

## 참조샘플을 이용한 분할가능한 KL 변환

김남욱, \*이영렬

세종대학교, \*세종대학교

nukim@sju.ac.kr, \*yllee@sejong.ac.kr

## Separable KL transform using reference samples

Nam Uk Kim \*Yung-Lyul Lee

Sejong University \*Sejong University

## 요 약

본 논문에서는 최신 비디오 코딩 기술에서 잔차(Residual)신호 변환을 효율적으로 수행하기 위한 부동 기저(Basis)를 사용하는 방법을 제안한다. 기존의 DCT-II 나 DST-VII 과 같은 고정 기저를 사용하는 방법은 대부분의 잔차신호들에 대해 효과적으로 비상관화(decorrelation)를 수행하지만 복잡한 잔차 신호일수록 성능이 떨어지는 문제가 있었다. 이러한 압축 성능하락 문제를 줄이기 위하여 PCA(Principle Component Analysis) 방법 중 하나인 KLT(Karhunen-Loeve Transform)를 이용하여 부동(floating) 변환 기저를 유도하는 방법을 제안한다. 기존의 KLT 를 이용한 변환 커널 유도 방법들의 문제점인 부호화기 및 복호화기 계산 복잡도를 줄이기 위하여 KL 커널을 분해가능한(Separable) 2 개의 1 차원 커널로 유도하는 방법을 제안하고, 원본 잔차신호와 유사한 텍스처를 찾아 커널을 예측하는 과정을 간소화하는 방법을 제안한다. 제안하는 방법은 HEVC 에서 실험되었으며 정지영상 코딩 Main-Profile 에서 평균 1.4%가량의 BD-PSNR(Bjontegaard Delta-Peak Signal to Noise Ratio) 성능 향상을 보였으며 특히 스크린 콘텐츠 영상에서 최대 4.5%의 성능 향상을 보인다.

## 1. 서론

최근 VVC(Versatile Video Coding)[1] 비디오 코딩 표준화가 진행되면서 잔차신호를 보다 더 효율적으로 압축하기 위한 기술들이 많이 제안되었다. Qualcomm 에서는 EMT(Enhanced Multiple Transform)[2]라는 여러 개의 고정 기저들을 스위칭하여 사용하는 방법과 NSST(Non-Separable Secondary Transform)[3]이라는 변환 계수를 예측 모드의 방향에 따라 이차적으로 변환하는 방법들이 제안되었다. Microsoft 에서는 템플릿 매칭(Template Matching)을 통해 잔차신호와 유사한 예측 블록을 생성하고 해당 예측 블록의 KL 기저를 계산하여 잔차신호 변환에 사용하는 방법이 제안되었다[4]. EMT 와 NSST 와같이 오프라인 학습된 여러 개의 고정(fixed) 기저를 사용하는 방법은 기존의 DCT-II(Discrete Cosine Transform type-II)와 같은 단일 기저를 사용하여 변환하는 방법에 비해

많은 성능 향상을 가져왔다. 하지만 복잡한 패턴의 잔차신호에 대해서는 여전히 압축이 잘 되지 않는 문제가 있다. 특히 화면 내 예측 부호화에서의 잔차신호는 신호의 에너지가 크고 분산이 큰 편이므로 몇 개의 고정 기저들로는 복잡한 패턴과의 상관도(correlation)를 높이기 힘들다. 이러한 문제를 줄이기 위한 방법으로 Microsoft 에서 제안했던 TMKLT(Template Matching Karhunen Loeve Transform)[4] 가 있다. KLT 는 온라인으로 기저를 만드는 방법으로 신호를 완전히 비상관화(decorrelation)시켜 엔트로피 표현을 최소화시키는 방법이다. TMKLT 는 템플릿 매칭으로 유사한 블록들을 수집하여 블록들의 평균을 예측신호로, 블록들의 공분산 (covariance) 계산을 통해 기저를 만들어 잔차신호를 변환한다. TMKLT 는 온라인으로 기저를 생성하기 때문에 복잡한 잔차신호들에 대해서도 높은 압축 성능을 보이는 편이다. 하지만 템플릿 매칭을 사용하기 때문에 복호화기에서 상당한 계산 부담을 가져야 하는

단점이 있다. 그리고 TMKLT 에서는 2 차원 형태의 잔차신호 블록을 1 차원 벡터로 취급하여 공분산행렬을 계산하기 때문에 기저계산, 변환 연산 및 기저의 크기에 있어 계산 복잡도 및 저장 공간이 크다는 단점이 있다. 본 논문에서는 이러한 문제를 줄이기 위하여 크게 두가지 방법을 제안한다. 첫째로는 템플릿 매칭 계산 부담을 없애기 위하여 참조샘플(복원된 신호)로부터 바로 기저를 계산하는 방법을 제안한다. 이 방법을 적용함으로써 복호화기의 복잡도는 TMKLT 방법에 대비하여 1/3 이하로 감소하였으며 압축 성능차이도 크지 않았다. 둘째로 분리가능한 KLT 기저를 계산함으로써 기저 크기를 줄이는 방법을 제안한다. 가로방향과 세로방향의 공분산을 나누어서 계산함으로써 기존의  $M \times N$  블록에서의  $MN \times MN$  크기의 기저크기를  $M \times M$ ,  $N \times N$  크기의 두개의 기저로 줄여서 계산하였다. 해당 방법은 부호화기 및 복호화기의 복잡도 감소를 보였으며 압축성능에서도 큰 차이를 보이지 않았다.

본 논문의 구성은 다음과 같다. 2 절에서는 제안하는 방법에 대해 살펴본 후, 3 절에서는 제안하는 방법의 실험 결과 및 결론을 끝으로 본 논문을 마친다.

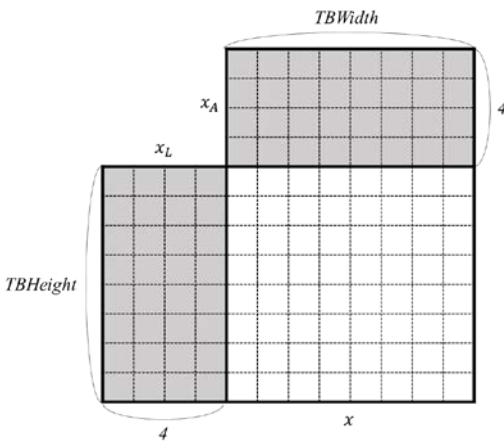


그림 1. 코딩블록  $x$  및 참조샘플 위치

## 2. 제안하는 방법

그림 1 은 코딩하고자 하는 블록  $x$ 와 두개의 참조블록  $x_A, x_L$  을 나타낸다.  $x, x_A, x_L$  은 각각  $H \times W, 4 \times W, H \times 4$  의 크기를 갖으며  $W, H$  는 TB(Transform Block)의 너비와 높이에 해당한다.  $x_A$  영역의 복원신호의 가로방향의 신호특성과  $x$  영역의 잔차신호의 가로방향 신호특성과 유사하다고 가정하고,  $x_L$  영역의 복원신호의 세로방향 신호특성과  $x$  잔차신호의 세로방향 신호특성이 유사하다고 가정한다면 다음과 같은 공분산행렬(Covariance Matrix) 식으로 표현할 수 있다.

$$C_{x,vert} \cong C_{x_A,vert}, \quad C_{x,hor} \cong C_{x_L,hor} \quad (1)$$

$$C_{x,vert} = E[(\bar{x}_i - \mu_x) \cdot (\bar{x}_j - \mu_j)], \quad (2)$$

$i, j$ : vertical vector component

$C_{x,vert}$  는  $x$ 에 대한 공분산행렬(Covariance Matrix)이며 식 (2)와 같이 계산할 수 있다. 위와 같은 가정을 통해 복호화기에서도 추가적인 데이터 없이 변환 기저를 유도를 할 수 있게 되었으며, 변환 기저는 다음 식과 같이 공분산 행렬의 고유벡터 계산을 통해 구할 수 있다.

$$C_{x_A,vert} \Phi_H = \Lambda_H \Phi_H, \quad C_{x_L,hor} \Phi_V = \Lambda_V \Phi_V \quad (3)$$

$\Phi_H, \Lambda_H$ 는  $C_{x_A,vert}$  의 고유벡터들과 고유값들이고,  $\Phi_V, \Lambda_V$ 는  $C_{x_L,hor}$  의 고유벡터들과 고유값들이다. 여기서  $\Phi_H$  와  $\Phi_V$  가 각각 가로방향 1 차원 변환 기저와 세로방향 1 차원 변환 기저에 해당한다. 그러므로 잔차신호  $x$  에 대한 변환은 다음과 같이 수행될 수 있다.

$$y = \Phi_V^T \cdot x \cdot \Phi_H \quad (4)$$

여기서  $C_{x_A,vert}$  와  $C_{x_L,hor}$  는 모두 Hermitian 행렬이므로  $\Phi_H$  와  $\Phi_V$  는 모두 실수이며, 부동소수점 연산을 피하기 위해 정수화 하여 사용할 수 있다.

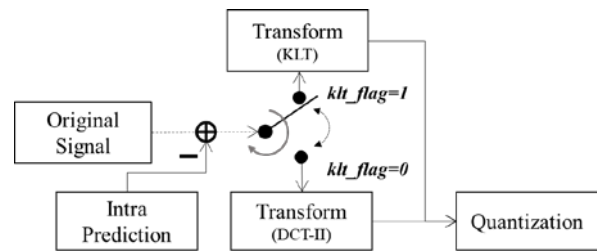


그림 2. 제안하는 방법의 부호화 다이어그램

$$\Phi'_H = [64\sqrt{N} \cdot \Phi_H + \epsilon], \quad \Phi_H \Phi_H^T = I \quad (5)$$

여기서  $N$  은  $\Phi_H$  의 가로방향 또는 세로방향 크기이고  $\epsilon$  는 반올림 상수이다.  $\Phi'_H$  도 유사하게 계산할 수 있다.

그림 2 는 제안하는 방법을 적용한 HEVC 부호화기 블록 다이어그램이다. KLT 와 DCT-II 중 하나를 부호화기에서 결정하여  $klt\_flag$  로 결정된 기저를 부호화기로 알려준다. 이렇게 잔차신호 변환을 스위칭 방식으로 구현한 이유는 KLT 만 단독으로 적용하기에는 식 (1)의 가정이 강하다고 볼 수도 있고 또는 잔차신호  $x$  의 분포가 매우 불규칙적이기 때문에 DCT-II 와 같이 저 차원 기저에서 높은 자기상관도를 갖는 기저를 선택적으로 사용하는 것이 유리하기 때문이다.

### 3. 실험 결과 및 결론

제안하는 방법은 HEVC HM16.9 Software 에서 구현하였으며 고유벡터 계산에는 Eigen C++ 라이브러리[5]를 사용하였다. HEVC 정지영상 CTC 를 사용하여 인코딩을 하였고, 양자화 파라미터는 22, 27, 32, 37 을 사용하여 BD-Rate 를 계산하였다.

테스트에는 4K 영상 Campfire, DaylightRoad, 1080P 영상 Cactus, BasketBallDrive, 스크린 콘텐츠 영상 ArenaOfValor, SlideEditing 을 사용하였다.

표 1. 실험 결과

영상	BD Rate	EΔT	DΔT
Campfire	-0.5%	385%	116%
DaylightRoad	-0.7%	422%	129%
Cactus	-0.2%	441%	109%
BasketBallDrive	-0.4%	435%	
ArenaOfValor	-4.5%	386%	184%
SlideEditing	-2.3%	432%	112%

BD-Rate 는 두 개의 코덱간의 압축성능 차이를 비교하는 metric 으로 [6]를 참조하여 계산하였고, EΔT 및 DΔT 는 다음 과 같다.

$$E\Delta T = \frac{\text{Proposed 인코딩시간}}{\text{HM 인코딩시간}} \% \quad (6)$$

$$D\Delta T = \frac{\text{Proposed 디코딩시간}}{\text{HM 디코딩시간}} \% \quad (7)$$

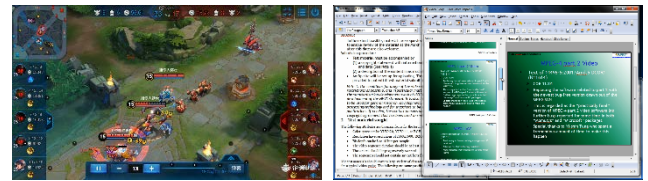
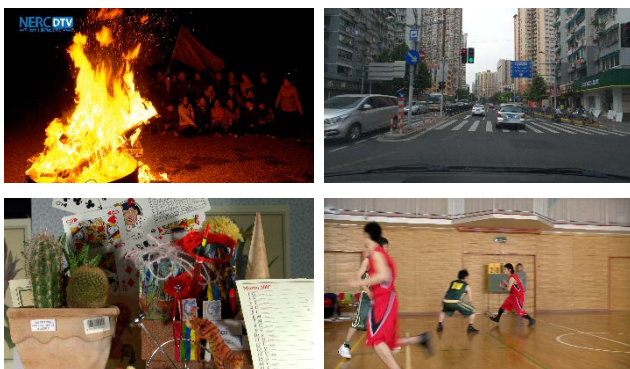


그림 3. 테스트 영상 (왼쪽 위부터 순서대로 Campfire, DaylightRoad, Cactus, BasketballDrive, ArenaOfValor, SlideEditing)

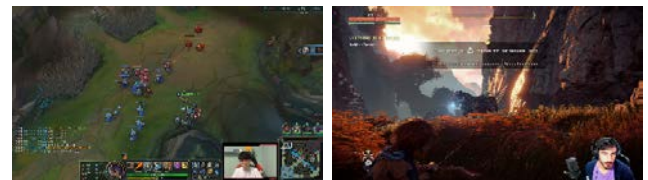


그림 4. 테스트 영상 League of legends (왼쪽), Horizon zero dawn(오른쪽)

표 1 의 실험 결과를 참조하면 일반적인 영상에서는 KLT 를 사용한 제안하는 방법이 그렇게 높은 성능 향상을 보이지 않았음을 알 수 있었지만 스크린 콘텐츠와 같이 규칙적인 뚜렷한 패턴을 갖는 영상에서는 높은 압축 성능 향상을 보인 것을 확인할 수 있었다. 특히 ArenaOfValor 와 같은 탑 뷰(Top-View) 형태의 게임 영상에서는 선형적인 에지(edge) 패턴들이 많이 발견되어 4.5% 정도의 높은 성능 향상을 보인 것으로 생각된다. 이와 같은 생각이 맞는지 대략적으로 알아보기 위하여 두 가지 게임 영상을 추가적으로 테스트해보았다. 첫번째로는 League of legends 게임 영상으로 Arena of valor 와 동일한 장르의 게임이며 선형적인 에지 패턴들이 많이 두드러지는 그래픽의 게임이다. 두번째로는 Horizon zero dawn 게임 영상으로 3 인칭 시점이며, 첫번째 영상의 게임보다는 비교적 더 복잡한 텍스처 디테일 및 비선형적인 경계를 더 많이 갖고 있다. 테스트 결과는 League of legends 영상이 -1.4% Horizon zero dawn 영상이 -0.3%의 BD-rate 향상을 보였다.

제안하는 방법은 참조 샘플을 이용하여 가로 및 세로방향의 KLT 를 생성하여 잔차신호 변환을 수행하는 방법이다. 스크린 콘텐츠에서 많은 압축률 향상을 보였으며 그 이유로는 식(1)의 가정에 부합하는 선형적인 패턴들이 많이 나타나기 때문으로 보인다. 비선형적인 패턴에 대해서 성능을 얻기 위해서는 신호의 분산 계산을 고차원적( $N^d$ ,  $N$  이 2 차원 신호의 너비 및 높이일 경우)으로 수행하거나 블록을 분할하여 각각의 블록들이 서로 다른 KL 기저를 생성해야 할 것으로 생각된다. 그리고 이러한 방법들의 계산 복잡도 문제를 해결하는 것이 다음 연구의 과제가 될 것이다.

#### 4. 참조

- [1] JVET-R2001, Versatile Video Coding(Draft 9); [http://phenix.it-sudparis.eu/jvet/doc\\_end\\_user/current\\_document.php?id=10155](http://phenix.it-sudparis.eu/jvet/doc_end_user/current_document.php?id=10155)
- [2] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li and W. Chien, "Enhanced Multiple Transform for Video Coding," 2016 Data Compression Conference (DCC), Snowbird, UT, 2016, pp. 73-82, doi: 10.1109/DCC.2016.9.
- [3] X. Zhao, J. Chen, A. Said, V. Seregin, H. E. Egilmez and M. Karczewicz, "NSST: Non-separable secondary transforms for next generation video coding," 2016 Picture Coding Symposium (PCS), Nuremberg, 2016, pp. 1-5, doi: 10.1109/PCS.2016.7906344.
- [4] C. Lan, J. Xu, W. Zeng, G. Shi and F. Wu, "Variable Block-Sized Signal-Dependent Transform for Video Coding," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 8, pp. 1920-1933, Aug. 2018, doi: 10.1109/TCSVT.2017.2689032.
- [5] <http://eigen.tuxfamily.org/>
- [6] G. Bjøntegaard, Calculation of Average PSNR Differences Between RD Curves, ITU-T Q.6/SG16, Document VCEG-M33, 2001.

#### 5. 감사의 글

이 논문의 일부는 2020 년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2018R1D1A1B07045156)