



HAL
open science

Hardware-Friendly Multiple Transform Selection Module for the VVC Standard

Wassim Hamidouche, Pierrick Philippe, Sid Ahmed Fezza, Mounir Haddou,
Fernando Pescador, Daniel Menard

► **To cite this version:**

Wassim Hamidouche, Pierrick Philippe, Sid Ahmed Fezza, Mounir Haddou, Fernando Pescador, et al.. Hardware-Friendly Multiple Transform Selection Module for the VVC Standard. IEEE Transactions on Consumer Electronics, 2022, 68 (2), pp.96-106. 10.1109/TCE.2022.3163345 . hal-03628463

HAL Id: hal-03628463

<https://hal.science/hal-03628463v1>

Submitted on 12 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Hardware-Friendly Multiple Transform Selection Module for the VVC Standard

Wassim Hamidouche, *Member, IEEE*, Pierrick Philippe, Sid Ahmed Fezza, Mounir Haddou, Fernando Pescador, *Senior Member, IEEE*, and Daniel Menard

Abstract—The H.266/versatile video coding (VVC) standard is the most recent ITU/ISO video coding standard finalized in July 2020. VVC includes several new coding tools at different levels of the coding scheme. These coding tools enable a significant bitrate saving of up to 50% for the same subjective video quality than its predecessor H.265/high efficiency video coding (HEVC). Among these tools, we can cite the multiple transform selection (MTS) which selects at the encoder horizontal and vertical transforms among three trigonometrical transforms, including discrete cosine transform (DCT) type II, discrete sine transform (DST) type VII and DCT type VIII. Unlike the DCT-II, the DST-VII does not have efficient fast algorithmic implementation. Moreover, the MTS increases the memory usage required to store the coefficients of the three transforms. Consequently, this paper targets an efficient approximation of the DST-VII kernel based on the DCT-II and adjustment stage. The approximation of the DST-VII is modeled as an integer optimization problem jointly minimizing the error and the orthogonality of the approximate DST-VII under sparsity constraint of the adjustment stage. The sparse nonlinear optimizer (SNOPT) solver with an additional relaxation constraint is used to solve the problem and find the best sparse adjustment band matrices for different transform sizes. The DCT-VIII is then computed from the approximate DST-VII with pre/post processing operations involving only sign changes and input/output reordering. The proposed approximation provides a significant reduction in both arithmetic operations and memory usage. Moreover, it preserves the coding gain brought by the MTS under the VVC reference software. These advantages make our solution suitable for energy-efficient hardware H.266/VVC encoders and decoders deployed on consumer electronic devices.

Index Terms—Multiple transform selection, VVC, Transform approximation, DCT, DST, Complexity reduction.

I. INTRODUCTION

THE new video applications together with emerging video contents in high frame rate (HFR), high dynamic range

W. Hamidouche and D. Menard are with Univ. Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France (e-mail: {wassim.hamidouche,daniel.menard}@insa-rennes.fr).

P. Philippe is with Orange Labs, Cesson-Sévigné, France (e-mail: pierrick.philippe@orange.com).

SA. Fezza is with National Institute of Telecommunications and ICT, Oran, Algeria (e-mail: sfezza@inttic.dz).

M. Haddou is with Univ. Rennes, INSA Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France (e-mail: mounir.haddou@insa-rennes.fr).

F. Pescador is with CITSEM at Universidad Politécnica de Madrid, Madrid, Spain (e-mail: fernando.pescador@upm.es).

This work was supported by both the Energy Efficient Enhanced Media Streaming (3EMS) project and the Eco-VVC Hubert Curien Partnerships (PHC) Maghreb project. The 3EMS project is jointly funded by Brittany region and Rennes Métropole. The PHC project is funded by the MESRS (Algeria), MEN, CNRST (Morocco), MESRS (Tunisia), MEAE, MESRI (France) through the PHC Maghreb 2021, No 45988WG (Eco-VVC project).

(HDR) and omnidirectional 360° formats rise the need to develop a new video coding standard with coding efficiency beyond the H.265/high efficiency video coding (HEVC) standard. Joint video experts team (JVET) has launched a call for proposal (CfP) for the development of a new video coding standard called H.266/versatile video coding (VVC) [1], [2] in 2017. H.266/VVC was finalized in July 2020 enabling up to 50% of bitrate reduction with respect to H.265/HEVC for the same subjective video quality [3], [4]. H.266/VVC relies on the classical hybrid video coding structure combining inter/intra predictions with transform coding. Several new coding tools have been adopted during the standardization process at different levels of the encoder, including frame partitioning, intra/inter predictions, transform/quantization, in-loop filters and entropy engine.

For instance, the multiple transform selection (MTS) has been adopted in H.266/VVC, which allows the encoder to select a couple of horizontal and vertical transforms from predefined sets of transforms. These sets consist of kernels among three trigonometrical transforms, namely discrete cosine transform (DCT) type II, discrete sine transform (DST) type VII and DCT type VIII. In the rest of this chapter, DCT and DST of types X and Y are denoted as DCT-X and DST-Y, respectively. The used set of vertical and horizontal transforms is defined based on intermediate coding parameters, such as intra prediction mode and block size. MTS enables a significant coding gain of 0.84% in all intra (AI) and 0.33% in random access (RA) coding configurations [5]. This coding efficiency is achieved at the expense of a slight increase in the encoder complexity estimated to 115% [5] under the H.266/VVC reference software (VTM-5.0). The coding gain enabled by the MTS was even higher under the VTM-3.0 with 2.81% and 1.26% bit rate reductions [6] for AI and RA configurations, respectively.

It has been shown in several studies [7]–[11] that the hardware implementation of the MTS on field-programmable gate array (FPGA) platform would require high logic and memory resources. This would be a bottleneck for implementing real time hardware VVC encoder and decoder on FPGA/application-specific integrated circuit (ASIC) platforms with limited logic and memory resources. Therefore, there is a real need to provide a low complexity solution that requires low logic resources while preserving the coding gain of the MTS. The DCT-II has been well studied and optimized with fast algorithmic implementations [12]–[14], while DST-VII/DCT-VIII still need to be optimized for lightweight hardware implementation [15], [16]. Moreover, hardware imple-

mentation of the DCT-II is already integrated in the previous generations of video coding standards.

This paper proposes to approximate the basis functions of DST-VII using the basis functions of DCT-II and a sparse band adjustment matrix A [17]. The DCT-VIII can be then linearly derived from DST-VII without any additional complexity. This approximation allows implementing the transformations in low-resources devices used in consumer electronic applications.

The contributions of this work are summarized as follows:

- Model the approximation of the DST-VII as a continuous constrained optimization problem. Then, this problem is solved with a sparse non linear solver to derive the adjustment matrices of integer values to approximate DST-VII transform of different sizes $N \in \{16, 32, 64\}$.
- Implement an efficient, lightweight and hardware-friendly solution for DST-VII and DCT-VIII approximations based on the DCT-II kernel with low number of multiplications and low memory usage, while preserving the coding gain of the MTS for consumer electronics applications.

The rest of this chapter is organized as follows. Section II first describes the MTS concept in the H.266/VVC standard, then Section II-B reviews the existing fast algorithmic implementations of the DCT-II and DST-VII transforms. The approximation of DST-VII is expressed in Section III as a continuous constrained optimization problem, which is solved to derive the adjustment matrices of different sizes. The performance of the proposed approximation is assessed in Section IV in terms of complexity, memory usage and coding efficiency under the VTM-3.0 software. Finally, Section V concludes this chapter.

II. RELATED WORK

In this section, we first provide a background on trigonometrical transforms used in H.266/VVC, then their fast algorithmic implementations are reviewed.

A. Trigonometrical transforms in H.266/VVC

In this section, we first give the definition of the three trigonometrical transforms used in the H.266/VVC standard, then we describe the concept of MTS in the recent video encoders, especially H.266/VVC.

1) *Transforms for video compression:* The orthogonal basis functions of the DCT-II transform are computed by (1)

$$C_{2,i,j}^N = \gamma_i \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(i-1)(2j-1)}{2N}\right), \quad (1)$$

$$\text{with } \gamma_i = \begin{cases} \sqrt{\frac{1}{2}} & i = 1, \\ 1 & i \in \{2, \dots, N\}. \end{cases}$$

The basis functions of DST-VII and DCT-VIII are given by (2) and (3), respectively.

$$S_{7,i,j}^N = \frac{2}{\sqrt{2N+1}} \sin\left(\frac{\pi(2i-1)j}{2N+1}\right) \quad (2)$$

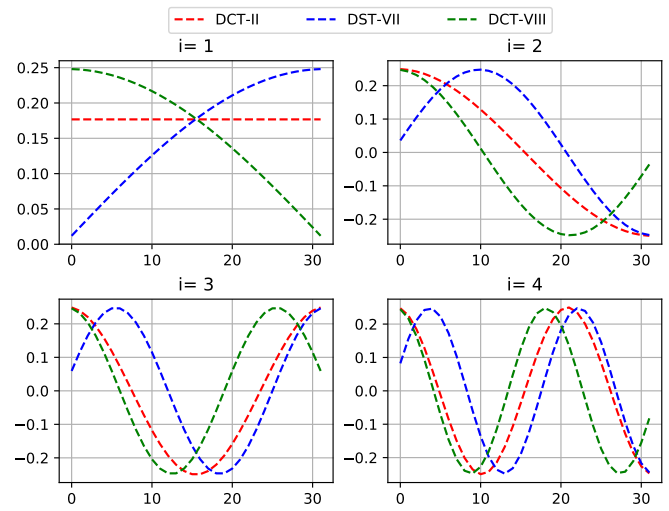


Fig. 1: First four basis of DCT-II, DCT-VIII and DST-VII for $N = 64$.

$$C_{8,i,j}^N = \frac{2}{\sqrt{2N+1}} \cos\left(\frac{\pi(2i-1)(2j-1)}{2(2N+1)}\right) \quad (3)$$

where $(i, j) \in \{1, \dots, N\}^2$ are the two indices of the transform matrix of dimension $N \times N$.

In the VVC standard, those basis functions are approximated by an integer representation, where each coefficient is encoded on $\beta = 8$ bits. Therefore, the transform coefficients are within the interval $[-2^{\beta-1} + 1, 2^{\beta-1}] \cap \mathbb{Z}$.

This paper is focused on the approximation of the DST-VII transform. The DCT-VIII can then be linearly derived from the DST-VII at no additional computational complexity, involving only vector reflection matrix (Γ^N) and sign changes matrix (Λ^N) as expressed in (4)

$$C_8^N = \Lambda^N \cdot S_7^N \cdot \Gamma^N, \quad (4)$$

where C_8^N and S_7^N are the coefficients matrices of DCT-VIII and DST-VII transforms, respectively, while matrices Λ^N and Γ^N are computed by (5) and (6), respectively.

$$\Gamma_{i,j}^N = \begin{cases} 1, & \text{if } j = N - i + 1, \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

$$\Lambda_{i,j}^N = \begin{cases} (-1)^{i-1}, & \text{if } j = i, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

with $i, j \in \{1, 2, \dots, N\}$.

Figure 1 illustrates the four first basis of DCT-II, DCT-VIII and DST-VII transforms.

2) *Multiple transform selection:* In the previous standard H.265/HEVC, separable transforms has been widely investigated [18], [19]. This standard only considers DCT-II along with DST-VII, used only for intra luma blocks of size 4×4 [20]. The transforms competition has then integrated into the joint exploration model (JEM) software [21]. This development generates five trigonometrical transform types (DCT-II, V and VIII, and DST-I and VII) and enables a significant increase in coding efficiency estimated to 3% of bitrate reduction [21].

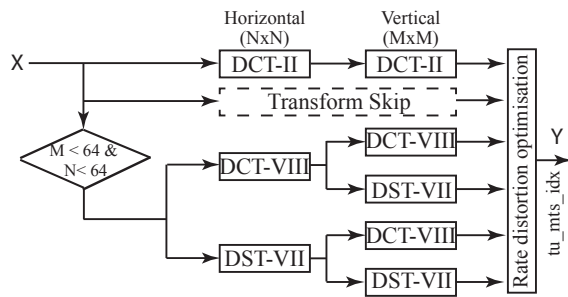


Fig. 2: X is the input block of residuals, Y is the output transformed block and MTS flag is the index of the selected set of transforms. Note that DCT-VIII and DST-VII are applied only on blocks of width and height lower than 64.

However, this improvement comes at the expense of both memory increase (used for the coefficients of the transforms) and complexity overhead required to test the transform candidates. To reduce the complexity, subsets of transform candidates are defined offline. Therefore, subsets of transforms are tested considering the prediction configurations such as the intra prediction mode and the block size [21].

The MTS in H.266/VVC relies only three transform types, including DCT-II, DCT-VIII and DST-VII. As illustrated in Figure 2, the MTS concept selects, for Luma blocks of size lower than 64, a set of transforms that minimizes the rate distortion cost among five transform sets and the skip configuration. However, only DCT-II is considered for chroma components and Luma blocks of size 64. The the MTS is activated at the encoder with the *sps_mts_enabled_flag* flag defined at the sequence parameter set (SPS) level. Two other flags are defined at the SPS level to signal whether implicit or explicit MTS signalling is used for intra and inter coded blocks, respectively. For the explicit signalling, used by default in the common test conditions (CTC), the *tu_mts_idx* syntax element signals the selected horizontal and vertical transforms as specified in Table I. This flag is coded with truncated rice (TRp) code with a rice parameter $p = 0$ and $cMax = 4$.

B. Fast computation methods of DCT/DST

Fast computing algorithms for DCTs/DSTs have been proposed in the literature. Most of them try to reduce the number of multiplications and additions compared to the matrix multiplication required for $N \times N$ square matrix, N^3 multiplications and $N^2(N - 1)$ additions, i.e., $\mathcal{O}(N^3)$. This computational complexity can be further reduced to $\mathcal{O}(N^{2.373})$ which is the state-of-the-art lower bound [22]. Some of the DCTs/DSTs offer symmetry, decomposition and recursion properties allowing to design fast and low complexity algorithms. The 1D transform of a residual vector x of size $N \times 1$ using the DST-VII basis S_7 is described in Equation (7)

$$y = S_7^N \cdot x. \quad (7)$$

Table II gives the number of multiplications and additions of the state-of-the-art fast implementations for both DCT-II and DST-VII 1D transforms to process a vector x of size $N \in \{8, 16, 32, 64\}$. The number of operations for is low

TABLE I: Primary transform signaling in VVC.

tu_mts_idx	Transform Direction	
	Horizontal Transform	Vertical Transform
0	DCT-II	DCT-II
1	DST-VII	DST-VII
2	DCT-VIII	DST-VII
3	DST-VII	DCT-VIII
4	DCT-VIII	DCT-VIII

compared to the naive multiplication for DCT-II while it significantly increases for DST-VII.

Zhang *et al.* [23] have investigated three properties of the DST-VII, exploring factorization to save multiplications. These properties allow reducing the number of multiplications by coefficients of the same absolute value within a basis function (row) in only one multiplication operation. This technique enables reducing the number of operations required by matrix multiplication without introducing any approximation. Rezni [24] has explored the relation between DCT-II and DST-VII transforms that enables their joint computation for certain transform sizes. It has also been shown in [25] that DCT-II of odd size is equivalent to computing the same length discrete Fourier transform (DFT). Thus, the fast DFT algorithm can be used to compute certain sizes of DCT-II and DST-VII transforms. Park *et al.* [26] explore the equality property between DST-VII and $(2N + 1)$ -point DFT to compute inverse/forward N -point DCT-VII. The linear relationship between DST-VII and DCT-VIII is then used to speedup the computation of the DCT-VIII. The DCT-V is approximated using similar properties by taking benefit of the relationship between DCT-VI and $(2N - 1)$ -point DFT, and the linear relation between DCT-VI and DCT-V. The $(2N + 1)$ -point and $(2N - 1)$ -point DFTs can efficiently be computed by Winograd fast Fourier transform (FFT) [27] for DFTs with the powers of prime lengths, while prime-factor FFT [28] is used for DFTs in the lengths that are the multiplications of numbers in relative prime.

The main contribution of this work is to propose and implement approximations for DST-VII for large sizes blocks ($N > 8$) that meet the following three main requirements:

- Lightweight solution reducing both the number of operations than the existing in previous fast computing algorithms proposed in [16], [23] and the memory usage to store the transform coefficients.
- Hardware-friendly solution that leverages the existing forward DCT-II and inverse DCT-II fast implementations to preserve memory and logic resources.
- Increase the coding efficiency achieved by the MTS under the VVC common test conditions (CTC) [29].

III. PROPOSED APPROXIMATION MODEL OF DST-VII

A. Model description

The DST-VII approximation is expressed as a constrained optimization problem. The approximate DST-VII is expressed

TABLE II: DCT-II and DST-VII performance analysis based on the number of multiplications and additions of the state of the art proposed algorithms

Transforms	N = 8				N = 16				N = 32				N = 64			
	Ref.	+	×	All	Ref.	+	×	All	Ref.	+	×	All	Ref.	+	×	All
DCT-II	[13]	29	11	40	[13]	81	31	112	[14]	209	80	289	[14]	192	513	707
DCT-II (HEVC)	[20]	37	24	61	[20]	113	86	199	[20]	401	342	743	[20]	807	683	1490
DST-VII	[15]	77	21	98	[16]	150	146	296	–	–	–	–	–	–	–	–
DST-VII	[23]	–	–	–	[23]	155	127	282	[23]	718	620	1338	[23]	2331	2207	4538
DST-VII	[26]	77	21	98	[26]	125	42	167	[26]	279	93	372	–	–	–	–
Matrix Multiplication	–	56	64	120	–	240	256	496	–	992	1024	2016	–	4032	4096	8128
Proposed DST-VII ($\theta = 4$)	–	24	32	56	–	48	64	112	–	96	128	224	–	192	256	448
Proposed DST-VII ($\theta = 5$)	–	32	40	72	–	64	80	144	–	128	160	288	–	256	320	576
Proposed DST-VII ($\theta = 6$)	–	40	48	88	–	80	96	176	–	160	192	352	–	320	384	704

according to the DCT-II [30], [31], as follows

$$\hat{S}_7^N = \Lambda^N \cdot [C_2^N]^T \cdot \Gamma^N \cdot A^N, \quad (8)$$

where $\Lambda^N \cdot [C_2^N]^T \cdot \Gamma^N$ is equivalent to the DST-III transform, Λ^N and Γ^N matrices are calculated using (5) and (6), respectively, and A^N being a sparse band matrix of size $N \times N$.

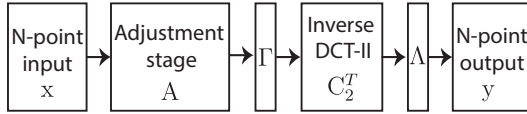


Fig. 3: Approximate forward DST-VII transform.

The inverse approximate DST-VII, used at the decoder side, involves the use of forward DCT-II

$$[\hat{S}_7^N]^T = [A^N]^T \cdot \Gamma^N \cdot C_2^N \cdot \Lambda^N. \quad (9)$$

The approximation workflows of the forward DST-VII and inverse DST-VII are illustrated in Figures 3 and 4, respectively. This approximate DST-VII \hat{S}_7 and its transpose, initially proposed in [30], reduce its computational complexity since it involves the DCT-II (Table II) and a multiplication by a band matrix A . Therefore, the complexity of this approximation is equal to the complexity of the DCT-II plus the complexity of the multiplication by the band matrix A , which depends on the maximum number of non-zero coefficients by row θ . The complexity of the multiplication by the matrix A in terms of numbers of multiplications and additions are given by θN and $(\theta - 1)N$, respectively. In this proposal, three values of non-zero coefficients by row are considered $\theta = \{4, 5, 6\}$. These three values offer a good trade-off between complexity reduction and coding efficiency.

Table II demonstrates that number of operations in terms of both additions and multiplications required for the three configurations of the approximate DST-VII is lower than the operations used in DST-VII fast implementations for large transform size (i.e., $N > 16$). Moreover, this architecture is suitable for hardware implementation since the logic and memory resources of DCT-II transform can be reused to calculate DST-VII and DCT-VIII transforms.

The proposed approach consists in minimizing the weighted least-squares error between the approximate version \hat{S}_7 and the original one DST-VII S_7

$$E(A) = \sum_{i=1}^N \omega_i \sum_{j=1}^N (S_{7i,j}^N - \hat{S}_{7i,j}^N)^2, \quad (10)$$

where ω_i is a weight vector of size N representing the

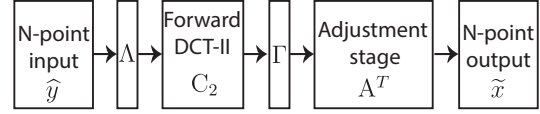


Fig. 4: Approximate inverse DST-VII transform.

relative importance of the frequency components. When the ω_i is constant equal to 1, the error function corresponds to the squared Frobenius norm.

The approximate DST-VII transform \hat{S}_7 must meet two constraints related to the matrix A , which are sparsity and orthogonality. This last requirement is an important property of the transform core since it enables the use of its transpose in inverse transform to recover the original signal (energy conservation). The orthogonality of the adjustment matrix A can be expressed by (11)

$$O(A) = \|A^N \cdot [A^N]^T - I^N\|_2^2, \quad (11)$$

where I^N is the identity matrix of size $N \times N$ and $\|\cdot\|_2$ stands for the Euclidean norm. Therefore, the objective function of this constrained optimization problem can be expressed with a Lagrangian multiplier λ as follows

$$\underset{A}{\text{minimize}} \quad E(A^N) + \lambda O(A^N). \quad (12)$$

This objective function aims at minimizing the trade-off between error $E(A)$ and orthogonality $O(A)$ of the approximate DST-VII \hat{S}_7 , where this trade-off is tuned by the Lagrangian parameter λ . The optimal solution of the optimization problem of (12) consists in the matrix A^* that leads to the original DST-VII S_7 expressed by (13).

$$A^* = \Gamma^N \cdot C_2^N \cdot \Lambda^N \cdot S_7^N, \quad (13)$$

with $E(A^*)$ and $O(A^*)$ terms are both equal to zero.

However, this optimal solution is not appropriate because it neither obtain integer values and neither reveal a sparse property, leading to fewer arithmetic operations. Then, consider the A^* matrix with values multiplied by 2^β (with β the bit-depth set to 7 bits) and rounded to the nearest integer. An example of the A^* matrix for $N = 8$ $A_{8,8}^*$ is provided in (14).

$$\begin{pmatrix} 127 & 11 & -6 & 4 & -2 & 2 & -1 & 0 \\ -10 & 125 & 20 & -10 & 6 & -4 & 2 & -1 \\ 6 & -16 & 122 & 30 & -13 & 7 & -4 & 1 \\ -4 & 10 & -22 & 118 & 39 & -15 & 7 & -2 \\ 4 & -8 & 14 & -27 & 114 & 47 & -15 & 4 \\ -3 & 7 & -11 & 18 & -32 & 110 & 53 & -10 \\ 3 & -6 & 10 & -15 & 23 & -38 & 109 & 47 \\ -2 & 4 & -7 & 10 & -15 & 22 & -39 & 118 \end{pmatrix} \quad (14)$$

The most significant absolute values (highlighted in (14)) of the matrix are around the diagonal and lower absolute values are located at lower-left and upper-right. This property of the adjustment matrix A is stronger for adjustment matrices of higher sizes $N \in \{16, 32, 64\}$.

In this work, adjustment band matrix that minimizes the trade-off between error and orthogonality is looked for with the constraint of A to include few integer values different from zero. The discrete constrained optimization problem is presented in (15).

$$\begin{aligned} & \underset{A}{\text{minimize}} && E(A^N) + \lambda O(A^N), \\ & \text{subject to} && A_{i,j}^N = 0, \quad \forall j > i + \lceil \theta/2 \rceil, \\ & && A_{i,j}^N = 0, \quad \forall j \leq i - \lceil \theta/2 \rceil, \\ & && i, j \in \{1, \dots, N\}^2, \\ & && A_{i,j}^N \in \mathbb{Z} \cap [-2^\beta + 1, 2^\beta], \\ & && \lambda \in \mathbb{R}^+ \end{aligned} \quad (15)$$

It has been shown in [32] that the DST-VII is optimal in terms of energy packing for image intra-predicted residuals. Indeed, these residuals have an auto-correlation matrix which is a tri-diagonal matrix R_x of size $N \times N$ expressed by (16).

$$\begin{aligned} R_{x,i,i}^N &= b, \\ R_{x,i,i+1}^N &= c, \\ R_{x,j-1,j}^N &= a, \\ R_{x,N,N}^N &= b - \alpha, \end{aligned} \quad (16)$$

with $(a, b, c, \alpha) = (-1, 2, -1, 1)$ and $1 \leq i < N$, $1 < j \leq N$.

The eigenvalue-vectors of the matrix R_x are the basis of the DST-VII transform [33]. The matrix R_x is expressed as follows:

$$R_x^N = \begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -1 & \dots & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{pmatrix}_{N \times N}$$

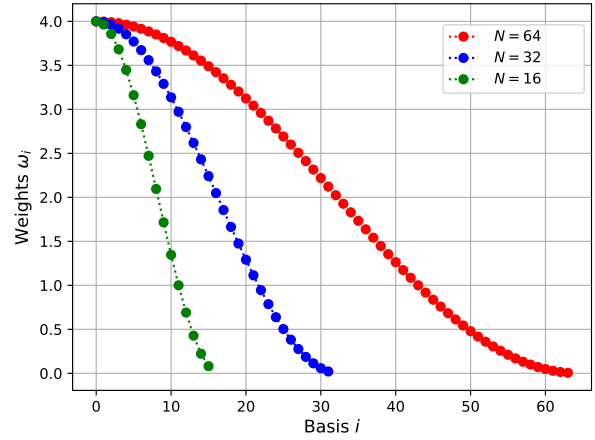


Fig. 5: The weight coefficients used for the approximate DST-VII transform optimization for different sizes $N \in \{16, 32, 64\}$.

Therefore, for the approximation of the DST-VII, the relative importance of the approximation basis with the eigenvalues of the matrix R_x is proposed to weight. This approach gives more relevance to the lower frequency range where an important part of the signal energy stands. According to [34], the eigenvalues are computed by (17)

$$\omega_i = b + 2\sqrt{ac} \cos\left(\frac{2i\pi}{2N+1}\right), \quad i = 1, \dots, N. \quad (17)$$

Figure 5 illustrates the weights ω_i assigned to the basis functions in the optimization of the approximate DST-VII transform for different kernel sizes $N \in \{16, 32, 64\}$.

B. Constrained optimization problem

This constrained integer optimization problem is non linear and non convex. We might use genetic algorithms or integer programming [35] to solve this problem. However, such approaches may be complex and take long time to converge towards an optimal solution. Also, an exhaustive search would result in calculating large number of combinations of $(2^{\beta+1} + 1)^{\theta N}$. Thus, our choice went to use the sparse nonlinear optimizer (SNOPT) solver [36]. This latter is based on sparse sequential quadratic programming (SQP) with limited-memory quasi-Newton approximations to the Hessian of the Lagrangian. The SQP programming is effective for solving constrained optimization problems with smooth nonlinear functions in the objective and constraints. We consider the problem as non linear problem with an objective function and a set of equality constraints. We add another relaxation constraint which ensures that the non-zero coefficients of the matrix A are within the integer subset $A_{i,j} \in \mathbb{Z} \cap [-2^\beta + 1, 2^\beta]$

$$\sum_{i=1}^N \sum_{j=1}^N |\sin(\pi A_{i,j}^N)| = 0. \quad (18)$$

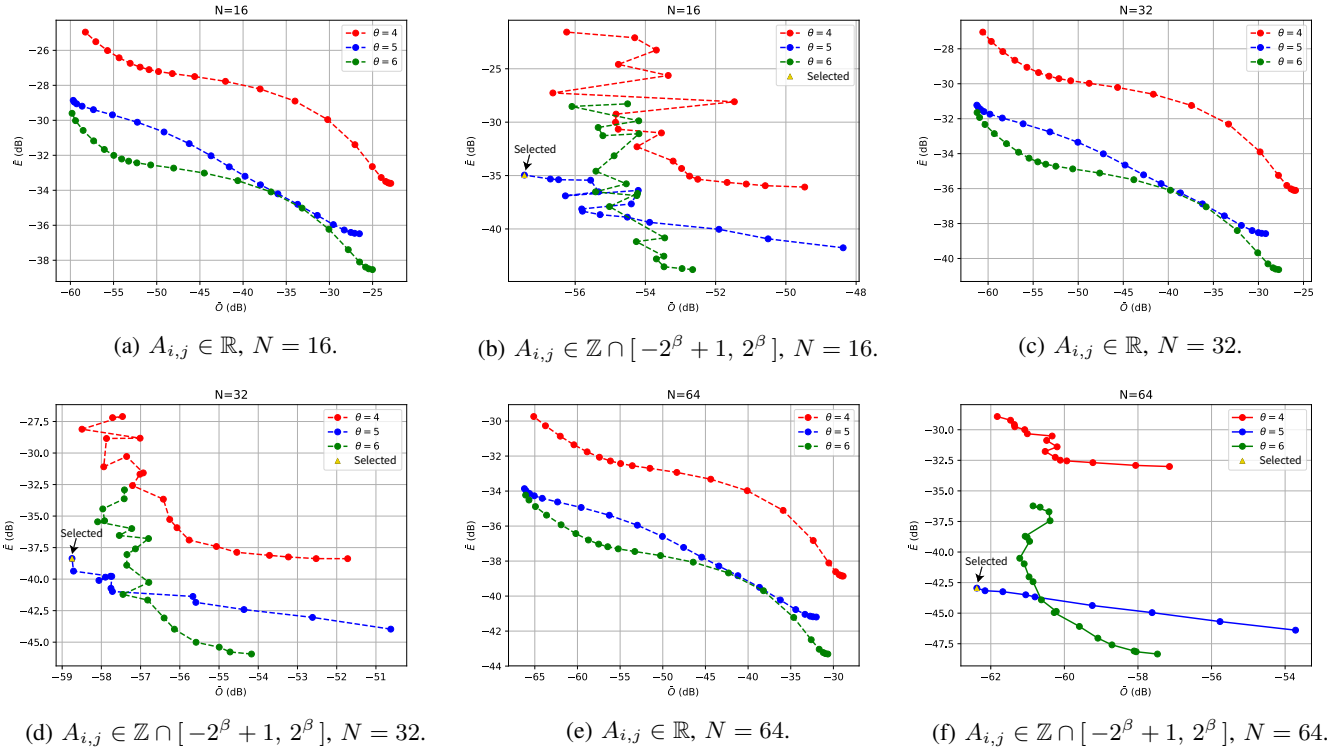


Fig. 6: Error $E(A^*)$ versus orthogonality $O(A^*)$ performance in dB of the proposed solution for different λ values.

To make sure that the SNOPT solver converges to a valid solution and not towards a local optimum, we randomly initialize the solver with different initial solutions. The solver converges to a similar solution whatever the initial solution. This does not assert that the solver converges to the optimal solution, but the obtained solution can be reached from different initial solutions. The problem was expressed with a mathematical programming language (AMPL) [37] that supports the SNOPT solver. The optimization program was then carried out on the NEOS server, which is a free internet-based service for solving numerical optimization problems [38], [39].

C. Error and orthogonality performance

To assess the performance of the proposed solution, we plot the error $E(A^*)$ versus the orthogonality $O(A^*)$ for different values of the Lagrangian multiplier λ . The error and orthogonal metrics are expressed in logarithmic scale computed as follow

$$\begin{aligned} \bar{E}(A^*) &= 10 \log_{10} (E(A^*)/N^2), \\ \bar{O}(A^*) &= 10 \log_{10} (O(A^*)/N^2). \end{aligned} \quad (19)$$

Figure 6 illustrates the performance in terms of orthogonality and error of the DST-VII approximated with the adjustment matrix of real coefficients (Figures 6a, 6c, 6e) and then constrained to integer coefficients with the additional constraint in (18) (Figures 6b, 6d, 6f). We can notice from Figures 6a, 6c, 6e that different λ values enables deriving solutions with different trade-off between orthogonality and error. Moreover, considering more non-zero coefficients by

row enables achieving a more efficient solution in both orthogonality and error at the expense of higher complexity. However, the integer constraint decreases the performance, especially in terms of orthogonality. We can notice that the proposed solution with only five non zeros coefficients $\theta = 5$ reaches a better trade-off between error and orthogonality compared to solutions obtained with $\theta = 6$ for different transform sizes.

Table III provides the error and orthogonality scores obtained for different λ values. We also provide the error and orthogonality scores of the integer DST-VII considered in the VTM reference software. The proposed solution may reach any trade-off between error and orthogonality of the approximate DST-VII. Moreover, the selected couple of error-orthogonality highlighted in black triangle in Figures 6b, 6d and 6f for $\theta = 5$ are selected to assess their coding performance in the VTM software under the CTC, and low quantization parameter (QP) (high bitrate) configurations.

Figure 7 illustrates the 16 first basis functions of the DST-VII computed by (2), DST-VII used in the VVC test model (VTM) and the proposed approximate DST-VII computed by (8). While the basis functions of the DST-VII and DST-VII used in the VTM are perfectly overlapping, those of the proposed approximate DST-VII are very close to the orthogonal DST-VII basis. This shows the accuracy of the proposed approximation with basis function very close to DST-VII used in the VTM.

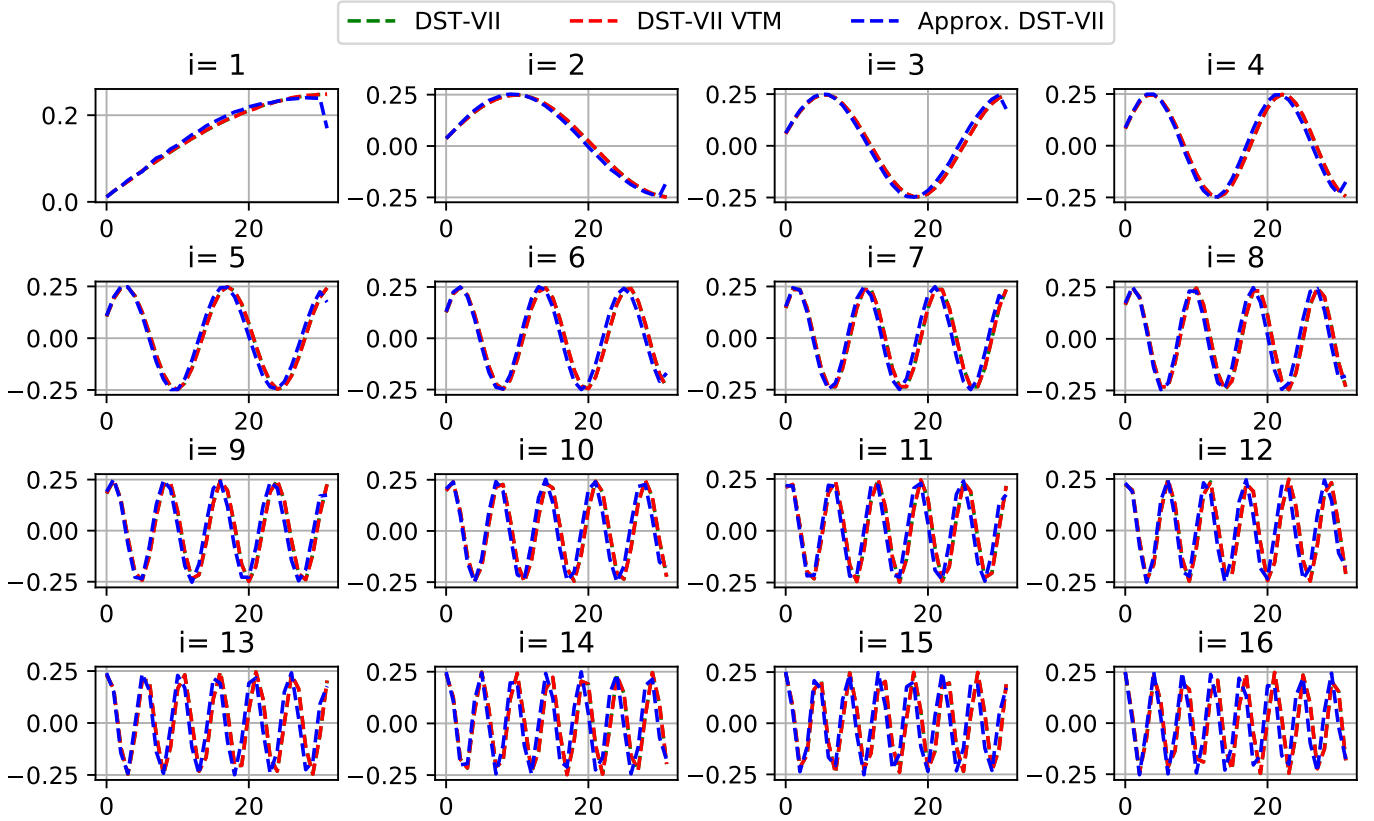


Fig. 7: The first 16 basis functions of DST-VII, DST-VIII VTM and the proposed DST-VII for $N = 32$.

TABLE III: Performance of the approximate DST-VII based on error $\bar{E}(A^*)$ and orthogonality $\bar{O}(A^*)$ in dB for different Lagrangian parameters and $\theta = 5$.

Transforms	$N = 16$			$N = 32$			$N = 64$		
	λ	$\bar{E}(A^*)$	$\bar{O}(A^*)$	λ	$\bar{E}(A^*)$	$\bar{O}(A^*)$	λ	$\bar{E}(A^*)$	$\bar{O}(A^*)$
DST-VII	22.57	-39.11	-53.11	6.72	-42.41	-53.72	41.38	-43.62	-59.55
DST-VII	41.38	-38.66	-54.02	254.85	-40.65	-56.91	254.85	-43.24	-60.74
DST-VII	5273.28	-37.42	-54.54	5273.28	-39.52	-57.30	59526.76	-41.90	-61.26
DST-VII VTM	-	-59.56	-57.64	-	-60.14	-58.32	-	-65.10	-63.36

IV. RESULTS AND ANALYSIS

A. Experimental setup

The proposed approximation has been integrated in the reference software VTM version 3.0. The approximation is applied only on DST-VII and DCT-VIII of large sizes $N = \{16, 32, 64\}$. The performance of the proposed solution are assessed in this section in terms of Bjøntegaard delta rate (BD-BR) [40] with respect to the anchor VTM-3.0, in addition, the complexity in number of multiplications/additions and memory required to store the transform coefficients are measured. The BD-BR is computed over the test video sequences defined in the VVC CTCs over four QPs $\in \{22, 27, 32, 37\}$. The CTCs video sequences are grouped into seven classes as follows: A1 (3840×2160), A2 (3840×2160), B (1920×1080), C (832×480), D (416×240), E (1280×720), and F (832×480 to 1920×1080). These classes feature different frame rates, bit depths, motions, textures and spatial resolutions. Classes A to E include natural scene sequences and class F contains

specific screen content sequences.

The run time complexity reduction (CR) of the encoder and decoder is computed by (20).

$$CR = \frac{1}{4} \sum_{QP_i \in \{22, 27, 32, 37\}} \frac{T_P(QP_i)}{T_A(QP_i)} 100\%, \quad (20)$$

where T_P and T_A are the run times in second of the modified encoder/decoder with the approximate transforms and the anchor, respectively.

B. Complexity performance

Table II gives the complexity of the proposed approximation of DST-VII, for three values of θ , in terms of number of multiplications and additions. Compared with the state of the art solutions depicted in Table II, we can notice that the proposed solution provides the lowest number of multiplications/additions for both 4 and 5 non-zero coefficients per row ($\theta = 4, 5$) whatever the transform size. The number

TABLE IV: Performance (%) in terms of BD-BR and run time complexity reduction (CR) of approximate DST-VII and DCT-VIII (MTS enabled only for intra blocks).

Class	All Intra Main 10					Random Access Main 10					Low Delay B Main 10				
	BD-BR (%)			CR (%)		BD-BR (%)			CR (%)		BD-BR (%)			CR (%)	
	Y	U	V	Enc.	Dec.	Y	U	V	Enc.	Dec.	Y	U	V	Enc.	Dec.
A1	-0.01	-0.18	-0.08	96	80	-0.15	-0.48	-0.46	102	97	-	-	-	-	-
A2	0.14	0.10	0.03	96	84	0.03	0.08	0.06	101	98	-	-	-	-	-
B	0.12	0.06	0.08	96	83	0.05	0.00	-0.17	101	98	0.04	-0.35	-0.10	101	101
C	0.07	-0.02	-0.06	97	89	0.05	0.05	0.24	100	100	0.05	0.36	0.10	101	102
E	0.13	0.04	0.13	96	86	-	-	-	-	-	0.15	0.26	0.24	100	97
Av.	0.09	0.01	0.02	96	85	0.01	-0.07	-0.07	101	98	0.07	0.04	0.05	101	100
D	0.07	0.08	0.00	96	91	0.01	-0.23	-0.03	100	100	0.05	-0.02	0.43	101	100
F	0.07	0.16	0.14	96	92	0.05	0.22	0.28	100	98	0.12	0.46	0.34	101	102

TABLE V: Performance (%) in terms of BD-BR and run time CR of approximate DST-VII and DCT-VIII (MTS enabled for both intra and inter blocks).

Class	Random Access Main 10					Low Delay B Main 10				
	BD-BR (%)			CR (%)		BD-BR (%)			CR (%)	
	Y	U	V	Encoder	Decoder	Y	U	V	Encoder	Decoder
A1	-0.24	-0.28	-0.33	105	96	-	-	-	-	-
A2	-0.01	0.05	0.07	103	96	-	-	-	-	-
B	0.01	0.03	-0.03	103	96	-0.16	0.03	-0.05	107	101
C	0.01	0.14	0.29	101	95	-0.04	-0.07	0.18	104	102
E	-	-	-	-	-	-0.03	0.54	-0.11	107	103
Av.	-0.04	0.00	0.01	103	96	-0.09	0.13	0.01	106	102
D	0.04	-0.19	0.03	100	96	0.00	-0.29	0.10	102	101
F	0.08	0.20	0.32	102	97	-0.11	-0.04	0.28	105	101

of operations of the approximation includes the complexity of the DCT-II and the complexity required to perform the multiplication by the sparse band matrix A , which mainly depends on the number of non-zero coefficients per row θ .

The proposed approach is especially more efficient for large transform sizes reaching for $\theta = 5$, 37%, 68%, 84% and 92% of multiplication savings with respect to naive matrix multiplication for transform sizes $N = 8, 16, 32$ and 64 , respectively. The computation of DCT-VIII can be derived on hardware platform from the DST-VII involving only sign changes and input/output reordering without any additional computational complexity.

C. Coding performance

Firstly, it is important to emphasize that the Bjøntegaard delta rate (BD-BR) gains of the MTS under the VTM-3.0 are 2.81% and 1.26% in AI and RA coding configurations, respectively [6]. Table IV gives the performance of the proposed approximation for $\theta = 5$ in terms of both BD-BR and codec run time complexity with respect to the anchor VTM-3.0. The non-zero integer coefficients of the selected adjustment matrices are provided in Appendix A for $\theta = 5$ and $N = 16, 32$. The coding and complexity results are provided for three coding configurations including AI, RA and low delay B (LDB) when MTS is enabled only for intra blocks. We can note that including 64×64 DST-VII and DCT-VIII in the approximate MTS enables slight coding gain for high resolution video sequences (A1) especially in inter

configuration that uses more large blocks. However, slight coding loss lower than 0.2 % in average is observed for other resolutions. This limited coding loss is mainly caused by the accurate proposed approximations of the DST-VII and DCT-VIII.

The coding and complexity performance of our approximation are depicted in Table V for LDB and RA coding configurations when the MTS is enabled on both intra and inter blocks. The gains enabled by the large DST-VII and DCT-VIII is even higher when the MTS is enabled for inter blocks, especially for high resolution video A1. For other video classes, the coding performance remains similar to the anchor.

Regarding the encoding run time complexity, the encoding time is slightly higher with the approximation since the large transform size 64×64 for both DST-VII and DCT-VIII are enabled only in modified VTM and they are restricted to maximum size of 32×32 in the anchor. At the decoder side, the decoding run time is decreased thanks to the approximations of the DST-VII and DCT-VIII.

Table VI gives the BD-BR loss achieved by the proposed approximation at low QP configuration: $QP \in \{1, 5, 9, 13\}$. This measure is important to assess the performance of the proposed solution at high bitrate where more high frequency coefficients are encoded. The reported BD-BR scores show that the coding loss remains limited below 0.2% for Luma component.

Figure 8 illustrates frames #1 (intra) and #4 (bi-predicted)

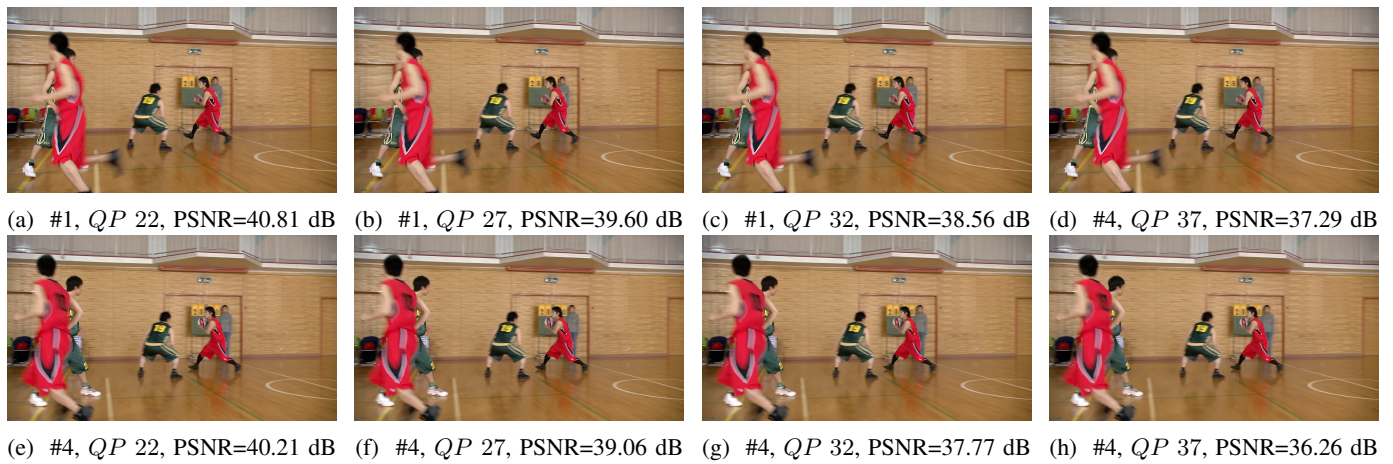


Fig. 8: Visual illustration of frames #1 (Intra) and #4 (bi-predicted) of *BasketballDrive* video (1080p) encoded at four QPs with the proposed MTS approximation solution.

TABLE VI: Performance (%) in terms of BD-BR and run time CR of approximate DST-VII and DCT-VIII in low QP configuration $QP \in \{1, 5, 9, 13\}$ (MTS enabled only for intra blocks).

Class	All Intra Main 10					Random Access Main 10				
	BD-BR (%)			CR (%)		BD-BR (%)			CR (%)	
	Y	U	V	Encoder	Decoder	Y	U	V	Encoder	Decoder
A1	0.13	0.2	0.20	97	100	0.11	0.29	0.39	99	100
A2	0.15	0.17	0.16	95	97	0.15	0.17	0.16	99	101
B	0.17	0.25	0.25	98	101	0.08	0.13	0.16	100	100
C	0.09	0.13	0.15	97	101	0.06	0.08	0.09	99	101
E	0.17	0.31	0.27	96	99	—	—	—	—	—
Av.	0.14	0.21	0.20	97	100	0.10	0.16	0.19	99	100
D	0.09	0.14	0.14	97	101	0.03	0.12	0.09	100	101
F	0.08	0.16	0.09	98	100	0.00	0.0	-0.01	100	101

TABLE VII: Memory usage for different transform types in number of coefficients (each coefficient is stored in 8 bits).

Block size	4	8	16	32	64	Total
DCT-II [20]	16	64	86	342	1366	1874
DST-VII [23]	16	64	256	1024	4096	5456
Proposed DST-VII $\theta = 4$	16	64	64	128	254	462
Proposed DST-VII $\theta = 5$	16	64	80	160	320	640
Proposed DST-VII $\theta = 6$	16	64	96	192	384	752

of *BasketballDrive* video sequence (1080p50) encoded with the approximate MTS at four QPs. We can notice the high visual quality of the decoded images even at high QP with similar coding visual artifacts as the anchor.

D. Memory usage

Table VII gives the memory required to store the coefficients of DCT-II using butterfly fast algorithm and the DST-VII computed with different fast algorithms including matrix multiplication [23] and approximation proposed in [31]. The proposed solution reduces the memory used to store the coefficients to approximate DST-VII compared to the factorization-based solution proposed in [23]. In fact, the proposed solution requires to storage only the DCT-II coefficients and those of

the sparse adjustment matrices at different sizes. This advantage is convenient for area-efficient hardware implementations supporting DCT-II, DST-VII and DCT-VIII.

V. CONCLUSION

In this paper, the authors have investigated the approximation of the DST-VII and DCT-VIII transforms to reduce the complexity and memory required to implement the VVC MTS block, especially on hardware platforms with limited logic and memory resources. Unlike the DCT-II, the DST-VII does not have efficient fast algorithmic implementation and it relies on naive matrix multiplication. The proposed approximation aims to leverage the well known DCT-II transform with low complexity pre- and post-processing operations. The approximation of the DST-VII has been modeled as a constrained optimization problem jointly minimizing error and orthogonality of the approximate DST-VII under sparsity constraint of the pre-processing stage. This integer optimization problem is solved by the SNOPT solver with an additional relaxation constrains to compute coefficients in the discrete domain.

The proposed approximation has been integrated in the VVC reference software VTM-3.0 and assessed in terms of complexity in number of required additions/multiplications, coding efficiency and memory usage. The obtained results showed that the proposed approximation achieved a significant multiplication saving up to 92% with respect to matrix multiplication for large transform size $N = 64$. Moreover, the coding gains brought by the MTS are preserved, while the required memory to store the transform coefficients is significantly reduced.

This approximation is very interesting to implement the H.266/VVC decoder in consumer electronic devices with very limited resources in term of power consumption and memory usage. The proposed algorithm can be easily integrated in an ASIC to implement an efficient H.266/VVC decoder.

REFERENCES

[1] 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),” *Proceedings of the IEEE*, pp. 1–31, 2021.
- [2] W. Hamidouche, T. Biatek, M. Abdoli, E. Francois, F. Pescador, M. Radosavljevic, D. Menard, and M. Raullet, “Versatile video coding standard: A review from coding tools to consumers deployment,” *IEEE Consumer Electronics Magazine*, pp. 1–1, 2022.
- [3] N. Sidaty, W. Hamidouche, O. Déforges, P. Philippe, and J. Fournier, “Compression Performance of the Versatile Video Coding: HD and UHD Visual Quality Monitoring,” in *2019 Picture Coding Symposium (PCS)*, 2019, pp. 1–5.
- [4] P. Philippe, W. Hamidouche, J. Fournier, and J.-Y. Aubier, “Subjective comparison of VVC and HEVC (AHG4),” in *JVET document 00451 (JVET-00451)*, Jul 2019.
- [5] W.-J. Chien, J. Boyce, Y.-W. Chen, R. Chernyak, K. Choi, R. Hashimoto, Y.-W. Huang, H. Jang, S. Liu, and D. Luo, “JVET AHG report: Tool reporting procedure (AHG13),” in *in JVET document 00013 (JVET-00013), 15th JVET Meeting: Gothenburg, SE*, Jul 2019.
- [6] W.-J. Chien, J. Boyce, R. Chernyak, K.-C. Francois, R. Hashimoto, Y.-W. Huang, S. Liu, and D. Luo, “JVET AHG report: Tool reporting procedure (AHG13),” in *in JVET document M0013 (JVET-M0013), 13th JVET Meeting: Marrakech, Morocco*, Jan 2019.
- [7] 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, October 2018.
- [8] M. Garrido, F. Pescador, M. Chavarrias, P. Lobo, and C. Sanz, “A High Performance FPGA-based Architecture for Future Video Coding Adaptive Multiple Core Transform,” *IEEE Trans. Consum. Electron.*, March 2018.
- [9] A. Mert, E. Kalali, and I. Hamzaoglu, “High Performance 2D Transform Hardware for Future Video Coding,” *IEEE Trans. Consum. Electron.*, vol. 62, no. 2, May 2017.
- [10] I. Farhat, W. Hamidouche, A. Grill, D. Menard, and O. Deforges, “Lightweight Hardware Implementation of VVC Transform Block for ASIC Decoder,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020.
- [11] A. Kammoun, W. Hamidouche, P. Philippe, O. Déforges, F. Belghith, N. Masmoudi, and J. F. Nezan, “Forward-Inverse 2D Hardware Implementation of Approximate Transform Core for the VVC Standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4340–4354, 2020.
- [12] G. Plonka and M. Tasche, “Fast and Numerically Stable Algorithms for Discrete Cosine Transforms,” *Linear Algebra and its Applications*, vol. 394, pp. 309 – 345, 2005.
- [13] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, “Practical fast 1-D DCT algorithms with 11 multiplications,” in *International Conference on Acoustics, Speech, and Signal Processing*, 1989, pp. 988–991.
- [14] M. Vetterli and H. J. Nussbaumer, “Simple FFT and DCT algorithms with reduced number of operations,” *IEEE J. Signal Process.*, vol. 6, no. 4, pp. 267 – 278, 1984.
- [15] R. K. Chivukula and Y. A. Reznik, “Fast Computing of Discrete Cosine and Sine Transforms of Types VI and VII,” in *Proc. SPIE*, vol. 8135, 2011, pp. 8135 – 8135 – 10.
- [16] M. Puschel and J. M. F. Moura, “Algebraic Signal Processing Theory: Cooley-Tukey Type Algorithms for DCTs and DSTs,” *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1502 – 1521, April 2008.
- [17] A. Said, H. Egilmez, V. Seregin, and M. Karczewicz, “Complexity Reduction for Adaptive Multiple Transforms (AMTs) using Adjustment Stages,” in *Document JVET-J0066 10th JVET Meeting: San Diego, CA, USA*, April 2018.
- [18] A. Arrufat, P. Philippe, K. Reuzé, and O. Déforges, “Low complexity transform competition for HEVC,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 1461–1465.
- [19] T. Biatek, V. Lorcy, and P. Philippe, “Transform Competition for Temporal Prediction in Video Coding,” *IEEE Trans. Circuits Syst. Video Technol.*, pp. 1–1, 2018.
- [20] P. Meher, S. Park, B. Mohanty, K. Lim, and C. Yeo, “Efficient integer dct architectures for hevcc,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 168–178, Jan 2014.
- [21] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin, “Joint Separable and Non-Separable Transforms for Next-Generation Video Coding,” *IEEE Transactions Image Processing*, vol. 27, no. 5, pp. 2514–2525, May 2018.
- [22] F. Le Gall, “Powers of Tensors and Fast Matrix Multiplication,” in *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation*, ser. ISSAC ’14. New York, NY, USA: ACM, 2014, pp. 296–303.
- [23] Z. Zhang, X. Zhao, X. Li, Z. Li, and S. Liu, “Fast Adaptive Multiple Transform for Versatile Video Coding,” in *Data Compression Conference (DCC)*, 2019.
- [24] Y. A. Reznik, “Relationship between DCT-II, DCT-VI, and DST-VII transforms,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 5642–5646.
- [25] M. T. Heideman, “Computation of an odd-length DCT from a real-valued DFT of the same length,” *IEEE Trans. Image Process.*, vol. 40, no. 1, pp. 54–61, Jan 1992.
- [26] 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.*, pp. 1–1, 2019.
- [27] S. Winograd, “On Computing the Discrete Fourier Transform,” *Mathematics of Computation*, vol. 32, no. 141, pp. 175–199, Jan. 1978.
- [28] I. J. Good, “The Interaction Algorithm and Practical Fourier Analysis,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 20, no. 2, pp. 361–372, Aug. 1958.
- [29] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, “JVET common test conditions and software reference configurations for SDR video,” in *JVET document K1010 (JVET-K1010)*, July 2018.
- [30] A. Said, H. Egilmez, V. Seregin, and M. Karczewicz, “Complexity Reduction for Adaptive Multiple Transforms (AMTs) using Adjustment Stages,” in *JVET document J0066 (JVET-J0066)*, April 2018.
- [31] A. Said, H. E. Egilmez, Y.-H. Chao, V. Seregin, and M. Karczewicz, “CE6.1.6: Efficient Implementations of AMT with Transform Adjustment Filters (TAF),” in *JVET document L0386 (JVET-L0386), 12th JVET Meeting: Macao, CN*, Oct. 2018.
- [32] A. Saxena and F. C. Fernandes, “DCT/DST-Based Transform Coding for Intra Prediction in Image/Video Coding,” *IEEE Transactions on Image Processing*, vol. 22, no. 10, pp. 3974–3981, Oct 2013.
- [33] A. K. Jain, “A Sinusoidal Family of Unitary Transforms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 4, pp. 356–365, Oct 1979.
- [34] W. chyan Yueh, “Eigenvalues of several tridiagonal matrices,” in *Applied Mathematics E-notes*, 2005, pp. 5–66.
- [35] L. A. Wolsey, “Integer Programming,” in *Wiley, ISBN: 978-0-471-28366-9*, Sep 1998.
- [36] P. E. Gill, W. Murray, and M. A. Saunders, “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” in *Society for Industrial and Applied Mathematics*, vol. 47, no. 1, 2015, pp. 99–131.
- [37] R. Fourer, D. Gay, and B. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, 01 2002, vol. 36.
- [38] J. Czyzyk, M. P. Mesnier, and J. J. Moré, “The neos server,” *IEEE Computational Science and Engineering*, vol. 5, no. 3, pp. 68–75, 1998.
- [39] E. D. Dolan, “The NEOS server 4.0 Administrative Guide,” in *Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory*, Oct. 2001.
- [40] G. Bjøntegaard, “Calculation of Average PSNR Differences Between RD-curves,” in *Document VCEG-M33 Thirteenth Meeting: Austin, Texas, USA*, April 2001.

APPENDIX A

16/32-POINT SELECTED ADJUSTMENT MATRICES WITH $\theta = 5$

The following matrix gives the non-zero coefficients of the adjustment matrix $A_{16 \times 16}^*$, where the matrix diagonal is highlighted in gray.

$$\begin{bmatrix} 128 & 4 & -1 & 0 & 0 \\ -4 & 128 & 8 & 0 & 0 \\ 1 & -8 & 127 & 10 & -1 \\ 1 & -10 & 127 & 14 & 0 \\ 2 & -14 & 126 & 16 & 1 \\ 2 & -16 & 126 & 16 & 1 \\ 1 & -16 & 125 & 24 & -1 \\ 2 & -23 & 123 & 26 & 1 \\ 6 & -25 & 124 & 19 & 2 \\ 3 & -19 & 125 & 21 & 2 \\ 1 & -21 & 124 & 24 & 3 \\ 2 & -24 & 123 & 27 & 3 \\ 2 & -27 & 121 & 31 & 5 \\ 4 & -31 & 119 & 35 & 2 \\ 0 & 4 & -34 & 117 & 39 \\ 0 & 0 & 9 & -38 & 122 \end{bmatrix}$$

The following matrix gives the non-zero coefficients of the adjustment matrix $A_{32 \times 32}^*$, where the matrix diagonal is highlighted in gray.

$$\begin{bmatrix} 128 & 2 & 0 & 0 & 0 \\ -2 & 128 & 3 & 0 & 0 \\ 0 & -3 & 128 & 4 & 0 \\ 0 & -4 & 128 & 5 & 0 \\ 0 & -5 & 128 & 7 & 0 \\ 0 & -7 & 128 & 8 & 0 \\ 0 & -8 & 127 & 9 & 0 \\ 0 & -9 & 127 & 11 & 1 \\ 0 & -11 & 127 & 12 & 1 \\ 1 & -12 & 127 & 14 & 1 \\ 1 & -14 & 126 & 15 & 1 \\ 1 & -15 & 126 & 16 & 1 \\ 1 & -16 & 126 & 16 & 1 \\ 1 & -16 & 126 & 18 & 1 \\ 1 & -18 & 125 & 20 & 2 \\ 1 & -20 & 125 & 21 & 2 \\ 2 & -21 & 124 & 22 & 2 \\ 2 & -22 & 124 & 23 & 2 \\ 2 & -23 & 123 & 25 & 3 \\ 2 & -25 & 123 & 26 & 3 \\ 3 & -26 & 122 & 27 & 3 \\ 3 & -27 & 122 & 28 & 3 \\ 3 & -28 & 121 & 29 & 4 \\ 3 & -29 & 121 & 30 & 4 \\ 4 & -30 & 120 & 32 & 5 \\ 4 & -32 & 119 & 34 & 5 \\ 5 & -34 & 118 & 34 & 5 \\ 5 & -34 & 118 & 36 & 6 \\ 5 & -36 & 117 & 37 & 6 \\ 6 & -37 & 117 & 36 & 6 \\ 0 & 6 & -36 & 116 & 41 \\ 0 & 0 & 6 & -41 & 121 \end{bmatrix}$$