Data-independent Low-complexity KLT Approximations for Image and Video Coding

A. P. Radünz * T. L. T. da Silveira † F. M. Bayer † R. J. Cintra §

Abstract

The Karhunen-Loève transform (KLT) is often used for data decorrelation and dimensionality reduction. The KLT is able to optimally retain the signal energy in only a few transform components, being mathematically suitable for image and video compression. However, in practice, because of its high computational cost and dependence on the input signal, its application in real-time scenarios is precluded. This work proposes low-computational cost approximations for the KLT. We focus on the blocklengths $N \in \{4, 8, 16, 32\}$ because they are widely employed in image and video coding standards such as JPEG and high efficiency video coding (HEVC). Extensive computational experiments demonstrate the suitability of the proposed low-complexity transforms for image and video compression.

Keywords

Approximate transform, image compression, Karhunen-Loève transform, low-complexity transforms, signed KLT.

1 Introduction

The Karhunen-Loève transform (KLT) [1] is a commonly used tool for data decorrelation and dimensionality reduction [2, 3]. It consists of a linear transformation that maps correlated variables into uncorrelated variables, sometimes referred to as principal components [4]. Usually, only the first coefficients of the transformed data are sufficient to represent the signal. The KLT capability for energy compaction is paramount for data compression, since most information can be preserved even reducing the dimensionality of the data [5]. In fact, considering first-order Markovian processes, the KLT is an optimal linear transform capable of minimizing the mean square error in data compression and concentrating energy in few coefficients of the output signal [1]. Although it is a well-established optimal transform in terms of energy compaction and decorrelation [6], the KLT is not widely applied because its computation depends on the covariance matrix of the input data. Indeed, such data-dependent requirements can hinder the development of fast algorithms for an efficient implementation of the transform.

However, if the input data is a first-order Markov process with known correlation coefficient $p$, then it was shown in [7] that we can derive an analytical solution for the elements of the KLT matrix. Nevertheless, even with the transform matrix known, computational complexity of its implementation can be infeasible for practical data compression scenarios. In this context, several fast approximations for the KLT have been proposed [8–17] aiming at reducing the computational costs. Although such methods generate fast approximations for the KLT, their scope is relatively limited because the data-dependence is still present; in some cases either depending on the covariance matrix of the input data [9, 13, 15, 17] or on the correlation coefficient in case of first-order Markovian signals [8].

When considering first-order Markovian random signals, [18] and [19] have shown that the discrete cosine transform (DCT-II) and the discrete sine transform (DST-I) are asymptotic approximations for the KLT, with the correlation coefficient of the input signal tending to unity and to zero, respectively [18]. Both the DCT and the DST are independent of the input signal, allowing the development of computationally efficient fast algorithms. The DCT is widely adopted in image and video compression standards such as JPEG [21] and high efficiency video coding (HEVC) [22], just to name a few. However, the use of this transform can still be prohibitive in contexts under severe restrictions on processing power or energy autonomy [23–26]. In fact, DCT realizations that require multiplications implemented in floating-point arithmetic-based hardware [27] demand significant circuitry complexity and energy consumption [1]. In this sense, several multiplication-free approximations for the DCT have been proposed [23–25, 28–38], including the signed DCT (SDCT) [39]. The SDCT is derived by applying the signum function to the elements of the DCT matrix, thus resulting in a matrix of trivial multiplicands ($-1, +1$). Therefore, the transform computation requires only additions. Such reduction in the arithmetic cost implies in a lower computational cost, favoring applications in real-time and in low-consumption devices [1].

The present work employs the sign function as a means to obtain computationally efficient alternatives to the KLT for first-order Markov processes. We follow an entirely different approach when compared with the fast KLT approximations already known in the literature. Here, we focus on the proposition of deterministically defined multiplierless low-complexity approximations for the KLT that does not depend on the input signal and is capable of coping with a wide range of correlation coefficients. Our analyses are devoted to the blocklengths $N \in \{4, 8, 16, 32\}$ because of their relevance in image and video standards as JPEG [21] and HEVC [22]. In order to find the best-performing low-cost approximations, we propose a constrained optimization approach according to suitable figures of merit for the KLT analysis. The considered approximation method is specifically tailored to furnish low-complexity transformations appropriate for dedicated highly-efficient circuitry design. The resulting KLT approximations are sought to be numerically evaluated according to coding performance [40, 41], and similarity/proximity metrics [1, 28] with respect to the exact KLT. The obtained transforms are then embedded into (i) a JPEG-like image compression scheme, and (ii) an HEVC reference software for video coding assessment.

To the best of our knowledge, literature lacks KLT approximations that combine the following properties:

1. deterministic definition;
2. suitability for fast algorithm design;
3. data-independence; and

\*Programa de Pós-Graduação em Estatística, Universidade Federal de Pernambuco, Recife, Brazil. E-mail: apr1@de.ufpe.br
†Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil. E-mail: tlt.t.dasilveira@gmail.com
‡Departamento de Estatística and LACESM, Universidade Federal de Santa Maria, Santa Maria, Brazil. E-mail: bayer@ufsm.br
§Signal Processing Group, CCEN, UFPE, Brazil. E-mail: rjdesc@de.ufpe.br
4. capability of processing data at a wide range correlation.

We aim, therefore, at a proposition of a new class of KLT approximations that addresses these gaps. The main goal of our paper is to propose low-complexity approximate transforms for the KLT considering different values of the correlation coefficients ρ, so low and mid-correlated signals could be properly treated as well. Since, to the best of our knowledge, the literature lacks efficient KLT-based methods considering lowly correlated data, there is no competing method for a fair comparison.

This paper is structured as follows. In Section 2 we revise the mathematical formulation of the KLT for first-order Markovian signals and define the general framework for signed KLT (SKLT). Section 3 describes the computational approach for obtaining new transforms and presents these transforms for different lengths attaining optimality according to the proposed figures of merit based on classical metrics. Section 4 also presents fast algorithms for the proposed transforms. In Section 5 we assess the proposed 4-, 8-, 16-, and 32-point SKLT in image and video coding. Section 6 concludes the paper.

2 Signed KLT

2.1 Karhunen-Loève Transform for the First-order Markov Process

The KLT is a linear transformation represented by an orthogonal matrix \( K_N^{(ρ)} \) which decorrelates an input signal \( \mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]^T \) resulting in uncorrelated signal \( \mathbf{y} = [y_0, y_1, \ldots, y_{N-1}]^T \). The \((i,j)\)th elements of the transformation matrix \( K_N^{(ρ)} \), for an arbitrary value of \( ρ \in [0,1] \), are given by [1]

\[
k_{ij} = \frac{2}{N+1} \sin \left( \frac{\omega_j (i-\frac{N-1}{2}) + (j+1)\pi}{2} \right), \quad i,j = 0,1,\ldots,N-1,
\]

(1)

where the eigenvalues of the transformed signal \( y \) covariance matrix are obtained by

\[
\lambda_j = \frac{1 - ρ^2}{1 + ρ^2 - 2ρ \cos \omega_j}, \quad j = 0,1,\ldots,N-1,
\]

(2)

and \( \omega_1, \omega_2, \ldots, \omega_N \) are the \( N \) solutions of the non-linear equation

\[
\tan N\omega = \frac{-(1 - ρ^2) \sin \omega}{(1 + ρ^2) \cos \omega - 2ρ}.
\]

(3)

It is a well-known fact that adjacent pixels from natural images are highly correlated [42], being \( ρ = 0.95 \) a widely adopted assumption [1]. When the correlation of the input signal tends the unity, \( ρ = 1 \), the KLT converges to the DCT [13].

For instance, if \( N = 8 \) and \( ρ = 0.95 \), then the KLT matrix is given by

\[
K_8^{(0.95)} = \begin{bmatrix}
0.338 & 0.351 & 0.360 & 0.364 & 0.360 & 0.351 & 0.338 \\
0.481 & 0.420 & 0.296 & 0.101 & -0.101 & -0.286 & -0.420 & -0.481 \\
0.467 & 0.207 & -0.179 & -0.456 & -0.456 & -0.179 & 0.207 & 0.467 \\
0.423 & -0.085 & -0.487 & -0.278 & 0.278 & 0.487 & 0.085 & -0.423 \\
0.351 & -0.356 & 0.351 & 0.351 & -0.356 & -0.347 & 0.360 & 0.283 \\
0.283 & -0.488 & 0.094 & 0.415 & -0.415 & -0.094 & 0.488 & -0.283 \\
0.198 & -0.462 & 0.460 & -0.190 & -0.190 & 0.460 & -0.462 & 0.198 \\
0.100 & -0.279 & 0.416 & -0.490 & 0.490 & -0.416 & 0.279 & -0.100
\end{bmatrix}
\]

(4)

2.2 KLT Approximations

Our approach is based on the technique used in [39] for proposing the classical signed DCT (SDCT). The proposed transform, as well as the SDCT, is motivated by the reduction of the total number of arithmetic operations required for the computation of the transform at the cost of some accuracy loss [39]. The technique considers the signum function to generate a matrix approximation for the KLT. Thus, we propose the following approximate transformation matrix:

\[
\hat{K}_N^{(ρ)} \approx \frac{1}{\sqrt{N}} \text{sign}(K_N^{(ρ)}),
\]

(5)

where

\[
\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0, \end{cases}
\]

(6)

and \( K_N^{(ρ)} \) is the KLT matrix of order \( N \) with a predefined correlation coefficient \( ρ \), which entries are given by [1]. When applied to a matrix, the signum function operates element-wise.

In other words, we map a given KLT matrix to a low-complexity matrix close to it. Note that, if \( ρ = 0 \), then the Equation (5) degenerates into the null matrix. Also if \( ρ = 1 \), then \( K(1) \) is the DCT and the resulting approximation is the signed DCT [39]. Therefore, in practice, our analysis is constrained to \( 0 < ρ < 1 \). Note that because of the non-linearity and discrete nature of the signum function, different KLT matrices might be mapped to the same approximate matrix. For instance, considering \( ρ_1 = 0.7 \) and \( ρ_2 = 0.9 \), we obtain that

\[
\hat{K}_8^{(ρ_1)} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 \end{bmatrix}.
\]

(7)

Exhaustively computing all KLT matrices in the range \( ρ \in [10^{-3}, 1-10^{-3}] \) in steps of \( 10^{-3} \) returns 999 matrices. However, the number of approximations is much lower: 1, 2, 9, and 37 different approximations for \( N = 4, 8, 16, \) and 32, respectively. In view of the above, a methodology for selecting best-performing approximations is necessary, which is the topic of the next section.

Although the KLT is an orthogonal matrix, the proposed transforms are not constrained to be, therefore, given the proposed transform \( \hat{K}_N^{(ρ)} \), the transformed signal is given by

\[
\mathbf{y} = \hat{K}_N^{(ρ)} \mathbf{x}.
\]

(8)

and the inverse transformation can be written as

\[
\mathbf{x} = (\hat{K}_N^{(ρ)})^{-1} \mathbf{y}.
\]

(9)

3 Optimal SKLT

In this section, we describe an optimization problem, aiming at the identification of best-performing SKLT matrices, according to the figures of merit detailed next.

3.1 Figures of Merit for Approximate Transforms

Approximate transform methods [23, 25, 28, 39] are usually assessed in terms of (i) coding metrics such as the coding gain [40] and transform efficiency [41], which measure the power of decorrelation and energy compression; and (ii) proximity metrics with respect to the exact transform, such as the mean-square error [1] and
3.1.1 Unified Coding Gain

The unified coding gain of a transform $\hat{T}_N$ is given by [43]

$$Cg(\hat{T}_N) = 10 \cdot \log_{10} \left( \prod_{k=1}^{N} \frac{1}{\sqrt{A_k B_k}} \right),$$

where $A_k = \sum (h_k^T \cdot h_k) \odot R_k$, $h_k$ is the $k$th row vector from $\hat{T}_N$, the function $su(\cdot)$ returns the sum of the elements of its matrix argument, $\odot$ is the Hadamard matrix product operator [44], $R_k$ is the autocorrelation matrix of the considered first-order Markovian signal, $B_k = \|h_k\|^2$ and $g_k$ is the $k$th row vector from $\hat{T}_N^{-1}$, and $\|\cdot\|$ is the Frobenius norm [44].

3.1.2 Transform Efficiency

Another coding related figure of merit is the transform efficiency, given by [41]

$$\eta(\hat{T}_N) = \frac{100 \sum_{i=1}^{N} |r_{i,i}|}{\sum_{i=1}^{N} \sum_{j=1}^{N} |r_{i,j}|},$$

where $r_{i,j}$ is the $(i,j)$th element from $\hat{T}_N \cdot R_x \cdot \hat{T}_N^{-1}$.

3.1.3 Mean-Square Error

The mean-square error (MSE) relative to the KLT is given by [1]:

$$\text{MSE}(\hat{T}_N) = \frac{1}{N} \cdot \text{tr} \left\{ (K_{N}^{(\rho)} - \hat{T}_N) \cdot R_x \cdot (K_{N}^{(\rho)} - \hat{T}_N)^T \right\},$$

where tr$(\cdot)$ is the trace function [45].

3.1.4 Total Error Energy

The total error energy of an approximation relative to the KLT is computed by [28]:

$$\epsilon(\hat{T}_N) = \pi \cdot \|K_{N}^{(\rho)} - T_N\|^2.$$

3.1.5 Proposed Figures of Merit

Because the above discussed figures of merit are defined for a fixed value of $\rho$, we propose the following total metrics which take into account the performance for all values of $0 < \rho < 1$:

$$Cg_T(\hat{T}_N) = \int_0^1 (Cg(K_{N}^{(\rho)}) - Cg(T_N)) \, d\rho,$$

$$\eta_T(\hat{T}_N) = \int_0^1 |\eta(K_{N}^{(\rho)}) - \eta(T_N)| \, d\rho,$$

$$\text{MSE}_T(\hat{T}_N) = \int_0^1 \text{MSE}(T_N) \, d\rho,$$

$$\epsilon_T(\hat{T}_N) = \int_0^1 \epsilon(T_N) \, d\rho.$$

3.2 Optimization Problem

In order to identify the overall best-performing approximations, we propose the following optimization problem:

$$\hat{T}_N^* = \arg \min_{0 < \rho < 1} \text{error}(\hat{T}_N^{(\rho)}),$$

where error$(\cdot)$ is one of the proposed measures, $CG_T(\cdot)$, $\eta_T(\cdot)$, $\text{MSE}_T(\cdot)$, $\epsilon_T(\cdot)$, presented in the previous subsection. Note that, for a fixed transform length $N$, up to four optimal SKLT can be obtained, each one optimizing a total metric. Hereafter we denote the optimal transforms of length $N \in \{4, 8, 16, 32\}$ as $\hat{T}_{N,i}$, which are indexed by the subscript $i = 1, 2, \ldots, J$, where $J$ is the number of approximate transforms for $N$.

Table I summarizes the results for the optimal SKLT with different transform lengths $N$ and for the intervals of $\rho$ that each transform is defined. DCT, DST, and the SDCT results are included only for comparison purposes. It is important to emphasize again that this comparison is not completely fair since the proposed transforms cover different intervals of $\rho$ while the DCT and DST are defined for $\rho$ tending merely to one and zero, respectively. All metrics are computed with respect to the exact KLT. The values in bold are the best measurements for each transform length $N$. The transforms $\hat{T}_{4,1}$, $\hat{T}_{8,2}$, and $\hat{T}_{16,3}$ are already known in the literature, and coincide with the SDCT. The remaining transforms are, to the best of our knowledge, new ones.

3.3 Fast Algorithms

The direct implementation of the proposed transforms requires $N(N - 1)$ additions and no multiplications. Besides searching for multiplication-free transforms, the development of fast algorithms capable of reducing the arithmetic cost of computing the transforms is important. Using sparse matrix factorization [46], such as butterfly-based structures, we can derive the factorization for the optimal proposed transforms. In order to assess the complexity of the proposed fast algorithms we considered the number of arithmetic operations needed for its implementation. The arithmetic complexity does not depend on the available architecture or technology, an issue that may occur when considering measures such as computation time [46][49]. The derived matrix factorization for $N = 4$ and $8$ are presented in the following, and for $N = 16$ and $32$, the respective matrix factorization are detailed in Appendix A.

3.3.1 Matrix Factorization for $N = 4$

For $N = 4$, we can factorize $\hat{T}_{4,1}$ [39] as follows:

$$\hat{T}_{4,1} = \frac{1}{2} P_4 \cdot A_{4,2} \cdot A_{4,1},$$

where

$$P_4 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad A_{4,2} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

and

$$A_{4,1} = \begin{bmatrix} I_2 & \tilde{I}_2 \\ I_2 & -\tilde{I}_2 \end{bmatrix},$$

where $I_2$ and $\tilde{I}_2$ are, respectively, the identity and counter-identity matrices of order 2. The minus sign $-$ represents $-1$ and blank spaces are zeroes.
3.3.2 Matrix Factorization for $N = 8$

For $N = 8$, we have:

\[
\hat{T}_{8,1} = \frac{1}{\sqrt{8}} P_8 \hat{A}_{8,3} \hat{A}_{8,2} \hat{A}_{8,1}, \\
\hat{T}_{8,2} = \frac{1}{\sqrt{8}} P_8 \hat{A}_{8,3}'' \hat{A}_{8,2} \hat{A}_{8,1},
\]

where

\[
P_8 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix},
\]

\[
A''_{8,3} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & 1 \end{bmatrix},
\]

\[
A_{8,2} = \begin{bmatrix} A_{4,1} \\ A_{4,1} \end{bmatrix}, \\
A_{8,1} = \begin{bmatrix} I_4 & I_4 \\ I_4 & -I_4 \end{bmatrix}.
\]

Table 2 presents the arithmetic cost of the proposed fast algorithms for the approximate transforms compared with the arithmetic cost of the direct implementation of the exact $N$-point KLT. Since the proposed transforms are, to the best of our knowledge, the first class of approximations for the KLT following this approach, there is nothing to compare with. However, to show that the performance of the proposed transforms is competitive, we introduced in Table 3 the arithmetic cost of some transforms in general that are already known in the literature. Thus, Table 3 presents the arithmetical cost of: the fast algorithms proposed by [49] and [50] for the 8- and 16-point DCT, respectively; and approximations for the DCT already known in the literature proposed in [25,28,31,33,39,51,56]. We can note that the introduced fast algorithms are multiplierless and offer substantial reductions in the additive complexity when compared with the exact KLT and lower or similar arithmetic cost compared with the DCT approximations.

For a better visualization of the results presented in Tables 4 and 5, we have combined graphically the number of additions and the proposed figures of merit of each proposed transform, for $N = 4$, $8$, $16$, and $32$, as presented in Figures 4, 5, 6, respectively. Note that both DCT and DST also require multiplications when imple-
Table 3: Computational cost comparison for 8-, 16-, and 32-point DCT approximation transforms

| Transform | Additions | Multiplications | Bit-Shifting | Addition reduction (%) |
|-----------|-----------|-----------------|--------------|------------------------|
| T_{4,1}   | 4         | 0               | 0            | 66%                    |
| K_4       | 12        | 16              | 0            | -                      |

| T_{8,1}   | 24        | 0               | 0            | 57%                    |
| T_{8,2}   | 24        | 0               | 0            | 57%                    |
| K_8       | 56        | 64              | 0            | -                      |

| T_{16,1}  | 80        | 0               | 0            | 67%                    |
| T_{16,2}  | 75        | 0               | 0            | 69%                    |
| T_{16,3}  | 72        | 0               | 0            | 70%                    |
| K_{16}    | 240       | 256             | 0            | -                      |

| T_{32,1}  | 288       | 0               | 0            | 71%                    |
| T_{32,2}  | 288       | 0               | 0            | 71%                    |
| T_{32,3}  | 288       | 0               | 0            | 71%                    |
| T_{32,4}  | 232       | 0               | 0            | 77%                    |
| K_{32}    | 992       | 1024            | 0            | -                      |

4 Image and Video Coding

We submitted the proposed transforms to two different contexts that are classical in the approximation community area: (i) still image compression according to a JPEG-like algorithm [23, 25, 28] and (ii) video encoding as defined in the HEVC reference software [55]. In this section, we compared the proposed transforms with the exact KLT for \( \rho = 0.1, 0.6 \) and 0.9 (\( K^{(0.1)}, K^{(0.6)} \) and \( K^{(0.9)} \)), and with the exact DCT. The transforms \( K^{(0.1)}, K^{(0.6)} \), and \( K^{(0.9)} \) were selected because they are suitable for decorrelation of lowly, moderately, and highly correlated data, respectively. Currently literature is restricted to the high correlation scenario, which is mainly addressed by the DCT and its related approximations.

4.1 Image Compression

In this section, we evaluated the performance of the proposed transforms in image compression, similarly to [23, 25, 28]. If \( A \) is a two-dimensional (2D) image, then the direct and inverse transformations induced by the SKLT are computed, respectively, by

\[
B = T_{N}^{(0)} \cdot A \cdot (T_{N}^{(0)})^{-1},
\]

(12)

\[
A = (T_{N}^{(0)})^{-1} \cdot B \cdot T_{N}^{(0)}.
\]

(13)

The adopted compression scheme is described as follows [59]: (i) the image is divided into disjoint sub-blocks \( A_j \) of size \( N \times N \); (ii) each sub-block is submitted to a selected transform \( T_N \) according to (12); (iii) using the standard zig-zag sequence [59], only the initial \( r \) coefficients in each sub-block \( B_j \) are retained and the remaining \( N^2 - r \) coefficients are zeroed, resulting in sub-block \( \tilde{B}_j \); (iv) the two-dimensional inverse transform is applied according (13), and (v) the reconstructed sub-blocks \( \tilde{A}_j \) are adequately rearranged. The final reconstructed image \( \tilde{A} \) is compared with the original image \( A \) for assessing the performance of \( T_N \). We adopted \( N \in \{4, 8, 16, 32\} \) and used the peak signal-to-noise ratio (PSNR) [60], and the mean.
structural similarity index (MSSIM) [61] as figures of merit for image quality evaluation. The results were taken individually for 45 512 × 512 8-bit greyscale images obtained from [62] and averaged.

For each transform length, we considered two approaches: (i) a qualitative analysis, based on the compressed Lena image with approximately 85% compression and (ii) a quantitative one, varying the value of r for compression, considering the average MSSIM and PSNR values of the compressed images.

Figure 5 shows the original Lena image. The reconstructed images using the proposed transforms, DCT, and DST for N = 4, 8, 16, and 32 are presented, respectively, in Figures 5–7, 8, and 9. The corresponding compression ratio (CR) was CR = 81.25%, 84.38%, 84.86%, respectively. Visually, the reconstructed images after compression exhibit quality comparable with the original image.

Figure 10 presents the average image quality measurements for different values of N considering different levels of compression, comparing with the exact KLT for ρ = 0.1, 0.6, and 0.9 and the DCT. Figure 10(b) shows that the average MSSIM of T 4,1 is very close to the results from the KLT and DCT. Figure 10(a) also shows that the values of the average PSNR of T 4,1 are close to the values obtained by KLT and DCT. We can also notice that the approximation T 4,1 presents average MSSIM values better than KLT itself when we retain fewer coefficients, r ranging from zero to six. For N = 8, one can see that T 8,1 and T 8,2 behave in a similar way according to the image quality measures. For N = 16, T 16,3 transform has considerably closer values to the DCT and KLT than the other approximations. This may be related to the fact that T 16,3 is the obtained transform with a higher value of ρ (ρ > 0.9). One can see that, for r < 10, the approximation T 16,3 presents better average MSSIM values than KLT itself. Figures 10(g) and 10(h) present the image quality measurements for N = 32, that poses T 32,4 as the best-performing approximate KLT for compression.

4.2 Video coding

In order to demonstrate the suitability of the introduced SKLTS in the video coding context, we embed the proposed approximations into a public available HEVC reference software [68]. The HEVC employs an integer DCT (IDCT) of lengths 4, 8, 16, and 32 [69], unlike its predecessors [64,67]. According to [22], the larger transforms generally work better for smooth image regions, whereas the textured areas are better handled by the small sized transforms.

For our experiment, we substituted the original set of IDCTs natively defined in the HEVC standard by our proposed KLT approximations. The original integer DST-VI [68] of length 4, responsible for residual coding in HEVC, is kept unchanged in the reference software. The main reason is that the optimal SKLT for length 4 is unique for all ρ values (confer Table 1), and it approaches the coding capabilities of the DCT [89]. We separated four suites of approximations, relating to the optimality in Equation (11): (i) T 4,1, T 8,1, T 16,2, and T 32,2 (Group I); (ii) T 4,1, T 8,1, T 16,1, and T 32,3 (Group II); (iii) T 4,1, T 8,2, T 16,3, and T 32,1 (Group III); and (iv) T 4,1, T 8,1, T 16,3, and T 32,4 (Group IV). Namely, Groups I, II, III, and IV are optimal regarding total MSE, total error energy, total unified coding gain, and total transform efficiency, respectively. Therefore, we substituted the original IDCT by each SKLT Group I–IV in the HEVC reference software.

In our experiments, we encoded the first 100 frames of each video sequence of each A to F class in accordance with the Common Test Conditions (CTC) document [69]. The considered 8-bit standard video sequences were: PeopleOnStreet (2560×1600 at 30 fps), BasketballDrive (1920×1080 at 50 fps), RaceHorses (832×480 at 30 fps), BlowingBubbles (416×240 at 50 fps), KristenAndSara (1280×720 at 60 fps), and BasketballDrill11Text (832×480 at 50 fps). We further considered the Foreman (352×288 at 30 fps) [70], a standard 8-bit CIF video sequence adopted in related works like [81,83]. As done in [80], we set all the test parameters in accordance with the CTC documentation. Also, we considered the four standard 8-bit coding configurations in the Main profile: All Intra (AI), Random Access (RA) and Low-Delay B and P (LD-B and LD-P). We selected the frame-wise PSNR for each YUV color channel [63] as figure of merit. Then, for each video sequence, we computed the rate distortion (RD) curve considering quantization parameter (QP) values equal to 22, 27, 32, and 37 [69].

Moreover, we measured the Bjøntegaard’s delta PSNR (BD-PSNR) [71,72] for the modified versions of the HEVC software. The average results per video for all the transform groups and coding
configurations are presented in Table 4. One can note from Table 4 that the Group IV performed better than the Group III on average. The metrics used for selecting both SKLT Groups III and IV maximize the coding efficiency of the transforms. From the table, Group IV outperformed Groups I and II in most of the cases regardless the configuration mode. This information can be confirmed in Figures 11, 12, 13, and 14, which show the RD curves for the four groups of transforms in AI, RA, LD-B, and LD-P configurations.

One can notice that the group of transforms that optimize the total transform efficiency metric (Group IV) tended to outperform the other three groups (Groups I, II, and III). The results in video coding corroborate those of the still-image experiments presented in Section 4.1.

As a qualitative example, we present in Figure 15 the tenth frame of the KristenAndSara video encoded according to the default HEVC IDCT and the transforms in Groups I–IV in AI configuration. The presented metrics are a representation from the performance obtained by the proposed transforms on the other video sequences as well. Here, QP value was set to 32. Blocking artifacts are not easily perceptible, highlighting the applicability of the proposed SKLT.

5 Conclusions

Based on the signum function, we proposed a class of low-complexity KLT approximations, which are suitable for data decorrelation. These transforms are deterministically defined for processing data at a wide range correlation, which differs from the fast approximations for the KLT already known in the literature. Since the proposed transforms are the first approximations for the KLT following this approach, there is nothing to compare with. In principle, all approximations derived from this method are novelty. In particular, we explicitly derived new transforms of length 4, 8, 16, and 32 and submitted them to a comprehensive assessment in the context of image and video coding. Total figures of merit were proposed for the selection of optimal transforms. The proposed approximations were tailored to decorrelate Markovian first-order data, at a very low arithmetic complexity and multiplierless operation. Besides the lower complexity, we also derived fast algorithms for the proposed transforms that were capable of reducing, even more, the arithmetic cost of its implementation. The proposed SKLT showed good compaction energy properties at a very low cost. Still image and video experiments demonstrate the suitability of the proposed approximations for image/video encoding, being capable of generating high quality images according to coding and similarity metrics. For future works, we wish to consider larger blocklengths transforms and other applications for fully exploring the potential of the proposed transforms, such as the versatile video coding (VVC) standard [73–75], for example.

A Matrix Factorization

A.1 $N = 16$

For $N = 16$, we can rewrite $\mathbf{T}_{16,1}$, $\mathbf{T}_{16,2}$, and $\mathbf{T}_{16,3}$ as follows:

$$
\mathbf{T}_{16,1} = \frac{1}{4} \cdot \mathbf{P}_{16} \cdot \mathbf{A}_{16,3} \cdot \mathbf{A}_{16,2} \cdot \mathbf{A}_{16,1},
$$

$$
\mathbf{T}_{16,2} = \frac{1}{4} \cdot \mathbf{P}_{16} \cdot \mathbf{A}_{16,5} \cdot \mathbf{A}_{16,4} \cdot \mathbf{A}_{16,3} \cdot \mathbf{A}_{16,2} \cdot \mathbf{A}_{16,1},
$$

$$
\mathbf{T}_{16,3} = \frac{1}{4} \cdot \mathbf{P}_{16} \cdot \mathbf{A}_{16,5} \cdot \mathbf{A}_{16,4} \cdot \mathbf{A}_{16,3} \cdot \mathbf{A}_{16,2} \cdot \mathbf{A}_{16,1},
$$

where

$$
\mathbf{P}_{16} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 
\end{bmatrix},
$$

$$
\mathbf{A}_{16,1} = \begin{bmatrix}
\mathbf{I}_8 & \mathbf{I}_8 \\
\mathbf{I}_8 & -\mathbf{I}_8
\end{bmatrix}, \quad \mathbf{A}_{16,2} = \begin{bmatrix}
\mathbf{A}_{8,1} \\
\mathbf{A}_{8,1}
\end{bmatrix}.
$$
Figure 3: Arithmetic complexity versus the proposed figures of merit for each transform for \( N = 16 \).

\[
A_{16,3} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1
\end{bmatrix},
A'_{16,4} = \begin{bmatrix}
1 & 1 \\
1 & -1 \\
1 & 1 \\
1 & 1
\end{bmatrix},
A''_{16,5} = \begin{bmatrix}
1 & 1 \\
1 & 1 \\
1 & -1 \\
1 & -1
\end{bmatrix},
A''_{16,3} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1
\end{bmatrix},
A_{16,5} = \begin{bmatrix}
1 & 1 \\
1 & -1 \\
1 & 1 \\
1 & 1
\end{bmatrix},
A'_{16,6} = \begin{bmatrix}
1 & 1 \\
1 & -1 \\
1 & 1 \\
1 & 1
\end{bmatrix},
A''_{16,4} = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 \\
1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1
\end{bmatrix},
A''_{16,5} = \begin{bmatrix}
1 & 1 \\
1 & 1 \\
1 & -1 \\
1 & -1
\end{bmatrix}.
\]

A.2 \( N = 32 \)

Considering \( N = 32 \), we have:

\[
T_{32,1} = \frac{1}{\sqrt{32}} P_{32} \cdot A_{32,3} \cdot A_{32,2} \cdot A_{32,1},
T_{32,2} = \frac{1}{\sqrt{32}} P_{32} \cdot A_{32,3} \cdot A_{32,2} \cdot A_{32,1},
T_{32,3} = \frac{1}{\sqrt{32}} P_{32} \cdot A_{32,3} \cdot A_{32,2} \cdot A_{32,1},
T_{32,4} = \frac{1}{\sqrt{32}} P_{32} \cdot A_{32,6} \cdot A_{32,5} \cdot A_{32,4} \cdot A_{32,3} \cdot A_{32,2} \cdot A_{32,1}.
\]
Figure 4: Arithmetic complexity versus the proposed figures of merit for each transform for $N = 32$.

Figure 5: Original *Lena* image.

where

$P_{32} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$,

$A_{32,1} = \begin{bmatrix}
I_{16} & I_{16} \\
I_{16} & -I_{16}
\end{bmatrix}$,

$A_{32,2} = \begin{bmatrix}
A_{16,1} & A'_{32,3}
\end{bmatrix}$.
Figure 6: Compressed Lena images for $N = 4$ and $r = 3$. 

Figure 7: Compressed Lena images for $N = 8$ and $r = 10$. 

$$A''_{32,3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}$$

$$A''_{32,3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}$$
Figure 8: Compressed Lena images for $N = 16$ and $r = 40$.

and

$$A_{32,6}' = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
\end{bmatrix}.$$ 

Acknowledgments

We gratefully acknowledge partial financial support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Fundação de Amparo a Ciência e Tecnologia de Pernambuco (FACEPE), Brazil.
Figure 9: Compressed Lena images for N = 32 and r = 155.

Table 4: Average BD-PSNR of the modified HEVC reference software for tested video sequences

| Configuration | Video sequence      | Transforms | Group I | Group II | Group III | Group IV |
|---------------|---------------------|------------|---------|----------|-----------|----------|
| AI            | PeopleOnStreet      | -0.6074    | -0.6127 | -0.5696  | -0.5774   |          |
|               | BasketballDrive     | -0.6194    | -0.6434 | -0.4549  | -0.4452   |          |
|               | RaceHorses          | -0.8221    | -0.8219 | -0.8280  | -0.8290   |          |
|               | KristenAndSara      | -0.3446    | -0.3464 | -0.3142  | -0.3361   |          |
|               | BlowningBubbles     | -0.7275    | -0.7425 | -0.5495  | -0.5391   |          |
|               | BasketballDrillText | -0.3163    | -0.3217 | -0.2680  | -0.2844   |          |
|               | Foreman             | -0.3030    | -0.3075 | -0.2636  | -0.2794   |          |
| RA            | PeopleOnStreet      | -0.3995    | -0.4030 | -0.3748  | -0.3797   |          |
|               | BasketballDrive     | -0.4855    | -0.5054 | -0.3629  | -0.3423   |          |
|               | RaceHorses          | -1.2096    | -1.2150 | -1.1687  | -1.1435   |          |
|               | BlowningBubbles     | -0.2784    | -0.2849 | -0.2440  | -0.2525   |          |
|               | BasketballDrillText | -0.4108    | -0.4095 | -0.3184  | -0.3241   |          |
|               | Foreman             | -0.2532    | -0.2561 | -0.2127  | -0.2290   |          |
| LD-B          | BasketballDrive     | -0.4900    | -0.5089 | -0.3548  | -0.3117   |          |
|               | RaceHorses          | -1.1279    | -1.1342 | -1.1324  | -1.1028   |          |
|               | BlowningBubbles     | -0.2978    | -0.3150 | -0.2519  | -0.2610   |          |
|               | KristenAndSara      | -0.4069    | -0.4245 | -0.2958  | -0.2740   |          |
|               | BasketballDrillText | -0.4609    | -0.4629 | -0.3616  | -0.3528   |          |
|               | Foreman             | -0.3147    | -0.2988 | -0.2421  | -0.2436   |          |
| LD-P          | BasketballDrive     | -0.4932    | -0.5139 | -0.3542  | -0.3153   |          |
|               | RaceHorses          | -1.0903    | -1.0940 | -1.0929  | -1.0637   |          |
|               | BlowningBubbles     | -0.2905    | -0.2958 | -0.2438  | -0.2404   |          |
|               | KristenAndSara      | -0.3889    | -0.4062 | -0.2791  | -0.2544   |          |
|               | BasketballDrillText | -0.4317    | -0.4369 | -0.3400  | -0.3361   |          |
|               | Foreman             | -0.2828    | -0.2927 | -0.2308  | -0.2381   |          |
Figure 10: Image quality measurements for different levels of compression.
Figure 11: RD curves of the modified HEVC versions for test sequences in Main profile and AI configuration: (a) PeopleOnStreet, (b) BasketballDrive, (c) RaceHorses, (d) BlowingBubbles, (e) KristenAndSara, (f) BasketballDrillText, and (g) Foreman.
Figure 12: RD curves of the modified HEVC versions for test sequences in Main profile and RA configuration: (a) PeopleOnStreet, (b) BasketballDrive, (c) RaceHorses, (d) BlowingBubbles, (e) BasketballDrillText, and (f) Foreman.

Figure 13: RD curves of the modified HEVC versions for test sequences in Main profile and LD-B configuration: (a) BasketballDrive, (b) RaceHorses, (c) BlowingBubbles, (d) KristenAndSara, (e) BasketballDrillText, and (f) Foreman.
Figure 14: RD curves of the modified HEVC versions for test sequences in Main profile and LD-P configuration: (a) BasketballDrive, (b) RaceHorses, (c) BlowingBubbles, (d) KristenAndSara, (e) BasketballDrillText, and (f) Foreman.

Figure 15: Compression of the tenth frame of KristenAndSara using the default and modified versions of the HEVC software in configured to the AI mode and QP = 32. Results for the IDCT are shown in (a), and for the SKLT Groups I to IV in (b) to (e), respectively.
