.

$$T(ax + by) = aT(x) + bT(y) \tag{9}$$

수식 (9)에서 T는 변환을 의미하고, x, y는 변환의 입력, a, b는 임의의 상수이다. 수식 (8)과 수식 (9)를 이용하여 수식 (7)의 역변환 과정을 다시 표현하면 아래 수식 (10)과 같다.

$$X' = \sum_{k=1}^{N} A^T y_k B \tag{10}$$

제안하는 역변환 방법의 계산은 다음과 같다. 먼저, N개의 0이 아닌 계수를 가지는 블록 Y를 단 1개의 0이 아닌 계수를 가지는 서브 블록으로 분리한다. 각 서브 블록에 대해 역변환을 수행한 후, 그 결과를 모두 합해 최종 역변환 결과를 생성한다. 하나의 서브 블록에 대해 역변환을 수행하는 방법은 다음과 같다. 가로 길이가 m, 세로 길이가 n인 서브 블록 $y_l (1 \le l \le N)$의 (i,j)번째 위치에 0이 아닌 계수 $x_{i,j}$가 존재할 때, y_l에 대한 역변환은 아래 수식 (11)과 같이 계산할 수 있다.

$$A^T y_l B = \left[\overrightarrow{v_1} \cdots \overrightarrow{v_n}\right] \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & x_{i,j} & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \overrightarrow{w_1}^T \\ \vdots \\ \overrightarrow{w_m}^T \end{bmatrix} \tag{11}$$

수식 (11)에서 $\overrightarrow{v_1} \cdots \overrightarrow{v_n}$은 A^T의 기저 벡터, $\overrightarrow{w_1} \cdots \overrightarrow{w_m}$은 B의 기저 벡터를 의미한다. 먼저 $A^T y_l$을 계산하면 수식 (12)와 같이 결과 행렬의 j번째 열을 제외한 나머지 열은 모두 0이 되고, j번째 열에만 0이 아닌 값이 존재하게 된다.

$$\begin{aligned} A^T y_l &= \begin{bmatrix} v_{1,1} \cdots v_{n,1} \\ \vdots \quad \ddots \quad \vdots \\ v_{1,n} \cdots v_{n,n} \end{bmatrix} \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & x_{i,j} & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & \cdots & v_{i,1} \times x_{i,j} & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & v_{i,n} \times x_{i,j} & \cdots & 0 \end{bmatrix} \end{aligned} \tag{12}$$

수식 (12)에서 $v_{i,j}$는 A^T의 i번째 기저 벡터 $\overrightarrow{v_i}$의 j번째 원소를 의미한다. 수식 (12)에서 구한 $A^T y_l$를 이용하여 최

$$\begin{bmatrix} 3 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

그림 1. 4×4블록의 분할 예시
Fig. 1. Example of 4×4 block

종적으로 $A^T y_l B$를 계산하면 수식 (13)과 같이 된다.

$$A^T y_l B = \begin{bmatrix} 0 \cdots v_{i,1} \times x_{i,j} \cdots 0 \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ 0 \cdots v_{i,n} \times x_{i,j} \cdots 0 \end{bmatrix} \begin{bmatrix} w_{1,1} \cdots w_{1,m} \\ \vdots \quad \ddots \quad \vdots \\ w_{m,1} \cdots w_{m,m} \end{bmatrix}$$
$$= \begin{bmatrix} v_{i,1} \times x_{i,j} \times w_{j,1} \cdots v_{i,1} \times x_{i,j} \times w_{j,m} \\ \vdots \quad\quad\quad \ddots \quad\quad\quad \vdots \\ v_{i,n} \times x_{i,j} \times w_{j,1} \cdots v_{i,n} \times x_{i,j} \times w_{j,m} \end{bmatrix} \quad (13)$$
$$= \begin{bmatrix} v_{i,1} \times x_{i,j} \times \overrightarrow{w_j}^T \\ \vdots \\ v_{i,n} \times x_{i,j} \times \overrightarrow{w_j}^T \end{bmatrix}$$

수식 (13)에서 $w_{i,j}$는 B의 i번째 기저 벡터 $\overrightarrow{w_i}$의 j번째 원소를 의미한다. 제안하는 역변환 방법을 사용하여 하나의 서브 블록에 대해 역변환 수행 시, 수식 (12)에서 n개의 곱셈이 필요하고, 수식 (12)의 계산 결과를 활용하여 수식 (13)을 계산하면 $(n \times m)$개의 곱셈이 필요하다. 따라서 하나의 서브 블록을 역변환하는 데 필요한 곱셈 연산 수는 $(n+(n \times m))$개이고, N개의 0이 아닌 계수를 가지는 변환 블록에 대해 필요한 곱셈 연산 수는 총 $N \times (n+(n \times m))$개다.

기존의 변환 방법과 제안하는 선형성을 이용한 역변환 방법 중 어느 것을 사용하여 역변환을 수행할지는 곱셈 연산량에 따라 결정된다. 변환 블록에 대하여 제안하는 방법으로 역변환 수행 시 VVC 참조 소프트웨어인 VTM(VVC Test Model)에 구현된 역변환보다 곱셈 연산량이 적거나 같은 경우에만 제안하는 선형성을 이용한 역변환 방법으로 역변환을 수행한다. 제안하는 고속 역변환 방법의 총연산량은 변환 블록 내의 0이 아닌 계수의 개수(N)에 따라 달라진다. 따라서 VVC에서 가능한 모든 변환 블록 크기에 대해 현재 VTM에서 사용되고 있는 역변환 방법의 곱셈 횟수를 계산하고, 이를 이용하여 모든 블록 크기에 대한 임곗값을 계산하였다. 임곗값은 각 블록 크기에 대해 제안하는 방법의 곱셈 횟수가 현재 VTM의 곱셈 횟수보다 작거나 같게 되는 최대의 0이 아닌 계수의 개수(N)로 결정하였다. 블록 내의 0이 아닌 계수의 개수가 임곗값 이하인 경우 제안하는 방법을 사용하여 역변환을 수행하고, 임곗값을 초과하는 경우 기존 방법을 사용하여 역변환을 수행한다. 역변환을 수행하기 전 반복문을 이용하여 블록 내의 0이 아닌 계수의 개수를 세고, 개수가 임곗값을 초과하는 즉시 반복문을 종료하고 기존 방법을 이용하여 역변환을 수행한다. 제안하는 방법은 이미 부호화된 정보와 소프트웨어 안에 내장된 변환 기저를 사용하기 때문에 추가적인 플래그가 필요 없다. 제안하는 방법을 부호화기와 복호화기에 적용된다. VVC는 변환 및 역변환 과정 중 16비트 내에서 연산이 수행되도록 절삭 연산을 수행한다. 기존 방법의 경우 각 방향의 역변환을 수행한 후 절삭 연산을 수행하므로, 총 2번의 절삭 연산이 역변환 과정 중에 필요하다. 제안하는 방법은 블록에 대해 가로, 세로 방향의 역변환을 모두 수행한 뒤 1번의 절삭 연산을 수행한다. 절삭 연산의 차이로 인해 하나의 블록에 대해 역변환을 수행할 때, 기존 방법을 사용하여 역변환을 수행하는 결과와 제안하는 방법을 이용하여 역변환을 수행한 결과 간의 차이가 발생할 수 있다.

표 1. VTM-8.2에서의 곱셈 연산량
Table 1. The number of multiplications in VTM-8.2

Width	Height	DCT-2	DST-7/DCT-8
4	4	64	64
4	8	160	320
4	16	480	636
4	32	1632	2736
4	64	3248	N/A
8	4	160	320
8	8	384	1024
8	16	1088	2040
8	32	3520	7008
8	64	7008	N/A
16	4	480	636
16	8	1088	2040
16	16	2816	4064
16	32	8320	13984
16	64	16576	N/A
32	4	1632	2608
32	8	3520	5984
32	16	8320	11952
32	32	22016	29760
32	64	43904	N/A
64	4	2992	N/A
64	8	6240	N/A
64	16	13760	N/A
64	32	32896	N/A
64	64	65664	N/A

표 1은 VTM-8.2(VVC Test Model-8.2)[12]에서 블록 크기

에 따라 역변환을 수행하는 데 필요한 곱셈양을 나타낸 표이다. 첫 번째 열과 두 번째 열은 각각 블록의 가로, 세로 길이를 나타내고, 세 번째 열은 가로, 세로 방향의 변환 기저가 모두 DCT-2인 경우의 곱셈 횟수, 마지막 열은 가로, 세로 방향의 변환 기저가 모두 DST-7 또는 DCT-8인 경우의 곱셈 횟수를 나타낸다.

표 2와 3은 각 블록 크기에 대한 임곗값을 나타내는 표이다. 표 2는 가로, 세로 변환 기저가 모두 DCT-2인 경우와 비교하여 결정된 임곗값이고, 표 3은 가로, 세로 변환 기저가 모두 DST-7 또는 DCT-8인 경우와 비교하여 결정된 임곗값이다. 가로, 세로 변환 기저가 DCT-2와 DST-7 또는 DCT-8의 조합으로 이루어진 경우 표 2의 임곗값을 사용한다.

표 2. 가로, 세로 변환 기저가 모두 DCT-2 이거나 DCT-2와 DST-7 또는 DCT-8의 조합인 경우 블록 크기 별 임곗값
Table 2. Threshold representing the number of non-zero coefficients in each block size, when the horizontal and vertical kernels are DCT-II/DCT-II or other combinations listed in Table 5

		Width						
		1	2	4	8	16	32	64
Height	1	N\A	1	2	3	5	10	10
	2	1	1	2	3	6	11	11
	4	2	2	3	4	7	12	11
	8	3	2	4	5	8	13	12
	16	5	4	6	7	10	15	13
	32	10	7	10	12	15	20	15
	64	10	7	10	12	15	20	15

표 3. 가로, 세로 변환 기저가 모두 DST-7 또는 DCT-8의 조합인 경우 블록 크기 별 임곗값
Table 3. Threshold representing the number of non-zero coefficients in each block size, when the horizontal and vertical kernels are a combination of DST-VII and DCT-VIII (DST-VII/DCT-VIII).

		Width			
		4	8	16	32
Height	4	3	8	9	19
	8	8	14	15	22
	16	7	14	14	22
	32	17	24	25	28

Ⅳ. 실험 결과 및 분석

본 장에서는 제안하는 곱셈 연산을 고려한 선형성을 이용한 역변환 방법에 대한 성능을 보인다. 제안하는 고속 역변환 방법은 VTM-8.2에 구현되었다. 실험은 VVC 공통 실험 조건(Common Test Condition)[13]에 따라 AI(All Intra)와 RA(Random Access) 부호화 환경에서 진행되었다. 실험에 사용된 양자화 파라미터(QP)는 22, 27, 32, 37이다. 실험에 사용한 영상은 표 4와 같다. 표 4에서 4K는 3840 × 2160, 1080p는 1920 × 1080, WVGA는 832 × 480, WQVGA는 416 × 240 해상도의 영상이다.

표 4. 실험에 사용된 영상 정보
Table 4. Information on video sequence

Class	Sequence name	Frame Count	Frame Rate	Bit Depth
A1(4K)	Tango2	294	60	10
	FoodMarket4	300	60	10
	Campfire	300	30	10
A2 (4K)	CatRobot	300	60	10
	DaylightRoad2	300	60	10
	ParkRunning3	300	50	10
B (1080p)	MarketPlace	600	60	10
	RitualDance	600	60	10
	Cactus	500	50	8
	BasketballDrive	500	50	8
	BQTerrace	600	60	8
C (WVGA)	RaceHorses	300	30	8
	BQMall	600	60	8
	PartyScene	500	50	8
	BasketballDrill	500	50	8
D (WQVA)	RaceHorces	300	30	8
	BQSquare	600	60	8
	BlowingBubbles	500	50	8
	BasketballPass	500	50	8

표 5. 실험 결과
Table 5. Experimental result

Class	Sequence	AI			RA		
		Y	EncT	DecT	Y	EncT	DecT
A1	Tango2	0.03%	98%	98%	0.04%	90%	89%
	FoodMarket4	-0.01%	98%	99%	0.04%	87%	82%
	Campfire	0.00%	99%	97%	0.02%	93%	96%
	Average	0.01%	98%	98%	0.03%	90%	89%
A2	CatRobot	0.01%	98%	94%	-0.02%	95%	87%
	DaylightRoad2	0.02%	99%	98%	-0.01%	97%	90%
	ParkRunning3	0.00%	98%	95%	-0.01%	94%	86%
	Average	0.01%	98%	96%	-0.01%	95%	88%
B	MarketPlace	-0.01%	71%	99%	0.01%	99%	84%
	RitualDance	0.01%	68%	98%	0.07%	98%	99%
	Cactus	0.03%	74%	94%	0.01%	98%	97%
	BasketballDrive	0.03%	72%	98%	0.06%	99%	93%
	BQTerrace	0.00%	79%	89%	-0.01%	95%	96%
	Average	0.01%	73%	95%	0.03%	98%	94%
C	BasketballDrill	0.01%	62%	68%	0.02%	98%	77%
	BQMall	0.03%	63%	70%	0.07%	98%	86%
	PartyScene	0.01%	64%	86%	0.00%	99%	91%
	RaceHorces	0.02%	61%	64%	0.01%	99%	94%
	Average	0.02%	63%	72%	0.02%	99%	87%
D	BasketballPass	0.04%	58%	67%	0.01%	99%	95%
	BQSquare	0.01%	59%	79%	0.04%	99%	100%
	BlowingBubbles	0.03%	58%	79%	0.00%	97%	98%
	RaceHorces	0.00%	58%	64%	0.04%	98%	81%
	Average	0.02%	58%	72%	0.02%	98%	93%
Overall		0.01%	74%	85%	0.02%	96%	90%

표 6. AI에서 QP에 따른 실험 결과
Table 6. Experimental result per QP in AI

Class	Sequence	QP=22		QP=37	
		EncT	DecT	EncT	DecT
A1	Tango2	99%	100%	99%	100%
	FoodMarket4	100%	100%	97%	98%
	Campfire	98%	97%	99%	100%
A2	CatRobot	100%	88%	97%	97%
	DaylightRoad2	98%	100%	97%	94%
	ParkRunning3	98%	99%	98%	95%
B	MarketPlace	78%	97%	65%	115%
	RitualDance	72%	94%	63%	109%
	Cactus	80%	94%	69%	108%
	BasketballDrive	79%	85%	68%	109%
	BQTerrace	80%	98%	81%	70%
C	BasketballDrill	65%	71%	62%	72%
	BQMall	69%	105%	62%	62%
	PartyScene	70%	98%	60%	71%
	RaceHorses	59%	57%	63%	76%
D	BasketballPass	58%	60%	58%	79%
	BQSquare	59%	68%	58%	77%
	BlowingBubbles	58%	77%	58%	54%
	RaceHorses	58%	54%	57%	82%

표 5는 제안하는 방법과 VTM-8.2의 부호화 성능과 계산 복잡도를 비교한 결과이다. 부호화 시간 EncT와 복호화 시간 DecT는 다음 수식 (14)와 같이 계산된다.

$$\triangle T = \frac{T_{proposed}}{T_{anchor}} \times 100\,\% \tag{14}$$

제안하는 방법은 기존 방법 대비 약간의 BD-rate 손실을 일으켰지만, 부호화 시간과 복호화 시간을 AI에 대해 각각 26%, 15%, RA에 대해 각각 4%, 10% 감소했다. 또한, 저해상도 영상인 class C, D에 대해 복호화 시간이 많이 감소한 것을 확인할 수 있다.

표 6은 QP가 22인 경우의 실험 결과와 QP가 37인 경우의 실험 결과를 비교한 표이다. 대부분의 영상에서 QP 값이 커질수록 부호화 시간이 감소하는 것을 확인할 수 있다. QP 값이 커질수록 양자화 과정에서 계수가 0이 될 확률이 증가한다. 따라서 역양자화 후 변환 블록 내의 0이 아닌 계

수의 개수가 임곗값 이하일 확률이 증가한다.

V. 결 론

본 논문에서는 부호화기와 복호화기의 계산 복잡도를 낮추기 위해 변환의 선형성을 이용한 고속 역변환 방법을 제안한다. VVC에 새로운 기술들이 추가되면서 부호화기와 복호화기의 복잡도가 매우 증가하였다. 제안된 방법은 역변환에서의 곱셈 연산을 줄이기 위해 변환의 선형성을 이용한다. 제안된 역변환 방법은 VVC의 주변환에 적용되며, 기존의 고속 변환 방법과 달리 변환 기저의 종류와 관계없이 적용할 수 있다는 점에서 의의를 가진다. 제안된 방법은 기존 역변환 방법과의 곱셈 연산량 비교를 통해 선택적으로 수행되며, 추가적인 플래그 전송이 필요하지 않다. 실험은 VTM-8.2에서 진행되었으며, 부호화와 복호화 시간이 AI에서 각각 평균 26%, 15% RA에서 각각 평균 4%, 10% 감소하였다. 또한, 제안하는 고속 역변환 방법과 유사하게 이차변환에서 선형성과 변환의 크기, 이차변환이 적용되는 영역 내의 0이 아닌 계수의 개수를 이용하면 추가적인 복잡도 감소가 가능할 것으로 예상된다.

참 고 문 헌 (References)

[1] B. Bross, J. Chen, S. Liu and Y. -K. Wang, “Versatile Video Coding (Draft 10)“, JVET-S2001, Jul.2020

[2] G. J. Sullivan, J. -R. Ohm, W. -J. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol.22, No.12, pp.1649-1668, Dec. 2012
doi: https://doi.org/10.1109/TCSVT.2012.2221191.

[3] Y. -W. Huang et al., "Block Partitioning Structure in the VVC Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol.31, No.10, pp.3818-3833, Oct. 2021
doi: https://doi.org/10.1109/TCSVT.2021.3088134

[4] J. Pfaff et al., "Intra Prediction and Mode Coding in VVC," IEEE Transactions on Circuits and Systems for Video Technology, Vol.31, No.10, pp.3834-3847, Oct. 2021.
doi: https://doi.org/10.1109/TCSVT.2021.3072430

[5] H. Yang et al., "Subblock-Based Motion Derivation and Inter Prediction Refinement in the Versatile Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol.31, No.10, pp. 3862-3877, Oct. 2021.
doi: https://doi.org/10.1109/TCSVT.2021.3100744

[6] M. Koo, M. Salehifar, J. Lim and S. -H. Kim, "Low Frequency Non-Separable Transform (LFNST)," 2019 Picture Coding Symposium (PCS), Ningbo, China, pp.1-5, 2019.
doi: https://doi.org/10.1109/PCS48520.2019.8954507

[7] M. Karczewicz et al., "VVC In-Loop Filters," IEEE Transactions on Circuits and Systems for Video Technology, Vol.31, No.10, pp.3907-3925, Oct. 2021.
doi: https://doi.org/10.1109/TCSVT.2021.3072297

[8] N. Ahmed, T. Natarajan and K. R. Rao, "Discrete Cosine Transform," IEEE Transactions on Computers, Vol.C-23, No.1, pp.90-93, Jan. 1974.
doi: https://doi.org/10.1109/T-C.1974.223784

[9] M. Budagavi, A. Fuldseth, G. Bjøntegaard, V. Sze and M. Sadafale, "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," IEEE Journal of Selected Topics in Signal Processing. Vol.7, No.6, pp.1029-1041, Dec. 2013.
doi: https://doi.org/10.1109/JSTSP.2013.2270429

[10] X. Zhao et al., "Transform Coding in the VVC Standard," IEEE Transactions on Circuits and Systems for Video Technology, Vol.31, No.10, pp.3878-3890, Oct. 2021.
doi: https://doi.org/10.1109/TCSVT.2021.3087706

[11] Z. Zhang et al., "Fast DST-VII/DCT-VIII With Dual Implementation Support for Versatile Video Coding," IEEE Transactions on Circuits and Systems for Video Technology, Vol.31, No.1, pp.355-371, Jan. 2021.
doi: https://doi.org/10.1109/TCSVT.2020.2977118

[12] Versatile Video Coding Test Model(VTM) 8.2 https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tree/VTM-8.2

[13] Bossen, F. Boyce, J. Suehring, K. Li, X. Seregin, V, “JVET Common Test Conditions and Software Reference Configurations for SDR Video”, JVET-T2010, October 2020.

저 자 소 개

송 현 주

- 2021년 2월 : 세종대학교 컴퓨터공학과 학사
- 2021년 3월 ~ 현재 : 세종대학교 컴퓨터공학과 석사과정
- ORCID : https://orcid.org/0000-0002-4122-3679
- 주관심분야 : Video Compression, Image/Video Processing, Deep Learning, CNN-based Video Coding

이 영 렬

- 1985년 2월 : 서강대학교 전자공학과 학사
- 1987년 2월 : 서강대학교 전자공학과 석사
- 1999년 2월 : 한국 과학기술원 전기·전자공학과 박사
- 1987년 1월 ~ 1994년 2월 : 삼성전자 R&D 센터 Digital Media Lab.
- 1999년 3월 ~ 2001년 8월 : 삼성전자 R&D 센터 Digital Media Lab. 수석연구원
- 2001년 9월 ~ 현재 : 세종대학교 소프트웨어융합대학 컴퓨터공학과 교수
- ORCID : https://orcid.org/0000-0003-2709-8282
- 주관심분야 : Video Compression, Digital Signal Processing, Image Processing, 3-D Video Coding, Deep Learning, CNN-based Video Coding, Object Detection-based Video Coding