High Efficiency Intra Video Coding Based on Data-driven Transform

Na Li, Yun Zhang, *Senior Member, IEEE,* C.-C. Jay Kuo, *Fellow, IEEE*

***Abstract*—In this work, we propose a high efficiency intra video coding based on data-driven transform, which is able to learn the source distributions of intra prediction residuals and improve the subsequent transform coding efficiency. Firstly, we model learning based transform design as an optimization problem of maximizing energy compaction or decorrelation. A data-driven Subspace Approximation with Adjusted Bias (Saab) transform is analyzed and compared with the mainstream Discrete Cosine Transform (DCT) on their energy compaction and decorrelation capabilities. Secondly, we propose a Saab transform based intra video coding framework with offline Saab transform learning. Meanwhile, intra mode dependent Saab transform is developed. Then, Rate-Distortion (RD) gain of Saab transform based intra video coding is theoretically and experimentally analyzed in detail. Finally, three strategies that apply the Saab transform to intra video coding are developed to improve the coding efficiency. Experimental results demonstrate that the proposed 8×8 Saab transform based intra coding can achieve Bjønteggard Delta Bit Rate (BDBR) from -1.19% to -10.00% and -3.07% on average as compared with the mainstream 8×8 DCT based intra coding. In case of variable size transform unit setting, the proposed algorithm achieves BDBR from -0.17% to -6.09% and -1.80% on average, which outperforms DCT-based and convolutional neural network-based transform schemes.**

***Index Terms*—Video Coding, Saab Transform, Transform Coding, Intra Prediction, Energy Compaction, Decorrelation.**

I. Introduction

VIDEO data contributes the most data volume increase in the era of big data due to its realistic representation and wide applications. Video resolutions are expected to be extended from High Definition (HD) to 4K/8K Ultra-HD (UHD) in the near future. Meanwhile, High Dynamic Range (HDR), holograph Three-Dimension (3D) and Virtual Reality (VR) videos are booming as they are capable of providing realistic, 3D and immersive visual experiences. In addition, the usage of these video applications is boosting dramatically with an increasing number of video devices connected to the internet or Internet of Things (IoT), e.g., TV, laptops, smartphones, surveillance cameras, drones, etc. Along with the increase of both the usage and quality of videos, the volume of global video data doubles every two years, which is the bottleneck for video storage and transmission over network. In the development of video coding standards, from MPEG-2, H.264/Advanced Video Coding (AVC), H.265/High Efficiency Video Coding (HEVC) [1] to the latest Versatile Video Coding (VVC) [2], the video compression ratio is doubled almost every ten years. Although researchers keep on developing video coding techniques in the past decades, there is a big gap between the increasing ratios of the compression efficiency and the global video data volume [3]. Coding optimizations for higher efficiency are highly desired.

In the latest three generations of video coding standards, hybrid video coding framework has been adopted, which is composed of predictive coding, transform coding and entropy coding. Firstly, predictive coding is to remove the spatial and temporal redundancies of video content on the basis of exploiting correlation among spatial neighboring blocks and temporal successive frames. Higher prediction accuracy leads to smaller and fewer residuals to be encoded, and thus leads to a higher compression ratio. The predictive coding can be classified as intra prediction or inter prediction based on whether the reference pixels are from spatial or temporal domains. Secondly, transform coding [4] that mainly consists of transform and quantization is to transform residuals from predictive coding to a spectral domain, and then quantize the spectral coefficients to further exploit spatial and perceptual redundancies. For example, Human Visual System (HVS) is generally more sensitive to low frequency than high frequency signals, where larger quantization scales could be given. Finally, entropy coding exploits the statistical property of transformed coefficients so as to approach its entropy. Generally, it encodes symbols of higher probability with fewer bits and encodes symbols of lower probability with more bits. In this paper, we develop a data-driven transform to improve the coding efficiency of the hybrid video coding framework.

Karhunen-Loéve Transform (KLT) is an ideal transform for energy compaction and decorrelation, which requires calculating an autocorrelation matrix for each source input block. In video coding, the autocorrelation matrix shall be encoded and transmitted with associated transformed coefficients, which brings additional bit rate overhead while using KLT in video coding. The outstanding energy compaction and decorrelation capabilities of KLT attract researchers to study data-driven transform. Dvir *et al.* [5] constructed a new transform from an eigen-decomposition of a discrete directional Laplacian system matrix. Lan *et al.* [6] trained one dimensional KLT through searching patches similar to the current block from reconstructed frames with computational overhead. As a derivation of the secondary transform, Koo *et al.* [7] learned a number of non-separable transforms from both video sequences and still images, and adopted five of them with the lowest Rate-Distortion (RD) cost as the final transform. Cai *et al.* [8] estimated the residual covariance as a function of the coded

Na Li and Yun Zhang are with the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, e-mail:{na.li1, yun.zhang}@siat.ac.cn

C.-C. Jay Kuo is with Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, California, USA, e-mail:cckuo@sipi.usc.edu.

boundary gradient, considering prediction is very sensitive to the accuracy of the prediction direction in the image region with sharp discontinuities. Wang *et al.* [9] proposed to optimize transform and quantization together with RD Optimization (RDO). Zhang *et al.* [10] designed a highly efficient KLT based image compression algorithm. Graph Based Transform (GBT) was proposed as a derivation of the traditional KLT in [11], which incorporated Laplacian with structural constraints to reflect the inherent model assumptions about the video signal. Arrufat *et al.* [12] designed a KLT based transform for each intra prediction mode in Mode-Dependent Directional Transform (MDDT). Takamura *et al.* [13] proposed a non-separable mode-dependent transform and created offline 2D-KLT kernels for different intra prediction modes and Transform Unit (TU) sizes. In these studies, researchers focused on generating the autocorrelation matrix for the data-dependent KLT and optimizing the data-dependent KLT with the constrained autocorrelation matrix. However, the autocorrelation matrix is fixed for dynamic block residuals, which may not be accurate. Meanwhile, the transform kernels shall be transmitted to the client.

Discrete Cosine Transform (DCT) is similar to KLT on energy compaction when the input signal obeys Gaussian distribution. Due to its good energy compaction capability and relatively low complexity, DCT has been widely used in image and video coding standards, including MPEG-1, MPEG-2, MPEG-4, H.261, H.262 and H.263. H.264/AVC and later standards adopted Integer DCT (ICT), which replaced complicated floating point multiplications in DCT with light integer additions and shifts for lower complexity and hardware cost. Since DCT transform kernels are fixed and difficult to adapt to all video contents and modes, more advanced DCT optimizations were proposed to improve the transform coding efficiency through jointly utilizing multiple transforms and RDO [14]. For HEVC intra video coding, an integer Discrete Sine Transform (DST) [4] was further applied to 4×4 TU in luminance residuals. Han *et al.* [15] proposed a variant of the DST named Asymmetric DST (ADST) by considering the prediction direction and boundary information. Their transform kernels were fixed.

Furthermore, due to the diversity of video contents and various distributions, multiple transform kernels from DCT/DST families were jointly utilized to enhance the coding efficiency. Zhao *et al.* [16] presented the Enhanced Multiple Transform (EMT) by selecting the optimal transform from multiple candidates based on the source properties and distributions. EMT is intra mode dependent, where the DCT/DST transform kernels were selected based on the direction of the intra angular mode. As the coding efficiency of EMT is achieved at cost of high computational complexity burden at the encoder side, Kammoun *et al.* [17] proposed an efficient pipelined hardware implementation. In addition, EMT was simplified as Multiple Transform Selection (MTS) [18] and adopted in the VVC. Zeng *et al.* [19] presented a two-stage transform, where coefficients from all directional transforms at the first stage were re-arranged appropriately and input to the secondary transform for higher energy compaction. Considering that multi-kernel transform and non-separable transform were able to capture diverse directional texture patterns more efficiently, EMT [16] and Non-Separable Secondary Transform (NSST) [20] were combined to provide a better coding performance for VVC standard. Similarly, Zhang *et al.* [21] presented a method on Implicit-Selected Transform (IST) to improve the performance of transform coding for AVS-3. Pfaff *et al.* [22] applied mode-dependent transform with primary and secondary transforms to improve transform coding in HEVC. These studies utilized multiple transform kernels to improve the transform efficiency at the cost of multiple pass transform computations. To reduce the computational complexity, Park *et al.* [23] introduced fast implementation methods for n-point DCT-V and DCT-VIII. Garrido *et al.* [24] proposed a hardware architecture to accelerate different types of DCT/DST (i.e. DCT-II, DCT-VIII, and DST-VII) with variable unit sizes more effectively. DCT is a pre-defined and fixed transform to approach KLT's performance for Gaussian distributed signals. However, this Gaussian distribution assumption cannot be always guaranteed due to diverse video contents and variable block patterns, which enlarged the performance gap between DCT and the optimal KLT. In addition, the "try all and select the best" strategy, which tried multiple transform kernels for different source distributions and selected the optimal one with RD cost comparison, significantly increased the coding complexity.

Machine learning based transform is a possible solution to have a good trade-off between the data-dependent KLT and the fixed DCT, which can improve the video coding performance. Lu *et al.* [25] proposed an end-to-end learning framework for video compression, in which the residual encoder/decoder, motion estimation, MV encoder/decoder were improved with neural networks due to their powerful non-linear representation ability. Yang *et al.* [26] proposed a Convolutional Neural Network (CNN)-based non-linear transform for HEVC intra coding, in which the CNN based transform was trained by minimizing the loss from the reconstruction difference and energy compaction. However, deep learning lacks of interpretability and has higher computational complexity due to intensive convolutional operations. Motivated by the analyses on non-linear activation of CNN [27], [28], Kuo *et al.* [29] proposed a data-driven Subspace Approximation with Augmented Kernels (Saak) transform. The transform kernel was KLT kernel augmented with its negative so as to resolve the sign confusion problem, which was derived from the second-order statistics of input in one-pass feedforward manner. Visual quality assessment for compressed images was developed by exploiting the energy of Saak transformed coefficients [30]. Furthermore, Kuo *et al.* [31] proposed an interpretable and feedforward learning for data representation, called Subspace Approximation with Adjusted Bias (Saab) transform, which is statistics-centric and unsupervised. The sign confusion problem was solved by shifting the transform input with a relatively large bias [31]. Also, the Saab transform interpreted the cascaded convolutional layers as a sequence of approximating spatial-spectral subspaces. Following the Saab, a Successive Subspace Learning (SSL) principle was proposed to learn interpretable models for object recognition [32] and classification [33].

The Saab transform is a data-dependent, multi-stage and

non-separable transform, which can be applied to recognition tasks as well as image representation due to its superior energy compaction capability. In [34], Saab transforms were learned from video coding dataset and had potentials to outperform DCT on energy compaction capability while representing variable block-size Intra prediction residuals. However, the ultimate coding performance of using Saab transform shall be further investigated.

In this work, we propose a Saab transform based intra coding, which learns the source distributions of intra prediction residuals and improves the transform coding efficiency. To our best knowledge, it is the first work tries to apply the Saab transform to improve the video coding efficiency. In addition to the achieved coding gains, our major contributions are:

- The energy compaction and decorrelation capabilities of Saab transform are compared with the DCT and KLT. Then, RD performance for the Saab transform based intra coding is analyzed theoretically and experimentally.
- We propose a data driven transform based video coding framework, which consists of off-line learning for Saab transform kernel and Saab transform based video encoder/decoder.
- Intra mode dependent Saab transform and three strategies that integrate Saab and DCT kernels to intra video coding are developed for higher coding efficiency.

The paper is organized as follows. Saab transform and its performance are analyzed in Section II. A framework of Saab transform based intra video coding, intra mode dependent Saab transform and three integration strategies are illustrated in Section III. Then, the RD performance of the Saab transform based intra coding is analyzed in detail in Section IV. Experimental results and analyses are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Data-driven transform and analysis

A. Problem Formulation

Transform in image/video compression aims to improve energy compaction and decorrelation for the transformed coefficients. Let $\boldsymbol{x} = \{x_i\}$ be an input source, and it is forward transformed to output $\boldsymbol{y} = \{y_k\}$ in a spectral domain as

$$y_k = \sum_{i=0}^{K-1} x_i a_{k,i} \quad or \quad \boldsymbol{y} = \mathbf{A}\boldsymbol{x}, \tag{1}$$

where $a_{k,i}$ is a transform element in the forward transform kernel $\mathbf{A}$. The inverse transform from y_k to x_i is presented as

$$x_i = \sum_{k=0}^{K-1} y_k u_{k,i} \quad or \quad \boldsymbol{x} = \mathbf{U}\boldsymbol{y}, \tag{2}$$

where $u_{k,i}$ is a transform element in the inverse transform matrix $\mathbf{U}$. $\mathbf{U}$ is an inverse matrix of $\mathbf{A}$ satisfying $\mathbf{U} = \mathbf{A}^{-1}$ and $\mathbf{UA} = \mathbf{AU} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. If the transform is orthogonal, it means the rows of transform matrix are an orthogonal basis set and the inverse transform matrix $\mathbf{U}$ satisfies $\mathbf{U} = \mathbf{A}^{-1} = \mathbf{A}^{T}$.

In the data-driven transform, the transform kernel $\mathbf{A}$ is estimated from data samples $\mathcal{D} = [\mathbf{d}_0, ..., \mathbf{d}_{T-1}]$. Generally,

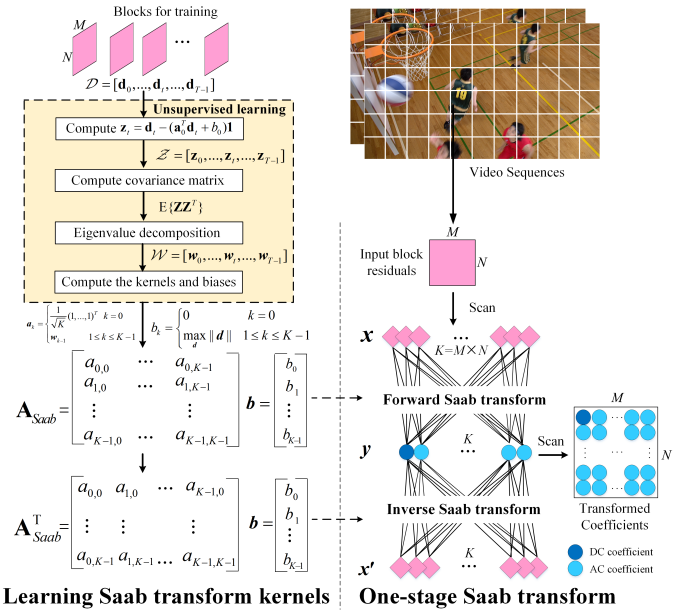


Fig. 1. Diagram of learning and testing the one-stage Saab transform.

different transform kernels can be learned from subspaces of data samples with different learning strategies. Given a transform kernel set $\mathcal{A}$, the optimal transform $\mathbf{A}^*$ is selected from $\mathcal{A}$ through solving the optimization problem expressed as

$$\mathbf{A}^* = \arg\max_{\mathbf{A}_i \in A} M(\boldsymbol{y}), \tag{3}$$

where $M(\boldsymbol{y})$ indicates a target transform performance of the transformed coefficients $\boldsymbol{y}$. The optimal transform could be achieved by maximizing the value of $M(\boldsymbol{y})$. In video coding, $M(\boldsymbol{y})$ can be defined as but not limited to the energy compaction or decorrelation capabilities, which relates to the compression efficiency. For example, DCT is predefined as an orthogonal transform for all block residuals. KLT kernel is derived by maximizing the decorrelation, which varies for each block residual. Although KLT outperforms DCT on energy compaction and decorrelation, KLT based video coding needs to transmit kernel information for each block, which causes a large number of overhead bits. Saab transform is able to learn statistics for a large number of blocks, which is a possible solution to improve the coding performance of existing codecs.

B. Saab Transform

Saab transform [31] is conducted as a data-dependent, multi-stage and non-separable transform in a local window to get a local spectral vector. Diagram of testing and training the 2D one-stage Saab transform is presented in Fig. 1, where the left part is learning Saab transform kernels and the right part is one-stage Saab transform.

Given an $M \times N$ dimensional input $\boldsymbol{x}$ in the space $\mathbf{R}^{M \times N}$, which is rearranged into a vector in lexicographic order as

$$\begin{aligned} \boldsymbol{x} = [&x_{00}, x_{01}, ..., x_{0,N-1}, x_{1,0}, x_{1,1}, ..., x_{1,N-1} \\ &..., x_{M-1,0}, ..., x_{M-1,N-1}]^T. \end{aligned} \tag{4}$$

Then, output transformed coefficients from Saab transform can be computed as

$$y_k = \sum_{j=0}^{K-1} a_{k,j} x_j + b_k = \mathbf{a}_k^T \boldsymbol{x} + b_k, \tag{5}$$

where $\mathbf{a}_k$ are transform kernels and $\mathbf{b}_k$ is the bias, K=$M \times N$, $k = 0, 1, ..., K-1$. In Saab transform, DC kernel and AC kernels are composed of $\mathbf{A}_{Saab}$=$\{\mathbf{a}_k\}_0^{K-1}$ and $\boldsymbol{b}$=$\{\mathbf{b}_k\}_0^{K-1}$, which are unsupervisedly learned from the training dataset $\mathcal{D} = [\mathbf{d}_0, ..., \mathbf{d}_{T-1}]$, as illustrated at the left part of Fig. 1. The number of samples in the training dataset, i.e., T, is around 60K. $\boldsymbol{y}$ is generally organized as a coefficients grid. In the forward Saab transform, for input $\boldsymbol{x}$ in the space $\mathbf{R}^{M \times N}$, the DC and AC coefficients for $\boldsymbol{y}$ are computed separately as

- *DC Coefficient*: The DC coefficient is computed with $y_0 = \frac{1}{\sqrt{K}} \sum_{j=0}^{K-1} x_j + b_0$, where DC kernel $\mathbf{a}_0 = \frac{1}{\sqrt{K}}(1, ..., 1)^T$ and the corresponding bias $b_0 = 0$.
- *AC Coefficients*: Firstly, $\mathbf{z}'$ is computed as $\mathbf{z}' = \mathbf{x} - (\mathbf{a}_0^T \mathbf{x} + b_0)\mathbf{1}$, where $\mathbf{1} = \mathbf{c}/||\mathbf{c}||$, and $\mathbf{c} = (1, 1, \cdots, 1, 1)$ is the constant unit vector. Then, AC coefficient is computed as $y_k = \sum_{j=0}^{K-1} a_{k,j} z'_j + b_k = \mathbf{a}_k^T \mathbf{z}' + b_k$. AC kernel $\mathbf{a}_k$ is the eigenvectors $\boldsymbol{w}_k$ of the covariance matrix $\mathbf{C} = E\{\mathbf{Z}\mathbf{Z}^T\}$, where $\mathcal{Z} = [\mathbf{z}_0, ..., \mathbf{z}_t, ..., \mathbf{z}_{T-1}]$ and $\mathbf{z} = \mathbf{d} - (\mathbf{a}_0^T \mathbf{d} + b_0)\mathbf{1}$ are derived from the training dataset $\mathcal{D}$. The corresponding bias is $b_k = \max_{\mathbf{d}} ||\mathbf{d}||$.

Since one-stage Saab transform is orthogonal [31], the inverse Saab transform kernel $\mathbf{U}_{Saab}$ is the transpose of the forward Saab transform kernel, noted as $\mathbf{U}_{Saab} = \mathbf{A}_{Saab}^T$. The vector $\boldsymbol{y}$ is inversely transformed into $\boldsymbol{x}'$ with $x_k{}' = \mathbf{a}_k{}'(\{y_k\}_1^{K-1} - \{b_k\}_1^{K-1}) + y_0$, where $\mathbf{a}_k{}'$=$[a_{1,k}, a_{2,k}, ..., a_{K-1,k}]$, $k = 0, 1, ..., K-1$.

One-stage Saab transform can be regarded as a combination of KLT and DCT. The DC coefficient of one-stage Saab transform equals to the DC coefficient of DCT. The AC coefficients can be regarded as the first K-1 elements from a variant KLT, where a bias is added to all input elements before transform. Thus, one-stage Saab transformed coefficients can be regarded a combination of DC coefficient from DCT and AC coefficients from a variant KLT. In this case, input source distribution, such as directionality of intra prediction residuals in Section III-B, can be reflected by AC coefficients of the Saab transform. In addition, multi-stage Saab transform [31] can be built by cascading multiple one-stage Saab transforms to extract high-level features. Similarly, multi-stage inverse transform is derived correspondingly by cascading multiple one-stage inverse Saab transforms. To solve the sign confusion problem, the input of the next stage is shifted to be positive by the bias. In this paper, we explore the potentials of Saab transform for video representation and coding.

C. Energy Compaction and Decorrelation Capabilities of Saab Transform

In video compression, one discipline of transform is to save bits by transforming input $\boldsymbol{x}$ to frequency domain with fewer non-zero elements, which is noted as energy compaction. The energy compaction is mathematically defined as [35]

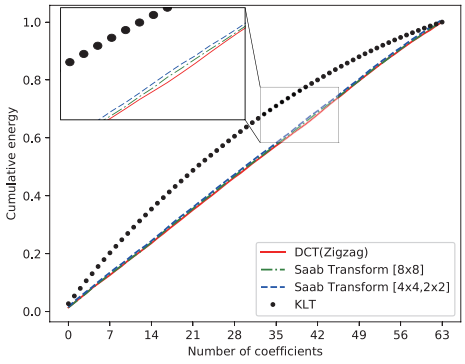


Fig. 2. Energy compaction $E(\boldsymbol{y})$ comparison among KLT, DCT and Saab transforms.

$$E(\boldsymbol{y}) = \frac{\sum_{k=0}^{i} y_k^2}{\sigma_{\boldsymbol{x}}^2}, \tag{6}$$

where $\sigma_{\boldsymbol{x}}^2$ is the variance of the input $\boldsymbol{x}$ and i is the number of coefficients. Thus, we shall compare the energy compaction capability of 8×8 transforms for video coding, including KLT, DCT, one-stage Saab transform and two-stage Saab transform.

In case of two-stage Saab transform, a 4×4 transform was cascaded with a 2×2 transform kernel to form a 8×8 transform. We first mapped one 4×4 subblock to one DC and 15 AC coefficients. Afterwards, we mapped a spatial-spectral cuboid, which had 2×2 dimensions in spatial and 16 dimensions in spectral domain, to a spectral vector with 64 dimensions. Such that, the two-stage Saab transform output one DC and 63 AC coefficients. In training, at the first stage 4×4 Saab transform, 15 AC kernel elements were learned from conducting Principal Components Analysis (PCA) on vectors with 16 elements, which were re-arranged from 4×4 sub-blocks with DC components removed. At the second stage 2×2 Saab transform, 63 AC kernel elements were learned from conducting the PCA on vectors with 64 elements that were re-arranged from a 2×2×16 spatial-spectral cuboid with the DC component removed.

For fair comparison, both the one-stage (denoted as "Saab Transform [8×8]") and the two-stage Saab transform (denoted as "Saab Transform [4×4, 2×2]") were learned from over 70K 8×8 luminance block residuals of "Planar" mode collected from encoding the video sequence "FourPeople" with Quantization Parameters (QPs) in {22, 27, 32, 37} in HEVC. Blocks of "Planar" mode were analyzed because "Planar" mode had the highest probability of being selected as the best among all 35 Intra modes [36]. Only one Saab transform was trained offline and applied to transform all blocks in "Saab Transform [8×8]" and "Saab Transform [4×4, 2×2]", respectively. Then, another 500 8×8 block residuals were randomly selected to compute the energy compaction of KLT, DCT and the two Saab transforms. Fig. 2 shows the energy compaction $E(\boldsymbol{y})$ comparison among KLT, DCT, the one-stage

TABLE I
DECORRELATION COST $C(\boldsymbol{y})$ COMPARISON AMONG KLT, DCT AND SAAB TRANSFORMS.

Sequence	QP	Decorrelation cost $C(\boldsymbol{y})$			
		DCT	Saab Transform		KLT
			[8×8]	[4×4,2×2]	
BasketballDrill	22	581.78	574.26	582.46	0
	27	1453.53	1309.17	1357.91	
	32	3266.85	2691.37	2769.59	
	37	5886.47	4574.13	4802.21	
RaceHorses	22	588.19	586.50	591.71	
	27	1385.87	1361.44	1382.41	
	32	3911.97	3814.71	3887.13	
	37	8281.33	7818.34	8148.87	
FourPeople	22	358.67	371.54	381.89	
	27	1042.21	1010.10	1054.60	
	32	1348.89	1331.15	1394.65	
	37	2842.04	2691.33	2950.05	
Average		2578.98	2344.50	2441.96	

Saab transform and two-stage Saab transform, and we can have the following two key observations: 1) There is a large gap between the KLT and DCT on energy compaction, since KLT is specified for each block and DCT is fixed for all blocks. Theoretically, DCT's energy compaction may approach to that of KLT for Gaussian distributed signals. However, this condition is not always satisfied in practical when encoding various video contents. 2) The one-stage and two-stage 8×8 Saab transforms, whose kernels are also fixed transform and learned offline, perform a little better than DCT in energy compaction. Therefore, 8×8 Saab transform has the potential to improve the coding efficiency of the HEVC.

Another discipline of transform in video coding is removing redundancy or correlation of the input signals $\boldsymbol{x}$ via transform, i.e., decorrelation. To evaluate the decorrelation capability of a transform, we measure the decorrelation cost of transformed coefficients $\boldsymbol{y}$ with its covariance as

$$\begin{aligned} C(\boldsymbol{y}) &= \sum_{i \neq j} |cov(y_i, y_j)| \\ &= \sum_{i \neq j} |E\{(y_i - \mu_i)(y_j - \mu_j)\}|, \ 0 \leq i, j \leq K-1 \end{aligned} \quad , \tag{7}$$

where $cov(y_i,y_j)$ is the covariance between y_i and y_j, $i \neq j$. μ_i and μ_j are the mean of y_i and y_j. Smaller $C(\boldsymbol{y})$ value indicates a better decorrelation capability of a transform. The value of $C(\boldsymbol{y})$ in the transform domain of KLT is 0, which means y_i and y_j are completely independent and their correlation is 0, as $i \neq j$. In other words, redundancy is minimized as 0 among the elements y_i in the transformed coefficients.

Experimental analyses on the decorrelation capability of one-stage Saab transform, two-stage Saab transform and DCT for 8×8 block residuals were performed. Saab transform kernels were learned from three video sequences in {"BasketballDrill", "RaceHorses", "FourPeople"}. For each video sequence, the value of $C(\boldsymbol{y})$ was computed for 500 of block coefficients randomly selected in the transform domain of each transform. As shown in Table I, the average decorrelation costs $C(\boldsymbol{y})$ of KLT, one-stage Saab transform, two-stage Saab transform and DCT are 0, 2344.50, 2441.96 and 2578.98. Lower $C(\boldsymbol{y})$ value indicates better decorrelation performance. We can have the following three findings: 1) the decorrelation cost $C(\boldsymbol{y})$ of KLT is 0, which is the best. 2) Saab transform generally performs better than DCT for most sequences and QPs. But for small QP, i.e., 22, and some sequences, e.g., "FourPeople", DCT performs better on the decorrelation than the two-stage Saab transform. 3) One-stage Saab transform is better than two-stage Saab transform on decorrelation, mainly because increasing the number of stages in Saab transform will be beneficial to extract high-level features, but not to reduce the decorrelation cost $C(\boldsymbol{y})$. This motivates us to explore one-stage Saab transform based video coding to improve the coding efficiency.

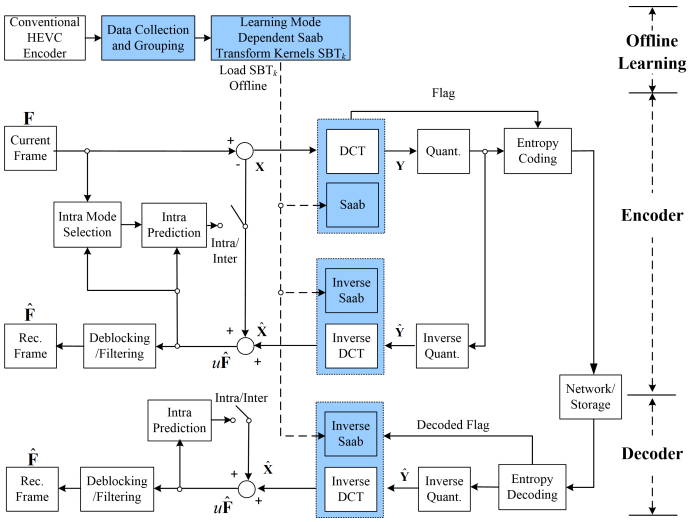


Fig. 3. Framework of the Saab transform based intra coding.

III. SAAB TRANSFORM BASED INTRA VIDEO CODING

In this section, Saab transform is applied to video codec to improve intra video coding efficiency. Firstly, we present the proposed framework for Saab transform based intra video coding. Then, intra mode dependent Saab transform kernels are developed. Finally, three strategies are proposed to integrate Saab transform to HEVC intra coding for higher efficiency.

A. Framework of Saab Transform based Intra Video Coding

Fig. 3 illustrates the proposed framework of Saab transform based intra coding, which includes offline learning for Saab transform kernels, Saab transform based encoder and decoder. In the figure, the white rectangles are the original coding modules in HEVC and the blue ones are the newly added modules for the proposed algorithm.

At the stage of learning Saab transform kernels, the intra prediction block residuals $\mathcal{D}$=$\{\boldsymbol{x}_{Train}\}$ are collected offline from conventional DCT based video encoder. Since the distribution of the residual data highly depends on the intra mode [16], all intra modes are divided into n mode sets, $\mathbf{M}_k$, $k \in [0, n]$ and $n <= 34$ for HEVC. Then, block residuals $\mathcal{D}$=$\{\boldsymbol{x}_{Train}\}$ are divided into groups $\{g_k\}$ regarding their intra modes are whether in $\mathbf{M}_k$ or not. A number of Saab transform kernels $\{\text{SBT}_0,...,\text{SBT}_n\}$ are learned individually based on the intra mode in $\mathbf{M}_k$ and their block residual groups $\{g_k\}$. Details

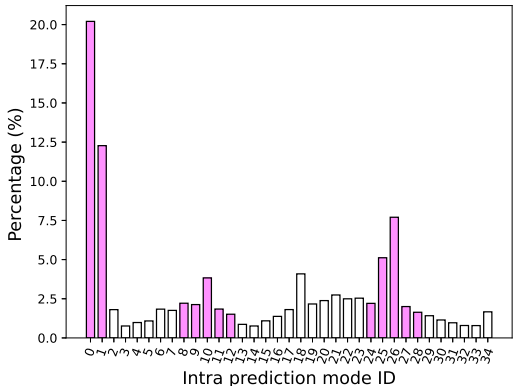


Fig. 4. Ratio of blocks encoded with the intra mode indexed by $0 \sim 34$.

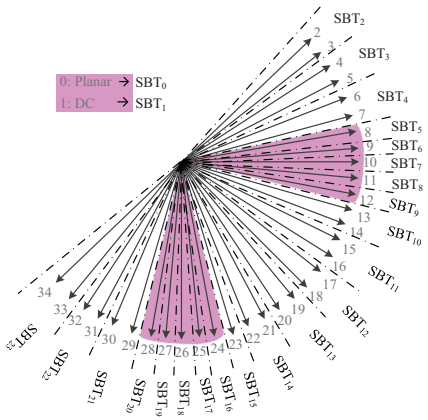


Fig. 5. Correspondence between intra prediction modes and Saab transform kernels SBT_k.

will be presented in Section III-B. Note that this is an offline training process that various video sequences and settings can be used to train the Saab transform kernels. The complexity of the Saab transform training is negligible. Also, the trained Saab transform kernels are transmitted offline and stored at client side for inverse transform.

At the encoder side, the learned Saab transform kernels $\{\text{SBT}_0, ...,\text{SBT}_n\}$ are utilized to transform block residuals based on the intra mode. For example, SBT_k will be used to transform the block residuals from mode in $\mathbf{M}_k$. There may be several cases to integrate Saab transform into the video encoder. One is to replace the DCT with the Saab transform. The other is to add the Saab transform as an alternative transform option and compete with the DCT using RD cost comparison. In the latter case, a flag indicating the choice of Saab transform and DCT will be encoded and transmitted to the client for decoding.

At the decoder side, if the conventional DCT is replaced with Saab transform, based on the intra mode in $\mathbf{M}_k$, the Saab transform kernel SBT_k will be used in the inverse Saab transform to reconstruct block residuals. Otherwise, based on the decoded flag and intra mode in $\mathbf{M}_k$, either DCT or SBT_k will be selectively used in the inverse transform. Note that Saab transform kernels SBT_k are loaded offline, thus, the coding bit is neglected.

B. Intra Mode Dependent Saab Transform

Saab transform is data-driven as it learned from the statistical characteristics of input source. However, the statistical characteristics of intra prediction block residuals $\{\boldsymbol{x}\}$ depends on intra prediction accuracy as well as image texture [37]. Since the angular intra prediction mode has a big impact on the prediction residuals in a block, it is necessary to learn Saab transform kernels based on the statistical characteristics of the angular intra mode. Thus we develop intra mode dependent Saab transform to maximize the coding performance.

Intra prediction block residuals shall be divided into groups g_k in terms of intra prediction modes $\mathbf{M}_k$, which are then used to learn Saab transform kernel SBT_k. The statistical characteristics of block residuals from single intra mode are relatively easier to be represented as compared with those of multiple intra modes. Thus, Saab transform learned from residuals of single intra mode, denoted as Fine-Grained Saab Learning (FGSL), may have better representation performance than that from multiple intra modes, noted as Coarse-Grained Saab Learning (CGSL). However, FGSL needs to train SBT_k for each mode, $0 \leq k \leq 34$. In other words, there are 35 SBTs for each TU size in HEVC, and the number is even larger for standards beyond HEVC. Thus, the FGSL drastically increases the difficulties in codec design. In addition, the ratio of blocks for each mode distributes unevenly, which may cause some difficulties in collecting sufficient data for training.

The distributions of 35 intra prediction modes were statistically analyzed on the number of 8×8 luminance block residuals encoded by each of these modes. 100 frames of each video sequence in {"BasketballDrill", "FourPeople", "RaceHorses"} were encoded with four QPs in {22, 27, 32, 37} in HEVC. As shown in Fig. 4, block residuals of "Planar", "DC", "Horizontal" and "Vertical" and their neighboring modes have higher percentages than those of rest modes, which indicates these modes have higher impacts on the coding efficiency than the others. Therefore, by considering the coding efficiency and design complexity, we train Saab transform kernels for "Planar (0)", "DC (1)", "Horizontal (10)" and "Vertical (26)" and their neighboring modes ($8 \sim 12$ and $24 \sim 28$) with FGSL scheme, shown as in Category B in Table II. Since neighboring intra angular modes have similar directions, two or more intra modes can be grouped for training. Saab transform kernels for the modes in Category A, as shown in Table II, are trained with CGSL scheme to reduce the number of kernels. 24 Saab transform kernels in total are trained with FGSL and CGSL schemes, and the correspondence between intra modes and $\text{SBT}_k, 0 \leq k \leq 23$, are illustrated in Fig. 5. The details of the FGSL and CGSL schemes are:

- *CGSL for modes in Category A*: Group residual blocks generated from intra prediction mode ID i and ID $i-1$ as one set and train Saab kernel SBT_k. Then, SBT_k will be applied to residual blocks from intra mode with ID $i-1$ and i. For example, SBT_2 is trained from residual blocks generated from intra mode 2 and 3.

TABLE II
TWO TRAINING STRATEGIES FOR MODE DEPENDENT SAAB TRANSFORM.

Category	Mode ID	Saab Learning Scheme
A	$2 \sim 7$, $13 \sim 23$, $29 \sim 34$	CGSL
B	$0 \sim 1$, $8 \sim 12$, $24 \sim 28$	FGSL

- *FGSL for modes in Category B*: Collect residual blocks generated from intra prediction mode ID i as one set and train Saab kernel SBT_k. Different from CGSL, the learned Saab transform SBT_k will be only applied to residual blocks from intra mode ID i. For example, SBT_5 is trained from residual blocks generated from intra mode 8. Note that SBT_0 and SBT_1 are learned from residual blocks of "Planar" and "DC" mode, respectively.

C. Integration Strategies for Saab Transform

Since Saab transform has good performance on energy compaction and decorrelation, as analyzed in Subsection II-C, we propose three integration strategies to apply intra mode dependent Saab transform to the HEVC codec. Table III shows these three integration strategies, noted as s_I, s_{II} and s_{III}. In s_I, each intra prediction mode adopts either Saab transform or DCT in transform coding. Intra modes ID $i \in \{0 \sim 7, 13 \sim 23, 29 \sim 34\}$ utilize SBT_k with index $k \in \{0 \sim 4, 10 \sim 15, 21 \sim 23\}$ as their transforms. For intra modes around the horizontal and vertical directions, i.e. intra modes ID $i \in \{8 \sim 12, 24 \sim 28\}$, the original DCT is utilized since the Saab kernel SBT_k is not always superior for these modes. Detailed analysis can be referred to Section IV-B. To complement Saab transform with DCT, strategy s_{II} in the middle row of Table III is proposed. In s_{II}, SBT_k is applied to intra modes in $\{0 \sim 7, 13 \sim 23, 29 \sim 34\}$. Meanwhile, the DCT is also activated. The optimal transform is selected from DCT and SBT_k by choosing the lower RD cost. N/A indicates there is no available SBT_k for modes in $\{8 \sim 12, 24 \sim 28\}$ and DCT is directly used for them. Furthermore, strategy s_{III} is proposed to maximize the coding efficiency, as shown in the bottom row in Table III. SBT_k, $0 \leq k \leq 23$, are applied to all 35 intra prediction modes, while DCT is also activated. The optimal transform is selected from SBT_k and DCT based on RD comparison. In strategies s_{II} and s_{III}, the optimal transform is selected with RDO from SBT_k and DCT. An 1 bit flag shall be added in the bitstream for each TU, where 0/1 indicates using DCT/SBT_k, respectively. Note that k in SBT_k is determined by the intra mode according to the learning scheme. The RD performance gain of Saab transform and three integration strategies in intra video coding are analyzed in following sections.

IV. RD PERFORMANCE AND COMPUTATIONAL COMPLEXITY OF SAAB TRANSFORM BASED INTRA CODING

In this section, RD performance and computational complexity of Saab transform based intra video coding are analyzed theoretically and experimentally. Firstly, we analyze the RD cost of Saab transform based intra video coding and two sufficient conditions are derived while using Saab transform to improve the RD performance. Then, these two sufficient conditions are validated individually with coding experiments. Finally, computational complexity of one-stage Saab transform is analyzed.

A. Theoretical RD Cost Analysis on Saab Transform

The objective of video coding is to minimize the distortion (D) subject to a given bit rate (R). By introducing the Lagrangian multiplier λ, the coding optimization can be formulated by minimizing RD cost $J(Q)$ as [1]

$$\min J(Q), J(Q) = D(Q) + \lambda \cdot R(Q), \tag{8}$$

where Q is quantization step, $D(Q)$ and $R(Q)$ are distortion and bit rate at given Q. So, it is necessary to analyze $D(Q)$ and $R(Q)$ of Saab transform based intra coding and compared with those of DCT based intra coding to validate its effectiveness.

The rate $R(Q)$ and distortion $D(Q)$ of using the Saab transform are modelled and theoretically analyzed. The bit rate R can be modelled with the entropy of the transformed coefficient y. Meanwhile, when the transformed coefficient y is quantized with quantization step Q, the bit rate $R(Q)$ can be modelled as its entropy minus $\log Q$, which is [38]

$$R(Q) \approx -\int_{-\infty}^{+\infty} f_y(y) \log f_y(y) dy - \log Q. \tag{9}$$

To analyze the transformed coefficient y output from Saab transform, we collected 1000 of 8×8 luminance block residuals generated by "Planar" mode from encoding "FourPeople" in HEVC. Histograms of transformed coefficients from Saab at locations (0,2) and (5,2), are presented in Fig. 6. We can observe that distributions of transformed coefficients from Saab generally conform Laplacian and Gaussian distributions, which are denoted as $y_{Saab} \sim \text{Laplace}(\mu_{y_{Saab}}, \sigma_{y_{Saab}})$ and $y_{Saab} \sim \text{N}(\mu_{y_{Saab}}, \sigma_{y_{Saab}})$. Since the Laplacian model achieves a higher accuracy than the Gaussian model, Laplacian distribution is finally used in modelling the Saab transformed coefficient y, i.e., $f_y(y) = \sqrt{2}\sigma_y e^{-\frac{\sqrt{2}}{\sigma_y}|y|}$. By applying $f_y(y)$ to Eq. 9, we can obtain the rate $R(Q)$ as

$$R(Q) \approx \log \frac{\sqrt{2}e\sigma_y}{Q} \quad . \tag{10}$$

Since uniform quantizer is used to quantize the transformed coefficient y, the range of y will be partitioned into an infinite number of intervals $I_t = (q_t, q_{t+1})$. y in the interval I_t will be mapped to s_t after quantization. As s_t is independent with t, given the quantization step size Q, the distortion caused by quantization $D(Q)$ can be calculated as [39]

$$D(Q) = \sum_{-\infty}^{+\infty} \int_{s_t - 0.5Q}^{s_t + 0.5Q} (y - s_t)^2 f_y(y) dy, \tag{11}$$

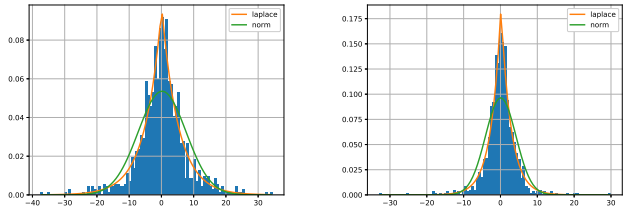
where Q is the quantization step size. As the distribution of the transformed coefficient y after uniform quantization still obey Laplacian distribution, i.e., $f_y(y) = \sqrt{2}\sigma_y e^{-\frac{\sqrt{2}}{\sigma_y}|y|}$, the distortion $D(Q)$ can be approximated as [40]

$$D(Q) \approx \sigma_y^2 \frac{Q^2}{12\sigma_y^2 + Q^2} \quad . \tag{12}$$

TABLE III
INTRA MODE DEPENDENT SAAB TRANSFORM SET $\{\text{SBT}_k\}$ AND INTEGRATION STRATEGIES.

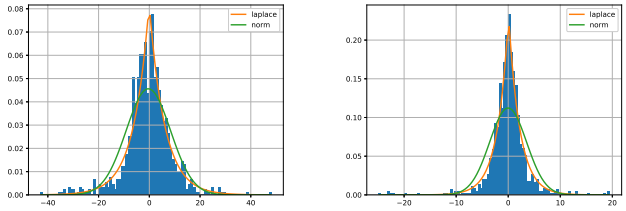
Integration Strategies	Transform	Intra Mode ID																																	
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
s_I [1]	SBT / DCT	0	1	2	2	3	3	4	4	DCT					10	10	11	11	12	12	13	13	14	14	15	DCT					21	21	22	22	23	23
s_{II} [2]	SBT index	0	1	2	2	3	3	4	4	N/A					10	10	11	11	12	12	13	13	14	14	15	N/A					21	21	22	22	23	23
	DCT	DCT																																	
s_{III} [2]	SBT index	0	1	2	2	3	3	4	4	5	6	7	8	9	10	10	11	11	12	12	13	13	14	14	15	16	17	18	19	20	21	21	22	22	23	23
	DCT	DCT																																	

[1] Either DCT or SBT_k is used depending on the intra mode.
[2] The optimal transform is selected from DCT and SBT_k with RDO. One bit signalling flag is transmitted to indicate the type, where 0 / 1 indicate DCT / SBT_k.



(a) (b)

Fig. 6. Distributions of transformed coefficients via Saab transform for 8×8 block residuals generated by "Planar" mode. (a) Transformed coefficient at location (0,2) from Saab, (b) Transformed coefficient at location (5,2) from Saab.



(a) (b)

Fig. 7. Distributions of DCT transformed coefficients for 8×8 residual blocks generated by "Planar" mode. (a) Transformed coefficient at location (0,2) from DCT, (b) Transformed coefficient at location (5,2) from DCT.

Therefore, the RD cost of the Saab transform based intra coding scheme can be calculated as

$$J \approx \kappa_y = \begin{cases} \sigma_y^2 \frac{Q^2}{12\sigma_y^2+Q^2} + \lambda \cdot \log \frac{\sqrt{2}e\sigma_y}{Q} & \sigma_y > \frac{Q}{\sqrt{2}e} \\ \sigma_y^2 \frac{Q^2}{12\sigma_y^2+Q^2} & \sigma_y \leq \frac{Q}{\sqrt{2}e} \end{cases}, \tag{13}$$

where the right part is defined as κ_y for further illustration. When $\sigma_y \leq \frac{Q}{\sqrt{2}e}$, $R(Q) = 0$, so $J \approx \kappa_y = \sigma_y^2 \frac{Q^2}{12\sigma_y^2+Q^2}$.

Similarly, we also analyze distributions of the transformed coefficients from DCT, as shown in Fig. 7. These distributions of transform coefficients of DCT are closer to Laplacian distribution $y_{DCT} \sim \text{Laplace}(\mu_{y_{DCT}}, \sigma_{y_{DCT}})$ than Gaussian distribution $y_{DCT} \sim \text{N}(\mu_{y_{DCT}}, \sigma_{y_{DCT}})$. Therefore, Eq.13 can also be derived for DCT based intra coding. To differentiate Saab transform from DCT, transformed coefficients of Saab are noted as y_{Saab} and those of DCT are noted as y_{DCT}. κ_y for Saab and DCT are denoted as $\kappa_{y_{Saab}}$ and $\kappa_{y_{DCT}}$. Therefore, for block residuals, RD gain can be achieved if the transformed coefficient of Saab transform satisfies condition

$$\kappa_{y_{Saab}} < \kappa_{y_{DCT}} \ . \tag{14}$$

Apply Eq.13 to Eq.14, we can obtain an inequality that relates to $\sigma^2_{y_{Saab}}$, $\sigma^2_{y_{DCT}}$ and quantization steps Q. For simplicity, we find Eq. 14 is satisfied by all quantization step Q when the variances of the transformed coefficients, $\sigma^2_{y_{Saab}}$ and $\sigma^2_{y_{DCT}}$, satisfy condition

$$\sigma^2_{y_{Saab}} < \sigma^2_{y_{DCT}}. \tag{15}$$

This inequality is a sufficient but not necessary condition for Eq.14, which is more critical. It means transform that minimizes the output variances of transformed coefficients will improve the RD performance of a codec. Both conditions in Eq.14 and Eq.15 will be experimentally analyzed in detail in the following subsection so as to testify the effectiveness of Saab transform.

B. Experimental RD Cost Analysis on Saab Transform

Two conditions in Eq.14 and Eq.15 were analyzed by comparing $\sigma^2_{y_{Saab}}$ and $\sigma^2_{y_{DCT}}$, $\kappa_{y_{Saab}}$ and $\kappa_{y_{DCT}}$ of transformed coefficients from Saab and DCT, respectively.

In the experiment, Saab training configurations were generally the same as those in Section II-C. Saab transform kernel SBT_1 was learned for intra mode "Planar". Then, the learned kernel SBT_1 was applied to transform "Planar" mode blocks. Three sequences with different resolutions, i.e. "PeopleOnStreet" at 2560×1600, "Johnny" at 1280×720 and "RaceHorses" at 416×240 , were tested when QP $\in \{22, 27, 32, 37\}$. Thousands of "Planar" mode residuals were randomly collected after intra prediction, which were transformed with SBT_1 and DCT to compute $\kappa_{y_{Saab}}$ and $\kappa_{y_{DCT}}$. To quantitatively analyze the difference between $\kappa_{y_{Saab}}$ and $\kappa_{y_{DCT}}$, $\Delta\kappa$ is defined as

$$\Delta\kappa = \kappa_{y_{Saab}} - \kappa_{y_{DCT}}, \tag{16}$$

where negative $\Delta\kappa$ indicates a better RD performance of using Saab transform as compared with DCT, while positive $\Delta\kappa$ indicates a worse RD performance. Table IV shows $\kappa_{y_{Saab}}$, $\kappa_{y_{DCT}}$ and $\Delta\kappa$ for different QPs and video sequences. We can observe that $\kappa_{y_{Saab}}$ is generally smaller than $\kappa_{y_{DCT}}$, and the average $\Delta\kappa$ are 0.0001, -0.0001, -0.0010 and -0.0045 when QP is 22, 27, 32 and 37, respectively. It means for the "Planar" mode the Saab transform can achieve better RD performance

TABLE IV
COMPARISONS BETWEEN $\kappa_{y_{Saab}}$ AND $\kappa_{y_{DCT}}$ FOR 8×8 LUMINANCE (Y) BLOCK RESIDUALS FROM "PLANAR" MODE.

QP	22			27			32			37		
Sequence	$\kappa_{y_{DCT}}$	$\kappa_{y_{Saab}}$	$\Delta\kappa$	$\kappa_{y_{DCT}}$	$\kappa_{y_{Saab}}$	$\Delta\kappa$	$\kappa_{y_{DCT}}$	κ_{Saab}	$\Delta\kappa$	$\kappa_{y_{DCT}}$	$\kappa_{y_{Saab}}$	$\Delta\kappa$
PeopleOnStreet	0.1035	0.1036	0.0001	0.6765	0.6764	-0.0001	4.3067	4.3052	-0.0015	21.6978	21.6912	-0.0066
RaceHorses	0.1263	0.1264	0.0001	0.9002	0.9001	-0.0001	5.3745	5.3743	-0.0002	22.6707	22.6658	-0.0049
Johnny	0.0751	0.0751	0.0000	0.2819	0.2818	-0.0001	1.3996	1.3982	-0.0014	11.1232	11.1211	-0.0021
Average	-	-	0.0001	-	-	-0.0001	-	-	-0.0010	-	-	-0.0045

TABLE V
COMPARISONS BETWEEN $\sigma^2_{y_{Saab}}$ AND $\sigma^2_{y_{DCT}}$ FOR 8×8 BLOCK RESIDUALS WHEN QP IS 37. FOUR INTRA PREDICTION MODES "PLANAR", "DC", "HORIZONTAL" AND "VERTICAL" ARE TESTED.

Intra mode	Planar			DC			Horizontal			Vertical		
Sequence	$\sigma^2_{y_{DCT}}$	$\sigma^2_{y_{Saab}}$	$\Delta\sigma^2$	$\sigma^2_{y_{DCT}}$	$\sigma^2_{y_{DCT}}$	$\Delta\sigma^2$	$\sigma^2_{y_{DCT}}$	$\sigma^2_{y_{Saab}}$	$\Delta\sigma^2$	$\sigma^2_{y_{DCT}}$	$\sigma^2_{y_{Saab}}$	$\Delta\sigma^2$
PeopleOnStreet	24.899	24.891	-0.008	32.741	32.735	-0.006	32.419	31.416	-1.003	21.673	21.673	0.000
RaceHorses	35.890	35.884	-0.006	54.590	54.589	-0.001	74.434	74.443	0.009	53.016	53.031	0.015
Johnny	11.908	11.906	-0.002	18.887	18.797	-0.09	30.122	30.119	-0.003	11.460	11.461	0.001
Average	-	-	-0.005	-	-	-0.032	-	-	-0.332	-	-	0.005

on average when QP are 27, 32 and 37 and a little worse than DCT on RD performance when QP is 22. So, Saab transform is actually more effective than DCT on average.

In addition, $\sigma^2_{y_{Saab}}$ and $\sigma^2_{y_{DCT}}$ were also analyzed and compared to validate the effectiveness of Saab transform. Four Saab transforms were learned from 80K 8×8 luminance block residuals for intra prediction modes in {"Planar", "DC", "Horizontal", "Vertical"}, respectively. Then, these Saab transforms were applied to block residuals of {"Planar", "DC", "Horizontal", "Vertical"} correspondingly. As a comparison, the same set of block residuals were also transformed by DCT. Then, $\sigma^2_{y_{Saab}}$ and $\sigma^2_{y_{DCT}}$ were computed from the transformed coefficients of Saab transform and DCT. Four intra modes {"Planar", "DC", "Horizontal", "Vertical"} and three video sequences {"PeopleOnStreet", "RaceHorses", "Johnny"} were tested. QP was fixed as 37. The difference $\Delta\sigma^2$ between $\sigma^2_{y_{Saab}}$ and $\sigma^2_{y_{DCT}}$ is defined as

$$\Delta\sigma^2 = \sigma^2_{y_{Saab}} - \sigma^2_{y_{DCT}}, \tag{17}$$

where negative $\Delta\sigma^2$ indicates a better RD performance of using Saab transform and positive $\Delta\sigma^2$ indicates a worse RD performance of using Saab transform as compared with DCT. $\sigma^2_{y_{Saab}}$ and $\sigma^2_{y_{DCT}}$ are variances of transformed coefficients from Saab transform and DCT, respectively. Table V shows $\sigma^2_{y_{Saab}}$, $\sigma^2_{y_{DCT}}$ and $\Delta\sigma^2$ for four different intra modes. We can observe that the average $\Delta\sigma^2$ of intra prediction modes "Planar", "DC", "Horizontal" and "Vertical" are -0.005, -0.032, -0.332 and 0.005, respectively. It indicates the Saab transform performed better than DCT for intra prediction modes "Planar", "DC" and "Horizontal" on average, but a little inferior to DCT for "Vertical" mode on average. In fact, other intra modes were compared and Saab transform performed better than DCT for most modes and sequences. Therefore, Saab transform is able to improve the video coding efficiency.

Overall, Saab transform has better performance than DCT for different sequences, QPs, and intra modes on average, which can be used to replace DCT to improve the coding efficiency. However, Saab transform is inferior to DCT in some cases, such as the cases when QP is 22 or intra prediction mode is "Vertical". Therefore, to maximize the coding efficiency, an alternative way is to combine Saab transform with DCT and select the optimal one with RD cost comparison.

C. Computational Complexity Analysis for Saab Transform

We measure the transform complexity via the number of float-point multiplications or divisions. Practical complexity is desired to be explored in the future. So, the computational complexity of applying DCT to the block of size $M \times N$ is $O(2MN^2 + 2M^2N)$. Saab transform for blocks of size $M \times N$ is a little different from DCT at the computational complexity. It requires an extra $3MN$ float-point computations in one-stage Saab transform before mapping one block of size $M \times N$ to one DC coefficient and $M \times N - 1$ AC coefficients. Therefore, the computational complexity of one-stage Saab transform is $O(3MN + 2(MN)^2)$. Theoretically, the complexity of DCT is relatively lower than the one-stage Saab transform. The ICT is low complexity approximation of DCT, which is implemented with integer arithmetic to avoid the float-point multiplication. Thus, the complexity of ICT in HEVC is much lower than that of DCT as well as one-stage Saab transform computed in float-point arithmetic.

V. EXPERIMENTAL RESULTS AND ANALYSIS

We evaluate the RD performance of Saab transform in comparison with DCT for intra video coding in HEVC. In learning Saab transform, 24 of the Saab transform kernels, noted as $\text{SBT}_k, 0 \leq k \leq 23$, were learned offline from around 80K block residuals separately. These 80K block residuals were collected from encoding the frames of four video sequences, i.e. "PeopleOnStreet" at 2560×1600, "BasketballDrill" at 832×480, "BasketballPass" at 416×240 and "FourPeople" at 1280×720, with $QP \in \{22, 27, 32, 37\}$. The Saab transform based intra video codecs were implemented on HEVC test model version 16.9 (HM16.9) and Saab transform was implemented with C++. The coding experiments were performed under All Intra (AI) configuration, where four QPs $\in \{22, 27, 32, 37\}$ were tested. Note that video sequences of Class B were clipped from 1920×1080 to 1920×1072

TABLE VI
BDBR OF SAAB TRANSFORM BASED INTRA CODING WHERE DCT IS REPLACED WITH SBT_k INDIVIDUALLY FOR EACH INTRA MODE.

Transform set with one SBT		s_0	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}	s_{21}	s_{22}	s_{23}
SBT index		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Intra prediction modes		0	1	2,3	4,5	6,7	8	9	10	11	12	13,14	15,16	17,18	19,20	21,22	23	24	25	26	27	28	29,30	31,32	33,34
Class	Sequence	BDBR (%)																							
A	Traffic	-0.21	-0.06	-0.25	-0.61	-0.41	-0.14	0.04	0.02	0.17	0.05	-0.01	0.00	0.03	-0.04	-0.03	0.02	0.04	0.10	0.11	-0.01	0.03	0.05	-0.07	-0.15
	PeopleOnStreet *	-0.42	-0.14	-0.12	-0.29	-0.28	-0.08	0.14	0.08	0.09	0.04	0.03	-0.24	0.04	0.01	-0.08	0.16	0.14	0.06	0.06	0.10	-0.05	0.12	-0.02	-0.20
C	BQMall	-0.19	-0.08	-0.28	-0.04	-0.06	-0.14	0.15	0.02	0.13	-0.09	-0.02	-0.18	-0.26	-0.34	-0.15	-0.11	-0.05	0.23	1.02	0.09	-0.01	-0.02	-0.23	-0.19
	BaketballDrill *	-0.13	0.08	-0.50	-0.51	-0.69	-0.37	-0.17	-0.22	-0.07	0.03	-0.42	-0.65	-3.81	-2.34	-1.65	-0.50	-0.43	-0.34	-0.28	-0.21	-0.43	-0.18	-0.05	0.11
D	BQSquare	-0.32	-0.23	-0.20	-0.26	-0.06	-0.15	-0.19	-0.12	-0.26	-0.08	-0.11	-0.23	-0.19	-0.21	-0.12	-0.22	0.01	-0.04	0.07	0.04	-0.14	-0.16	-0.12	-0.11
	BaketballPass *	-0.21	-0.16	-0.04	-0.20	-0.17	0.15	0.00	0.00	0.15	0.35	-0.06	-0.03	-0.50	-0.10	-0.45	-0.13	-0.25	-0.04	0.21	-0.22	-0.07	-0.25	-0.24	-0.11
E	KristenAndSara	0.34	0.08	-0.05	-0.24	-0.01	0.08	0.06	0.07	0.22	-0.06	0.12	-0.38	0.10	-0.23	-0.21	-0.06	0.20	0.25	0.66	0.72	-0.01	-0.12	0.04	0.02
	FourPeople *	-0.16	0.02	-0.17	-0.25	0.00	0.00	-0.09	0.15	0.16	-0.09	0.07	-0.23	-0.23	-0.18	-0.35	-0.06	-0.10	0.47	-0.04	0.20	-0.01	-0.02	-0.23	-0.25
Average		-0.16	-0.06	-0.20	-0.30	-0.21	-0.08	-0.01	0.00	0.07	0.02	-0.05	-0.24	-0.60	-0.43	-0.38	-0.11	-0.05	0.09	0.23	0.09	-0.09	-0.07	-0.11	-0.11

* Partial blocks of these video sequences were utilized to learn the Saab transforms.

and video sequences in Class A were encoded and decoded conforming to the main profile at level 4 for alignment.

All experiments were carried out in a workstation with 3.3GHz CPU and 96.0GB memory, Windows 10 operating system. Peak Signal-to-Noise Ratio (PSNR) and bit rate were utilized to evaluate the video quality and bit rate of the proposed Saab transform based intra video coding while Bjønteggard Delta PSNR (BDPSNR), Bjønteggard Delta Bit Rate (BDBR) [41] were adopted to represent coding gain.

A. Coding Efficiency Analysis

We evaluated the coding performance of Saab transform based intra video coding in three phases. In the first and second phases, CU size was fixed as size of 16×16 and TU size was fixed as 8×8 for both proposed schemes and benchmark HEVC so as to analyze the performance of 8×8 Saab transform as compared with 8×8 DCT. Firstly, the coding performance of each Saab transform kernel was validated one-by-one. In this experiment, DCT was replaced by the SBT_k for only one intra mode and the rest intra modes remained DCT, which had 24 combinations and denoted as s_k, $k \in [0, 23]$. Eight sequences were encoded for each s_k. Table VI shows the coding performance for proposed Saab transform based intra video codecs corresponding to each s_k as compared with the original DCT based codec. A negative BDBR value indicates coding gain and a positive value means coding loss. We have three observations: 1) BDBR from -0.01% to -0.60% can be achieved on average for most intra modes. 2) SBT_{12} can achieve the most significant BDBR gain, which is -0.60% on average, when it is applied to block residuals generated by intra mode 17 and 18. 3) BDBR values are positive for intra modes around horizontal and vertical directions, where mode ID $i \in \{10, 11, 12, 25, 26, 27\}$. It indicates that the RD performance of SBT_k is inferior to the DCT on average for these modes. Based on these results, if without RDO competition between DCT and Saab, we propose not to replace DCT with SBT_k for intra modes in $\{8 \sim 12, 24 \sim 28\}$, i.e., integration strategy s_I. If with RDO competition, s_{II} and s_{III} in Table III are proposed.

Secondly, in addition to evaluate each SBT_k individually, the joint RD performance of using all SBT_k were also evaluated in Saab transform based intra coding, which had three strategies s_I, s_{II} and s_{III}. Twenty-three video sequences with various contents and resolutions in {416×240, 832×480, 1280×720, 1920×1080, 2560×1600} were encoded with the proposed Saab transform based intra video codec and the benchmark HEVC in the coding experiment. RDO Quantization (RDOQ) was disabled. 100 frames were encoded for each test sequence. Table VII shows the RD performance of the Saab transform based intra video codec as compared with the state-of-the-art DCT based HEVC codec. We can observe that three schemes s_I, s_{II} and s_{III} can achieve BDBR gain -1.41%, -2.59% and -3.07% on average. Scheme s_I can improve the coding efficiency for most sequences while schemes s_{II} and s_{III} can improve the BDBR for all test sequences. In addition, maximum BDBR gains are up to -9.10%, -9.72% and -10.00% for schemes s_I, s_{II} and s_{III}, respectively, which is significant and promising. The competition between Saab transform and DCT with RDO improves the coding performance of replacing DCT with Saab transforms, i.e., s_I, with a large margin.

Thirdly, we also performed the coding experiments when fully flexible CU/PU/TU size selection and RDOQ were enabled based on the common test conditions. The proposed Saab transform was applied to 8×8 TU with RDO comparison, i.e. s_{III}, and transform for the rest TUs were the same as the original HEVC. The original HEVC and CNN-based nonlinear transform [26], denoted as Yang's scheme, were used for comparison. Note that in Yang's scheme [26], the CNN based transform was only applied to 8×8 TU in intra coding, which is the same as the proposed scheme. Table VIII shows RD performances of the proposed s_{III} and Yang's scheme [26] as they were compared with the original HEVC in intra coding. We can observe that Yang's scheme is able to achieve BDBR from 0.01 % to -1.79%, and -0.75% on average, for the test sequences from Class A to Class E. For the proposed s_{III}, it achieves BDBR from -0.40 % to -6.09%, and -1.78% on average, which outperforms that of Yang's scheme. While including screen content sequences from Class F, it achieves -1.80% BDBR on average. It demonstrates that the proposed Saab transform has better decorrelation ability for transform residual as compared with DCT. Also, the proposed Saab transform is beter than CNN based transform due to more fine-grained prediction mode dependent design.

TABLE VII
RD PERFORMANCES AND COMPUTATIONAL COMPLEXITY OF SAAB TRANSFORM BASED INTRA VIDEO CODEC AS COMPARED WITH THE HEVC CODEC.

Transform set		s_I				s_{II}				s_{III}			
Class	Sequence	BDBR (%)	BD PSNR (dB)	ΔT_{Enc} (%)	ΔT_{Dec} (%)	BDBR (%)	BD PSNR (dB)	ΔT_{Enc} (%)	ΔT_{Dec} (%)	BDBR (%)	BD PSNR (dB)	ΔT_{Enc} (%)	ΔT_{Dec} (%)
A	NebutaFestival	0.25	-0.019	212.9	159.3	-2.13	0.154	235.9	136.1	-2.30	0.167	267.4	139.3
	StreamLocomotive	0.87	-0.045	224.3	153.8	-1.40	0.074	254.3	131.4	-1.69	0.089	290.8	134.0
	Traffic	-0.88	0.047	216.2	143.4	-2.06	0.112	236.2	125.1	-2.81	0.154	295.4	137.0
	PeopleOnStreet *	-1.07	0.061	225.2	148.7	-2.37	0.137	260.2	127.9	-3.00	0.174	318.8	137.6
B	Kimono	0.74	-0.026	213.9	158.2	-0.97	0.033	251.0	141.8	-1.19	0.040	291.1	136.4
	ParkScene	-0.11	0.05	198.1	144.8	-1.74	0.080	236.9	125.8	-2.07	0.096	285.9	133.7
	Cactus	-0.94	0.036	212.6	156.9	-2.28	0.089	232.4	126.0	-2.91	0.115	296.1	139.5
	BQTerrace	-0.37	0.008	203.7	136.8	-1.76	0.105	230.4	124.3	-2.32	0.136	290.4	133.0
	BasketballDrive	-0.62	0.012	193.2	132.2	-1.60	0.046	223.3	126.5	-2.24	0.065	287.2	125.8
C	RaceHorses	-0.91	0.060	217.1	161.4	-2.34	0.158	249.8	137.8	-2.67	0.180	296.7	142.7
	PartyScene	-1.18	0.091	198.3	145.9	-1.99	0.157	238.7	133.0	-2.69	0.214	296.9	144.0
	BQMall	-0.19	0.012	195.0	139.2	-1.38	0.083	231.6	125.0	-2.03	0.124	297.8	134.1
	BaketballDrill *	-9.10	0.463	212.6	155.4	-9.72	0.498	265.0	157.4	-10.00	0.514	273.3	150.5
D	RaceHorses	-2.37	0.158	199.9	169.7	-3.45	0.233	268.0	160.4	-3.87	0.262	303.2	161.5
	BlowingBubbles	-2.19	0.132	195.5	197.5	-3.12	0.190	265.5	157.2	-3.78	0.232	292.6	162.9
	BQSquare	-0.68	0.058	188.2	141.8	-1.87	0.171	243.6	152.9	-2.47	0.227	304.0	146.3
	BaketballPass *	-0.34	0.019	175.7	139.9	-1.41	0.084	223.0	155.5	-2.04	0.124	277.2	119.4
E	Johnny	-1.38	0.062	184.6	120.2	-2.20	0.101	226.4	120.2	-2.51	0.116	256.5	128.9
	KristenAndSara	-1.15	0.061	198.6	131.6	-1.89	0.103	218.6	117.5	-2.47	0.135	271.9	122.7
	FourPeople*	-0.99	0.058	199.6	135.4	-1.89	0.109	228.1	120.5	-2.58	0.149	269.4	122.6
F	BasketballDrillText	-7.45	0.413	223.1	162.6	-8.10	0.453	259.4	157.5	-8.39	0.470	275.4	151.2
	ChinaSpeed	-0.43	0.040	181.3	126.3	-1.14	0.106	227.8	118.7	-1.66	0.156	262.6	117.0
	SlideShow	-1.89	0.176	184.8	147.0	-2.70	0.262	213.0	134.4	-2.83	0.272	240.9	140.3
Average		**-1.41**	**0.082**	**202.4**	**147.6**	**-2.59**	**0.154**	**240.0**	**134.2**	**-3.07**	**0.183**	**284.4**	**138.4**

* Partially utilized in learning the Saab transforms.

TABLE VIII
RD AND COMPLEXITY OF THE PROPOSED SAAB BASED INTRA CODING AS FULLY FLEXIBLE CU/PU/TU SIZE SELECTION ENABLED,[UNIT:%].

Class	Sequence	Yang's scheme [26]	Proposed s_{III}		
		BDBR	BDBR	ΔT_{Enc}	ΔT_{Dec}
A	NebutaFestival	-0.37	-0.65	172.9	104.5
	StreamLocomotive	-0.76	-0.64	174.8	104.3
	Traffic	-1.37	-2.52	163.2	113.0
	PeopleOnStreet	0.01	-2.37	171.3	112.5
B	Kimono	-0.29	-0.57	170.1	101.3
	ParkScene	-1.79	-1.66	164.9	108.8
	Cactus	-1.14	-1.76	165.7	108.5
	BQTerrace	-0.75	-1.29	165.6	106.2
	BasketballDrive	-0.42	-1.65	164.2	106.2
C	RaceHorses	-1.08	-1.72	169.2	112.5
	PartyScene	-0.40	-1.10	170.1	112.2
	BQMall	-0.78	-1.17	169.2	107.8
	BaketballDrill	-0.61	-6.09	167.8	131.8
D	RaceHorses	-1.16	-2.42	172.7	132.5
	BlowingBubbles	-0.81	-2.26	162.7	134.2
	BQSquare	-0.57	-0.40	156.8	115.9
	BaketballPass	-0.48	-1.21	171.4	118.2
E	Johnny	-0.75	-1.74	162.8	101.6
	KristenAndSara	-0.73	-1.89	164.2	108.6
	FourPeople	-0.83	-2.49	166.9	112.2
Average		**-0.75**	**-1.78**	**167.3**	**112.6**
F	BasketballDrillText	-	-5.13	170.9	131.8
	ChinaSpeed	-	-0.52	167.8	102.0
	SlideShow	-	-0.17	159.7	102.6
Average of all seqs.		**-**	**-1.80**	**167.2**	**112.6**

B. Coding Complexity Analysis

In addition to the coding efficiency, the coding complexity of the proposed Saab transform based intra video coding was also analyzed. The precision of the Saab transform kernels is stored and computed with 20 decimal digits, and the storage space for these 24 8×8 one-stage Saab transform kernels for codec is about 3 MB. The ratios of the computational complexities of the proposed Saab transform based intra video encoder/decoder to those of the DCT based anchor encoder/decoder are defined as

$$\begin{cases} \Delta T_{Enc} = \frac{1}{4}\sum_{i=1}^{4} \frac{T_{Enc,Saab}(QP_i)}{T_{Enc,DCT}(QP_i)} \times 100\% \\ \Delta T_{Dec} = \frac{1}{4}\sum_{i=1}^{4} \frac{T_{Dec,Saab}(QP_i)}{T_{Dec,DCT}(QP_i)} \times 100\% \end{cases}, \quad (18)$$

where $T_{Enc,Saab}(QP_i)$ and $T_{Dec,Saab}(QP_i)$ are encoding and decoding time for QP_i in Saab transform based intra video codec, and $T_{Enc,DCT}(QP_i)$ and $T_{Dec,DCT}(QP_i)$ are encoding/decoding time for QP_i in DCT based intra video codec. Note that the DCT was implemented with ICT and butterfly operation in DCT based intra video codec. In Table VII, ΔT_{Enc} of intra video codecs with schemes s_I, s_{II} and s_{III} are 202.4%, 240.0% and 284.4% on average respectively. There are three reasons. Firstly, the computational complexity of Saab transform is a little higher than that of DCT, as illustrated in Section IV-C. Secondly, the DCT was optimized in implementation and Saab transform was not. In fact, the implementation of Saab transform can be optimized in future. Thirdly, the complexity of the strategies s_{II} and s_{III} further increases because the optimal transform was selected based on the RD competition between Saab and DCT.

In addition, the decoding complexity of Saab transform based video decoder was also evaluated. The ΔT_{Dec} of s_I, s_{II} and s_{III} are 147.6%, 134.2% and 138.4% on average respectively. Similarly, the complexity is mainly brought by the implementation of Saab transform. The decoding time of s_{II} and s_{III} are reduced as compared with s_I because partial blocks were decoded with DCT in s_{II} and s_{III}, which had lower computational complexity than Saab transform.

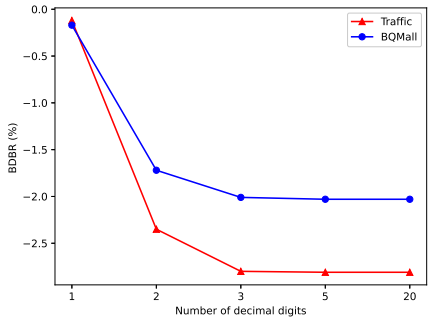


Fig. 8. BDBR of s_{III} with Saab transforms of different decimal digits.

TABLE IX
RATIO OF BLOCKS ENCODED WITH s_{III}.

Class	Sequence name	$P_{Saab}(QP_i)$ (%)				
		QP_i				Average
		22	27	32	37	
A	Traffic	43.72	43.06	37.93	28.66	**38.34**
	PeopleOnStreet	43.21	41.79	36.07	31.80	**38.22**
B	ParkScene	43.27	35.36	34.21	39.24	**38.02**
	Cactus	48.83	41.86	33.70	32.43	**39.20**
C	PartyScene	50.77	48.10	45.33	42.21	**46.60**
	BasketballDrill	79.20	83.81	86.70	85.89	**83.90**
D	RaceHorses	49.68	50.57	55.97	64.70	**55.23**
	BQSquare	40.38	45.51	43.06	34.68	**38.26**
E	KristenAndSara	33.73	19.79	15.26	11.21	**20.00**
	FourPeople	36.18	25.57	23.05	21.67	**26.61**
F	BasketballDrillText	78.77	74.52	76.55	90.34	**80.04**
Average		**49.60**	**46.36**	**44.35**	**43.89**	**46.05**

Moreover, Table VIII shows the coding complexities of the proposed s_{III} as compared with the original HEVC, where fully flexible CU/PU/TU size selection and RDOQ were enabled. We can observe that ΔT_{Enc} and ΔT_{Dec} of the proposed s_{III} schemes are 167.2% and 112.6% on average, respectively, which are lower than the coding complexities in Table VII. The main reason is that the encoding complexity of the original HEVC is significantly increased as flexible CU/PU/TU size selection was enabled. While at the decoder side, the decoding complexity of the s_{III} reduced since only partial CUs were decoded with Saab transfrom.

C. RD Impacts from Computational Precision

In this experiment, we evaluated the RD impacts from different precisions in performing Saab transform. Video sequences "Traffic" and "BQMall" were encoded by Scheme s_{III} with different precisions, i.e., decimal digit is set as 1, 2, 3, 5 and 20, respectively. Fig. 8 shows the BDBR of s_{III} while the Saab transform is computed with different decimal digits. We can observe that BDBR for "Traffic" and "BQMall" were converged from -2.35% and -1.72% to -2.80% and -2.01% as the number of decimal digits increases from 2 to 3. Increasing the decimal digits from 3 to 5 can achieve BDBR as -2.81% and -2.03% for "Traffic" and "BQMall", respectively. The BDBR gain becomes saturated when the precision is larger than 3 decimal digits. Accordingly, precision with 3 to 5 decimal digits is recommended for the one-stage Saab transform.

D. Ratio of Blocks using Saab Transform

We analyzed the ratio of 8×8 blocks using SBT_k as the optimal transform. This ratio is defined as

$$P_{Saab}(QP_i) = \frac{n_{Saab}(QP_i)}{n_{Total}(QP_i)} \times 100\%, \tag{19}$$

where $n_{Saab}(QP_i)$ is the number of 8×8 blocks using SBT_k as the optimal transform at QP_i. $n_{Total}(QP_i)$ is the total number of encoded 8×8 blocks at QP_i. Eleven different video sequences were encoded by the proposed s_{III} scheme. Four QPs were tested, i.e., $QP \in \{22, 27, 32, 37\}$. Table IX shows the ratio $P_{Saab}(QP_i)$ for different test sequences and QPs. We can observe that the ratio of blocks that select SBT_k as the optimal transform with RDO is 46.05% on average. It means the Saab transform is more effective than DCT for almost half amount of blocks. In addition, we can also observe that the average ratios reach 80.04% and 83.90%, respectively when encoding "BasketballDrill" and "BasketballDrillText". For the worst case, the ratio is from 11.21% to 33.73 %, and 20.00% on average, when encoding "KristenAndSara". Fig. 9 shows the blocks using Saab transform and DCT in the proposed s_{III} scheme while encoding the first frame of "BasketballDrillText" (the best case) and "KristenAndSara"(the worst case). We can observe that there are a large proportion of blocks selecting Saab transform (blocks in white) for "BasketballDrillText" while the ratio reduces for "KristenAndSara". It is because "KristenAndSara" has a large proportion of smooth regions, whose residuals from intra prediction are already sufficient small. Overall, the Saab transform is highly effective.

VI. CONCLUSIONS

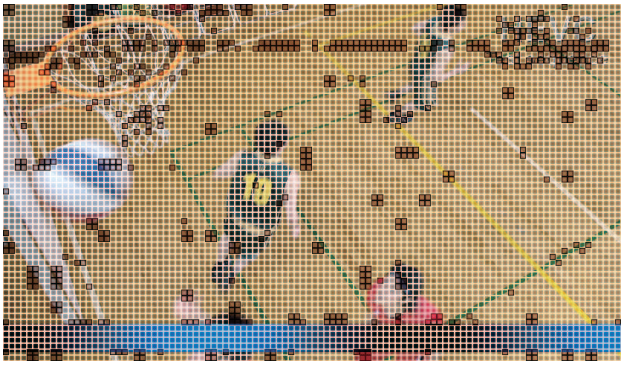
Learning based transform is able to capture the statistical characteristics of video contents, which is superior to the fixed Discrete Cosine Transform (DCT). In this work, learning based transform design is formulated as an optimization problem to maximize the energy compaction or decorrelation capability. Then, we propose a framework of Subspace approximation with adjusted bias (Saab) transform based intra coding, which includes intra mode dependent Saab transform learning and three integration strategies. And then, rate-distortion performance of one-stage Saab transform over DCT for intra video coding is theoretically analyzed and experimentally verified. Finally, extensive experiments shows that the proposed Saab transform based intra coding can significantly improve the coding efficiency, which proves the effectiveness and its applicability to advanced standards.

REFERENCES

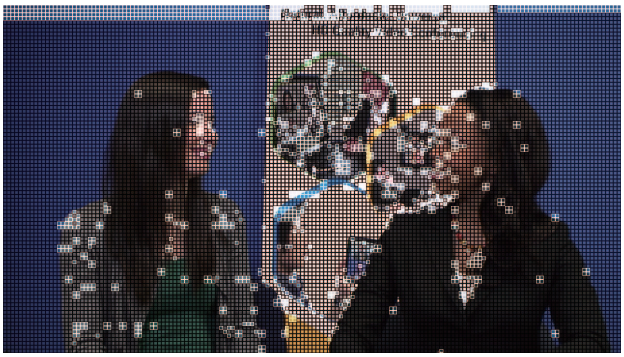
[1] B. Huang, Z. Chen, K. Su, J. Chen and N. Ling, "Low-complexity rate-distortion optimization for HEVC encoders," in *IEEE Trans. Broadcast.*, pp. 1-15, May. 2021, doi: 10.1109/TBC.2021.3077771.

[2] B. Bross, J. Chen, J. -R. Ohm, G. J. Sullivan and Y. -K. Wang, "Developments in international video coding standardization after AVC, with an overview of versatile video coding (VVC)," *Proc. IEEE*, 2021, in press, doi: 10.1109/JPROC.2020.3043399.

[3] Y. Zhang, S. Kwong, S. Wang, ""Machine learning based video coding optimizations: A survey", *Inf. Sci.*, vol. 506, pp. 395-423, 2020.



(a)



(b)

Fig. 9. The blocks using Saab transform and DCT in scheme s_{III} with QP 37, where white rectangles use Saab and black ones use DCT. (a) "BasketballDrillText"(the best case). and (b)"KristenAndSara"(the worst case).

[4] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, T. Wiegand, "Transform coding techniques in HEVC," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 978-989, Dec. 2013,

[5] I. Dvir, D. Irony, D. Drezner, A. Ecker, A. Allouche and N. Peterfreund, "Orthogonal directional transforms using discrete directional Laplacian eigen solutions for beyond HEVC intra coding," *IEEE Trans. Image Process.*, vol. 29, pp. 5136-5146, 2020.

[6] C. Lan, J. Xu, W. Zeng, G. Shi and F. Wu, "Variable block-sized signal-dependent transform for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1920-1933, Aug. 2018.

[7] M. Koo, M. Salehifar, J. Lim and S. Kim, "CE6: reduced secondary transform (RST) (CE6-3.1)," *JVET of VCEG and MPEG*, doc. JVET-N0193, Geneva, Mar. 2019.

[8] X. Cai and J.S. Lim, "Transforms for intra prediction residuals based on prediction inaccuracy modeling," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5505-5515, Dec. 2015.

[9] M. Wang, W. Xie, J. Xiong, D. Wang and J. Qin, "Joint optimization of transform and quantization for high efficiency video coding," *IEEE Access*, vol. 7, pp. 62534-62544, 2019.

[10] X. Zhang, C. Yang, X. Li, S. Liu, H. Yang, I. Katsavounidis, S.-M. Lei and C.-C. J. Kuo, "Image coding with data-driven transforms: methodology, performance and potential," *IEEE Trans. Image Process.*, vol. 29, pp. 9292-9304, 2020.

[11] H. E. Egilmez, Y.-H. Chao and A. Ortega, "Graph-based transforms for video coding," *IEEE Trans. Image Process.,* vol. 29, pp. 9330-9344, 2020.

[12] A. Arrufat, P. Philippe and O. Deforges,"Non-separable mode dependent transforms for intra coding in HEVC," in *Proc. IEEE Visual Commun. Image Process.*, pp. 61-64, Dec. 2014.

[13] S. Takamura and A. Shimizu, "On intra coding using mode dependent 2D-KLT," in *Proc. Pic. Coding Symp.*, pp. 137-140, 2013.

[14] X. Zhao, L. Zhang, S. Ma and W. Gao, "Video coding with rate-distortion optimized transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, pp. 138-151, Jan. 2012.

[15] J. Han, A. Saxena, V. Melkote and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1874-1884, April 2012.

[16] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li and W. Chien, "Enhanced multiple transform for video coding," in *Proc. Data Compress. Conf.*, pp. 73-82, 2016.

[17] A. Kammoun, W. Hamidouche, F. Belghith, J. Nezan and N. Masmoudi, "Hardware design and implementation of adaptive multiple transforms for the versatile video coding standard," *IEEE Trans. Consum. Electron.*, vol. 64, no. 4, pp. 424-432, Nov. 2018.

[18] X. Zhao, X. Li and S. Liu, "CE6: on 8-bit primary transform core (Test 6.1.3)," *JVET of ITU-T and ISO/IEC*, document JVET-L0285, Macao, CN, Oct. 2018.

[19] B. Zeng and J. Fu, "Directional discrete cosine transforms-a new framework for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 305-313, Mar. 2008.

[20] X. Zhao, J. Chen, M. Karczewicz, A. Said and V. Seregin, "Joint separable and non-separable transforms for next-generation video coding," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2514-2525, 2018.

[21] Y. Zhang, K. Zhang, L. Zhang, H. Liu, Y. Wang, S. Wang, S. Ma and W. Gao, "Implicit-selected transform in video coding," in *Proc. IEEE Intl Conf. Multimedia Expo Workshops*, pp. 1-6, 2020.

[22] J. Pfaff, H. Schwarz, D. Marpe, B. Bross, S. De-Luxán-Hernández, et al. "Video compression using generalized binary partitioning, trellis coded quantization, perceptually optimized encoding, and advanced prediction and transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1281-1295, 2020.

[23] W. Park, B. Lee and M. Kim, "Fast computation of integer DCT-V, DCT-VIII, and DST-VII for video coding," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5839-5851, Dec. 2019.

[24] M.J. Garrido, F. Pescador, M. Chavarrías, P.J. Lobo and C. Sanz, "A 2-D multiple transform processor for the versatile video coding standard," *IEEE Trans. Consum. Electron.*, vol. 65, no. 3, pp. 274-283, Aug. 2019.

[25] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao and D. Xu, "An end-to-end learning framework for video compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.43, no.10, pp. 3292-3308, Oct., 2021.

[26] K. Yang, D. Liu, F. Wu, "Deep learning-based nonlinear transform for HEVC Intra coding," in *Proc. IEEE Intl Conf. Visual Commun. Image Process.(VCIP)*, pp.387-390, Dec. 2020.

[27] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *J. Vis. Commun. Image Represent.*, vol. 41, pp. 406-413, 2016.

[28] C.-C. J. Kuo, "The CNN as a guided multilayer RECOS transform," *IEEE Signal Process. Mag.*, vol. 34, no. 3, pp. 81-89, 2017.

[29] C.-C. J. Kuo and Y. Chen, "On data-driven Saak transform," *J. Vis. Commun. Image Represent.*, vol. 50, pp. 237-246, 2018.

[30] X. Zhang, S. Kwong and C.-C. J. Kuo, "Data-driven transform based compressed image quality assessment," *IEEE Trans. Circuits Syst. Video Technol.*, 2021, in press, doi: 10.1109/TCSVT.2020.3041639.

[31] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan and Y. Chen, "Interpretable convolutional neural networks via feedforward design," *J. Vis. Commun. Image Represent.*, vol. 60, pp. 346-359, 2019.

[32] Y. Chen and C.-C.J. Kuo, "Pixelhop: a successive subspace learning (SSL) method for object recognition," *J. Vis. Commun. Image Represent.*, Jan. 2020.

[33] M. Zhang, H. You, P. Kadam, S. Liu and C. -C. J. Kuo, "PointHop: an explainable machine learning method for point cloud classification," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1744-1755, Jul. 2020.

[34] N. Li, Y. Zhang, Y. Zhang and C.-C. J. Kuo, "On energy compaction of 2D Saab image transforms," in *Proc. Asia-Pacific Signal Info. Process. Assoc. Annu. Summit Conf.*, Lanzhou, China, Nov. 2019.

[35] H. Lohscheller, "A subjectively adapted image communication system," *IEEE Trans. Commun.*, vol. 32, no. 12, pp. 1316-1322, Dec. 1984.

[36] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, and S. Kwong, "Effective data driven coding unit size decision approaches for HEVC Intra coding, *IEEE Trans Circuits Syst. Video Technol.*, vol. 28, no.11, pp. 3208-3222, Nov. 2018.

[37] L. Zhu, Y. Zhang, S. Kwong, X. Wang, and T. Zhao, "Fuzzy SVM based fast coding unit decision in HEVC,"*IEEE Trans. Broadcast.*, vol.64, no.3, pp. 681-694, Mar. 2018

[38] R.M. Gray, D.L. Neuhoff. "Quantization," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, Oct. 1998.

[39] T. Wiegand and H. Schwarz, "Source coding: part I of fundamentals of source and video coding," *Found. Trends Signal Process.*, pp. 1-222, Jan. 2011.

[40] L. Xu, X. Ji, W. Gao and D. Zhao, "Laplacian distortion model (LDM) for rate control in video coding," in *Proc. Pacific Rim Conf. Multimedia*, Berlin, Heidelberg, pp. 638-646, 2007.

[41] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *VCEG of ITU-T*, document VCEG-M33, 13th Meeting, Austin, Texas, USA, Apr. 2001.