UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

ANABETH PETRY RADÜNZ

LOW-COMPLEXITY APPROXIMATIONS FOR DISCRETE TRANSFORMS: DESIGN, FAST ALGORITHMS, IMAGE CODING, AND USE AS A TOOL IN STATISTICAL INFERENCE

Recife
2023

ANABETH PETRY RADÜNZ

LOW-COMPLEXITY APPROXIMATIONS FOR DISCRETE TRANSFORMS: DESIGN, FAST ALGORITHMS, IMAGE CODING, AND USE AS A TOOL IN STATISTICAL INFERENCE

Tese apresentada ao Programa de Pós-Graduação em Estatística do Centro de Ciências Exatas e da Natureza da Universidade Federal de Pernambuco, como requisito parcial à obtenção do título de Doutor em Estatística. Área de Concentração: Estatística aplicada

Orientador: Dr. Renato José de Sobral Cintra

Co-Orientador: Dr. Fábio Mariano Bayer

Recife

2023

Catalogação na fonte
Bibliotecária Nataly Soares Leite Moro, CRB4-1722

R132l Radünz, Anabeth Petry
Low-complexity approximations for discrete transforms: design, fast algorithms, image coding, and use as a tool in statistical inference / Anabeth Petry Radünz. – 2023.
173 f.: il., fig.

Orientador: Renato José de Sobral Cintra.
Tese (Doutorado) – Universidade Federal de Pernambuco. CCEN, Estatística, Recife, 2023.
Inclui referências e apêndices.

1. Estatística aplicada. 2. Transformadas discretas. 3. Transformadas aproximadas de baixa complexidade. 4. Compressão de imagens. I. Cintra, Renato José de Sobral (orientador). II. Título.

310 CDD (23. ed.) UFPE- CCEN 2023 - 101

ANABETH PETRY RADÜNZ

Low-complexity Approximations for Discrete Transforms: Design, Fast Algorithms, Image Coding, and Use as a Tool in Statistical Inference

Tese apresentada ao Programa de Pós-Graduação em Estatística da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutora em Estatística.

Aprovado em: 31 de março de 2023.

BANCA EXAMINADORA

Prof. Dr. Renato José de Sobral Cintra
Presidente

Prof. Dr. André Leite Wanderley
Examinador Externo ao Programa

Prof. Dr. Ricardo Menezes Campello de Souza
Examinador Externo ao Programa

Prof. Dr. Fábio Mariano Bayer
Examinador Externo à Instituição

Prof. Dr. Thiago Lopes Trugillo da Silveira
Examinador Externo à Instituição

Dr. Diego Felipe Gomes Coelho
Examinador Externo à Instituição

To Eduardo, with love.

ACKNOWLEDGEMENTS

To Eduardo, for the complicity, affection, and love. For being by my side in every moment, both the challenging and joyful ones, supporting and encouraging me always. You have made this journey a little easier and more meaningful.

To my beloved family, who has been my constant source of love, encouragement, and support throughout my doctoral journey. Their unwavering belief in me and their sacrifices have made this achievement possible. I am forever grateful to my parents, Roberto and Ana Cristina, for instilling in me the values of perseverance and lifelong learning. To my sister Amanda, for all the heartfelt conversations and for sharing both tears and laughter. Your journey is just as important, and I know you will achieve great things too.

I dedicate this thesis to my esteemed doctoral advisors, Prof. Renato and Prof. Fábio, whose guidance, expertise, and mentorship have shaped me as a researcher and scholar. Their insightful feedback, constructive criticism, and continuous encouragement have played a pivotal role in shaping the outcome of this research. You have inspired me to push boundaries and strive for greatness. Thank you!

To my dear friends and colleagues from the PPGE, for the countless afternoons of study that seemed never-ending. The journey was made easier with you by my side. César, Adenice, Eduardo, Luan, and Bruna, thank you for the partnership, laughter, and the special moments we shared in the doctoral office.

To Cátia and Cristine, you are important to me and you know it. Thank you for being by my side and listening to me whenever necessary.

Also, to the friends from my undergraduate years and high school, for always being there for me even from a distance.

To Prof. Arjuna and my colleagues from FIU, I am deeply grateful for the invaluable time we spent together, during which I learned an incredible amount in such a short period. The journey was undoubtedly challenging, yet it played an instrumental role in fostering my personal and academic growth.

I also express my gratitude to all the teachers who have crossed my path in my entire life. Undoubtedly, I wouldn't be here if it weren't for some of them.

Lastly, I dedicate this work to all those who have devoted their lives to the pursuit of knowledge and the improvement of society. It is not an easy journey, and I take great pride in being a part of it. May this thesis contribute, albeit in a small way, to the collective progress of our field and

serve as a testament to our shared commitment to learning and discovery.

Finally, I would like to thank CAPES and FACEPE for financial support.

ABSTRACT

Discrete transforms play an important role in the context of signal processing. They are pivotal tools because they allow us to analyze and interpret data in the domain of transforms, which often reveal useful patterns. In particular, we can mention the discrete Fourier transform (DFT), the Karhunen-Loève transform (KLT) and the discrete cosine transform (DCT) as the most relevant transforms in the context of signal and image processing. Although the relevance of using these transforms has been widely corroborated in several studies, the computational costs required for their implementations can become prohibitive in contexts where we have large amounts of data and/or demand for low-complexity devices. In this context, fast algorithms can be a solution for the reduction of arithmetic operations necessary for computing the transforms. However, it is still necessary to deal with the floating-point arithmetic. Thus, several low-complexity transform approximations have been developed, as a low-cost alternative for computing these transforms. This thesis is divided into two parts. In the first part, we propose several classes of low complexity approximations for the KLT and the DCT, fast algorithms, and demonstrate their usability in the context of image processing. In the second part of the thesis, we present approximation classes for the DFT and their applicability in problems of statistical inference, as in the context of signal detection. From the results obtained, we can conclude that the low complexity approximations for the transforms can be considered excellent alternatives in contexts where there is a massive amount of data to be processed or in the case of implementation in low-consumption hardware.

Keywords: discrete transforms; low-complexity discrete transforms; image compression; low-complexity parameter estimation.

RESUMO

Transformadas discretas desempenham um papel importante no contexto de processamento de sinais. Elas são ferramentas pivotais pois permitem analisar e interpretar dados no domínio das transformadas, que frequentemente revelam padrões úteis. Em particular, podemos citar a transformada discreta de Fourier (DFT), a transformada de Karhunen-Loève (KLT) e a transformada discreta do cosseno (DCT) como as transformadas mais relevantes no contexto de processamento de sinais e imagens. Embora a relevância do uso dessas transformadas tenha sido amplamente corroborado em diversos estudos, os custos computacionais necessários para suas implementações podem se tornar proibitivos em contextos em que há grande quantidade de dados e/ou a demanda por dispositivos de baixa complexidade. Nesse sentido, algoritmos rápidos podem ser uma solução para a redução das operações aritméticas necessárias para a computação das transformadas. Porém, ainda é preciso lidar com a aritmética de ponto flutuante. Dessa forma, diversas aproximações matriciais de baixa complexidade vêm sendo propostas, como sendo uma alternativa de baixo custo para o cômputo destas transformadas. A presente tese está dividida em duas partes. Na primeira parte, propomos diversas classes de aproximações de baixa complexidade para a KLT e para a DCT, algoritmos rápidos, e demonstramos sua usabilidade no contexto de processamento de imagens. Na segunda parte da tese, apresentamos classes de aproximação para a DFT e sua aplicabilidade em problemas de inferência estatística, como no contexto de detecção de sinais. Dos resultados obtidos, podemos concluir que as aproximações de baixa complexidade para as transformadas podem ser consideradas excelentes alternativas em contextos em que há uma quantidade massiva de dados a ser processada ou no caso de implementação em hardware de baixo consumo.

Palavras-chave: transformadas discretas; transformadas aproximadas de baixa complexidade; compressão de imagens; estimação de parâmetros de baixa complexidade.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ABREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
APE	Absolute Percentage Error
BAS	Bouguezel–Ahmad–Swamy
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DHT	Discrete Hartley Transform
DST	Discrete Sine Transform
DTT	Discrete Tchebichef Transform
FF	Flip-Flop
FFT	Fast Fourier Transforms
FPGA	Field-Programmable Gate Array
HEVC	High Efficiency Video Coding
IDCT	Integer Discrete Cosine Transform
IoT	Internet of Things
JAM	Jridi-Alfalou-Meher
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loève Transform
LUT	Look-Up Tables
MLE	Maximum Likelihood Estimator
MPEG	Moving Picture Experts Group
MSE	Mean Square Error
MSSIM	Mean Structural Similarity Index
PC	Personal Computer
PSNR	Peak Signal-to-Noise Ratio
RDCT	Rounded Discrete Cosine Transform
RKLT	Rounded Karhunen-Loève Transform
SDCT	Signed Discrete Cosine Transform
SFG	Signal Flow Graphs
SKLT	Signed Karhunen-Loève Transform
UART	Universal Asynchronous Receiver-Transmitter
VLSI	Very Large-Scale Integration

VVC Versatile Video Coding

WHT Walsh-Hadamard Transform

LIST OF SYMBOLS

ρ	Correlation coefficient
N	Matrix dimension
$\mathbf{K}_N^{(\rho)}$	KLT matrix for a given value of ρ and N
$\mathbf{R}_\mathbf{x}$	Covariance matrix from $\mathbf{x}$
$\widehat{\mathbf{T}}$	Approximate matrix
$\mathrm{Cg}(\cdot)$	Coding gain
$\mathrm{su}(\cdot)$	Return the sum of the elements of its matrix argument
$\odot$	Hadamard matrix product operator
$\|\cdot\|$	Frobenius norm
$\eta(\cdot)$	Transform efficiency
$\mathrm{MSE}(\cdot)$	Mean-square error
$\mathrm{tr}(\cdot)$	Return the trace of a square matrix
$\varepsilon(\cdot)$	Total error energy
$\mathbf{I}_N$	N-dimensional identity matrix
$\bar{\mathbf{I}}_N$	N-dimensional counter-identity matrix
$\mathbf{A}$	Arbitrary matrix
$\mathrm{diag}(\cdot)$	Return the diagonal matrix
$\top$	Matrix transpose
$\delta(\cdot)$	Deviation from orthogonality
$\mathrm{int}(\cdot)$	Integer round-off function
$\mathbf{C}_N$	Definition of the N-dimensional DCT matrix
$\mathrm{abs}(\cdot)$	Return the absolute value
$\circ$	Element wise multiplication
$\mathbf{F}_N$	Definition of the N-dimensional DFT matrix
$\Re(\cdot)$	Real part from a complex number
$\Im(\cdot)$	Imaginary part from a complex number

CONTENTS

1 INTRODUCTION

In this chapter, we provide motivation and context for the research topics investigated in this thesis. Additionally, we detail the main goals and the document structure.

1.1 MOTIVATION

With the advent of digital computation, a huge amount of data is generated every day at an unprecedented rate from the most different sources, including science, engineering, health, government, and social networks [1, 2]. The development of Big Data techniques to handle a large amount of information has become increasingly relevant in the past few years [3, 4], even more with technological trends such as the Internet of Things (IoT) [5, 6] and Artificial Intelligence (AI) [2, 7–9]. Statistical tools have been developed to analyze this type of data, such as machine learning and data mining. However, there are few advances in the development of inferential procedures at a low computational cost [10]. With the increasing demand for analyzing large masses of data in real-time, the development of efficient, low-power, and low-complexity methods for statistical tools becomes more necessary every day.

In the context of digital signal processing, many important discrete transforms have become useful tools for signal coding and data decorrelation [11–16] such as the DFT [17], the Discrete Hartley Transform (DHT) [18], the Walsh-Hadamard Transform (WHT) [19], the Discrete Tchebichef Transform (DTT) [20], the KLT [21, 22], the DCT [13], among others. These transforms play an important role in the context of signal processing because they are suitable in applications to real and complex data when using digital computers, which are only capable of numerical and finite calculations [23]. By applying the discrete transforms to the data, we are capable of analyzing the data from the transform domain, which can give us more information about them [11, 13, 24]. In this thesis, emphasis will be given to three discrete transforms: the KLT, the DCT, and the DFT.

The KLT is the optimal linear transform capable of minimizing the mean square error in data compression and concentrating higher energy in a few coefficients of the output signal [13]. Indeed, the KLT completely decorrelates the signal in the transform domain, being the optimal discrete transform in terms of data decorrelation and dimensionality reduction [21, 22, 25]. Because of its excellent performance for data compression, the KLT can be applied in several image and video compression patterns, such as Joint Photographic Experts Group (JPEG) [26], Moving Picture Experts Group (MPEG) [27], and HEVC [28].

Even with such good properties, the use of the KLT can be limited because its definition depends on the variance and covariance matrix of the input data, which can be a problem in contexts with low processing power. The DCT is the asymptotic approximation to the KLT when the input signal correlation tends to unity [13] and when the signal is a first-order Markovian process. The mathematical definition of the DCT is independent of the input signal, which allows the development of computationally efficient fast algorithms.

As much as the KLT and the DCT, the DFT is a very useful tool in the digital signal processing context [17, 29]. The transform decomposes a signal into the frequency of its components, enabling us to analyze, manipulate, and synthesize signals. It is used in the most different contexts, such as solving differential equations [30], image processing [31], beamforming [32, 33], analysis of radar signals [34], voice processing [35], time series [36], spectral estimation [37], among others [38]. In this thesis, we will be giving emphasis on the context where the DFT is used for the detection problem, where we aim to determine whether a radar system is detecting a signal or not [39, 40].

Although the relevance of the use of the discrete transforms has been corroborated in many works in this research field [11, 13, 14, 17, 38], their computational cost can be considered high in contexts where there is a severe shortage of energy resources. One alternative for reducing the computational cost of the transform can be the use of fast algorithms. An algorithm can be defined as a finite sequence of rules, which given a set of inputs, returns an output [23]. Fast algorithms are computational procedures or methods that are specifically designed to process inputs quickly and efficiently. Their primary objective is to solve tasks or problems using the minimum number of necessary calculations. In the past decades, many works were developed for proposing efficient fast algorithms for the KLT [25, 41–44], the DCT [45–63], and the DFT [64–70]. The fast algorithms proposed for the DFT are collectively called as Fast Fourier Transforms (FFT).

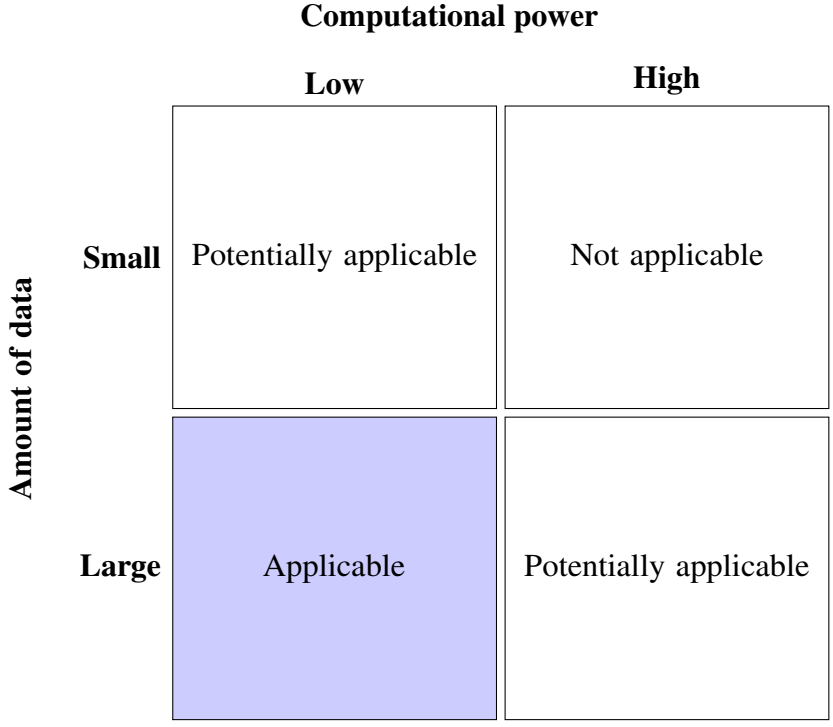
Despite the fact that fast algorithms can highly reduce the computational cost of the transforms, their use is still prohibited in contexts where we have a huge amount of data and/or a demand for low-complexity devices. This is because the transforms, in their exact form, require multiplications by irrationals numbers, which are usually approximated by rounding or truncating in rational form and implemented in floating-point arithmetic or fixed-point arithmetic with large integers [13, 23]. Methods of integer approximations to the transforms have been proposed to eliminate the floating-point operations required for implementing the exact transforms. The low-complexity approximations have proven to be excellent alternatives to their exact counterparts

due to their ability to provide transform matrices with similar mathematical behavior while significantly reducing the arithmetic cost involved. [14].

Particularly, we can cite the low-complexity approximations proposed for the DCT, such as the Signed Discrete Cosine Transform (SDCT) [71], the Rounded Discrete Cosine Transform (RDCT) [72], the DCT approximations based on integer functions [73], the Bouguezel–Ahmad–Swamy (BAS) series of DCT approximations [74–79], and the DCT approximation based on angle similarity [80], among others [81–93]. Considering the KLT, we can also find low-complexity approximations in the literature [94–99]. However, these methods still suffer the problem of data-dependency. For the DFT, we can cite multiplierless approximations, such as the ones proposed by Ariyarathna *et al.* [100], Kulasekera *et al.* [101], Suarez *et al.* [102], Coutinho *et al.* [103], Madanayake *et al.* [33], and more [104–107].

Approximate transforms can be useful in contexts where there is a massive amount of data to be processed or in the case of implementation on low-power hardware. For example, unless we have the high computational power and a small amount of data – where the use of approximations is not necessary, in all other cases might be convenient to use approximate transforms. Figure 1 illustrates this situation.

Figure 1 – Contexts where the applicability of approximate transforms is justified.



Source: Author (2023).

1.2 MAIN GOALS

Given the previous discussion of the relevance of low-complexity transforms in the context of signal processing the main goal of the proposed research in this thesis is to **develop low-complexity methods for statistical tools**. Specifically, we aim at:

1. proposing low-complexity KLT approximations based on integer functions;
2. proposing low-complexity approximations for the DCT for larger blocklengths based on angle similarity;
3. developing fast algorithms for the proposed approximations;
4. assessing the performance of the proposed approximations considering specific figures of merit;
5. assessing the quality of the proposed approximations on image compression experiments;
6. proposing low-complexity approximations for the DFT;
7. proposing a low-complexity detector for signal detection based on the DFT approximation;
8. assessing the performance of the low-complexity detector by numeric simulations.

1.3 ORGANIZATION OF THE THESIS

This doctorate thesis contains five self-contained chapters, each with its own notation and terminology. Each chapter corresponds to the research progress obtained by the candidate as of the writing of the thesis. The chapters referring to the KLT and DCT approximations have been more developed, and some of them are already published in the literature.

In Chapter 2, we present a class of data-independent low-complexity KLT approximations based on the signum function. We proposed 4-, 8-, 16-, and 32-point approximations because of the relevance of these lengths for image and video coding. We also proposed total figures of merit for the selection of the optimal transforms and fast algorithms for the approximations were derived. Finally, to show the applicability of the proposed approximations, experiments on image and video coding were conducted and demonstrated their good performance at a very low cost. The video coding experiments presented in Section 2.4.2 was executed by Dr. Thiago L. T. da Silveira.

In Chapter 3, we present another class of low-complexity KLT approximations based on the rounding-off function. The proposed KLT approximations cover the entire correlation scenario. That is, depending on the correlation coefficient of the input signal, we will have a specific low-complexity transform. We assessed the proposed transforms according to classical

figures of merit and showed that the performance is similar to the exact KLT at a greatly reduced computational cost. Image compression experiments were conducted and also showed the good performance of the proposed approximations when compared to the exact KLT.

In Chapter 4, we considered a wider range of rounding functions for searching for new KLT approximations. Since this search returned a large number of candidate matrices (7^8) we had to set an optimization problem to select the optimal approximations. The solution for the optimization problem comprises six new transforms. The proposed transforms were also assessed by coding and similarity measures, and experiments on image compression were carried out affirming their good performance. The 8-point low-complexity approximations were implemented on a field programmable gate array (FPGA), showing a trade-off between performance and resource usage. The hardware implementation section, presented in 4.7, was developed by Dr. Diego F. G. Coelho.

In Chapter 5, we presented new DCT approximations for larger blocklength based on minimal angle similarity. We considered an approach that is already known in the literature to propose 16-, 32-, and 64-point approximations for the DCT and showed that the approximations outperformed well-known low-complexity approximations according to some figures of merit. The proposed approximations were assessed in terms of image compression experiments, where we were, one more time, able to show that they outperformed approximations already known in the literature. Fast algorithms for the proposed DCT approximations were also developed. The approach used to derive the new approximations were already presented in a paper authored by Ms. Raiza S. Oliveira and others [80], and the fast algorithms presented in Section 5.5.4 were developed by Ms. Luan Portella.

In Chapter 6, we present the signal detection problem as in the context of a statistical hypothesis test. In this context, the usual statistical test can be defined in terms of the DFT, which can be an issue in scenarios where agility in detection is needed. Considering the problem of detecting a simple sinusoidal tone, we proposed a low-complexity statistical test based on an approximation for the DFT. We showed that the proposed test performs similarly to the exact test, at a lower cost. Simulated experiments were carried out to show the applicability of the test.

Finally, in Chapter 7 we present some conclusions and ideas for further work.

1.4 COMPUTATIONAL SUPPORT

The simulation results we presented in this thesis were obtained using R [108] and Python [109] software. The plots and images generated from Chapters 2, 3, 4, and 5 were produced using the R language. In Chapter 6, the Python language was considered to generate the plots.

2 DATA-INDEPENDENT LOW-COMPLEXITY KLT APPROXIMATIONS FOR IMAGE AND VIDEO CODING

2.1 INTRODUCTION

The Karhunen-Loève transform (KLT) [13] is a commonly used tool for data decorrelation and dimensionality reduction [110,111]. It consists of a linear transformation that maps correlated variables into uncorrelated ones, sometimes referred to as principal components [112]. 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 [113]. In fact, considering first-order Markov 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 [13]. Although it is a well-established optimal transform in terms of energy compaction and decorrelation [14], 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 ρ, then it was shown in [114] that we can derive an analytical solution for the elements of the KLT matrix. Nevertheless, even with the transform matrix known, the 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 [25, 41–44, 94–98] 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 [41, 43, 94–98] or on the correlation coefficient in case of first-order Markovian signals [25].

When considering first-order Markovian random signals, [115] and [116] have shown that the Discrete Cosine Transform (DCT-II) and the Discrete Sine Transform (DST) are asymptotic approximations for the KLT, with the correlation coefficient of the input signal tending to unity and to zero, respectively. 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 [26] and high efficiency video coding (HEVC) [28], 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 [74, 81, 117, 118]. In

fact, DCT realizations that require multiplications implemented in floating-point arithmetic-based hardware [82] demand significant circuitry complexity and energy consumption [13]. In this sense, several multiplication-free approximations for the DCT have been proposed [72, 74, 80, 81, 84–86, 90, 119–123], including the SDCT [71]. 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 [13].

The present work employs the signum 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 analysis are devoted to the blocklengths $N \in \{4, 8, 16, 32\}$ because of their relevance in image and video standards as JPEG [26] and HEVC [28]. 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 [124, 125], and similarity/proximity metrics [13, 72] 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, the literature lacks KLT approximations that combine the following properties:

1. deterministic definition;
2. suitability for fast algorithm design;
3. data-independence; and
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 this thesis chapter is to propose low-complexity approximate transforms for the KLT considering different values of the correlation coefficient 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 chapter is structured as follows. In Section 2.2, we revise the mathematical formulation of the KLT for first-order Markovian signals and define the general framework for the proposed SKLT. Section 2.3 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 2.3 also presents fast algorithms for the proposed transforms. In Section 2.4, we assess the proposed 4-, 8-, 16-, and 32-point SKLT in image and video coding. Section 2.5 concludes the chapter.

2.2 SIGNED KLT

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

The KLT is a linear transformation represented by an orthogonal matrix $\mathbf{K}_N^{(\rho)}$ which decorrelates an input signal

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \dots & x_{N-1} \end{bmatrix}^\top \tag{2.1}$$

resulting in an uncorrelated signal

$$\mathbf{y} = \begin{bmatrix} y_0 & y_1 & \dots & y_{N-1} \end{bmatrix}^\top. \tag{2.2}$$

The (i,j)th elements of the transform matrix $\mathbf{K}_N^{(\rho)}$, for an arbitrary value of $\rho \in [0,1]$, are given by [13]

$$k_{ij} = \sqrt{\frac{2}{N+\lambda_j}} \sin\left[\omega_j \left(i - \frac{N-1}{2}\right) + \frac{(j+1)\pi}{2}\right], \qquad i,j = 0,1,\dots,N-1, \tag{2.3}$$

where the eigenvalues of the transformed signal $\mathbf{y}$ covariance matrix are obtained by

$$\lambda_j = \frac{1-\rho^2}{1+\rho^2-2\rho\cos\omega_j}, \quad j = 0,1,\dots,N-1, \tag{2.4}$$

and $\omega_1, \omega_2, \dots, \omega_N$ are the N solutions of the non-linear equation

$$\tan(N\omega) = \frac{-(1-\rho^2)\sin\omega}{(1+\rho^2)\cos\omega - 2\rho}. \tag{2.5}$$

It is a well-known fact that adjacent pixels from natural images are highly correlated [24], being $\rho = 0.95$ a widely adopted assumption [13]. When the correlation of the input signal tends to the unity, $\rho \to 1$, the KLT converges to the DCT [115].

For instance, if $N = 8$ and $\rho = 0.95$, then the KLT matrix is given by

$$\mathbf{K}_8^{(0.95)} = \begin{bmatrix} 0.338 & 0.351 & 0.360 & 0.364 & 0.364 & 0.360 & 0.351 & 0.338 \\ 0.481 & 0.420 & 0.286 & 0.101 & -0.101 & -0.286 & -0.420 & -0.481 \\ 0.467 & 0.207 & -0.179 & -0.456 & -0.456 & -0.179 & 0.207 & 0.467 \\ 0.423 & -0.085 & -0.487 & -0.278 & 0.278 & 0.487 & 0.085 & -0.423 \\ 0.360 & -0.347 & -0.356 & 0.351 & 0.351 & -0.356 & -0.347 & 0.360 \\ 0.283 & -0.488 & 0.094 & 0.415 & -0.415 & -0.094 & 0.488 & -0.283 \\ 0.195 & -0.462 & 0.460 & -0.190 & -0.190 & 0.460 & -0.462 & 0.195 \\ 0.100 & -0.279 & 0.416 & -0.490 & 0.490 & -0.416 & 0.279 & -0.100 \end{bmatrix}. \tag{2.6}$$

2.2.2 KLT Approximations

Our approach is based on the technique used in [71] for proposing the classical 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 [71]. The technique considers the signum function to generate a matrix approximation for the KLT. Thus, we propose the following approximate transformation matrix:

$$\widehat{\mathbf{T}}_N^{(\rho)} \triangleq \frac{1}{\sqrt{N}} \operatorname{sign}\left(\mathbf{K}_N^{(\rho)}\right), \tag{2.7}$$

where

$$\operatorname{sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0, \end{cases} \tag{2.8}$$

and $\mathbf{K}_N^{(\rho)}$ is the KLT matrix of order N with a predefined correlation coefficient ρ, the entries of which are given by (2.3). When applied to a matrix, the signum function operates element-wise.

In other words, we map a given KLT matrix into a low-complexity matrix close to it. Note that, if $\rho = 0$, then the Equation (2.7) degenerates into the null matrix. Also if $\rho = 1$, then $\mathbf{K}_N^{(1)}$ is the DCT and the resulting approximation is the signed DCT (SDCT) [71]. Therefore, in practice, our analysis is constrained to $0 < \rho < 1$. 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 $\rho_1 = 0.7$ and $\rho_2 = 0.9$, we obtain that

$$\widehat{\mathbf{T}}_8^{(\rho_1)} = \widehat{\mathbf{T}}_8^{(\rho_2)} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \tag{2.9}$$

Exhaustively computing all KLT matrices in the range $\rho \in [10^{-3}, 1 - 10^{-3}]$ in steps of 10^{-3} returns 999 matrices. However, the number of approximations is lower due to the fact that many matrices can collapse into the same approximation. Therefore, we have : 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 $\widehat{\mathbf{T}}_N^{(\rho)}$, the transformed signal is given by

$$\mathbf{y} = \widehat{\mathbf{T}}_N^{(\rho)} \cdot \mathbf{x}, \tag{2.10}$$

and the inverse transformation can be written as

$$\mathbf{y} = (\widehat{\mathbf{T}}_N^{(\rho)})^{-1} \cdot \mathbf{x}. \tag{2.11}$$

2.3 OPTIMAL SKLT

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

2.3.1 Figures of Merit for Approximate Transforms

Approximate transform methods [71, 72, 74, 81, 117] are usually assessed in terms of (i) coding metrics such as the coding gain [124] and transform efficiency [125], 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 [13] and total error energy [72], which measure similarities or dissimilarities between approximate and exact transforms. In the following, let $\widehat{\mathbf{T}}_N$ be a candidate matrix to be assessed.

2.3.1.1 Unified Coding Gain

The unified coding gain of a transform $\widehat{\mathbf{T}}_N$ is given by [126]

$$\mathrm{C}g(\widehat{\mathbf{T}}_N) = 10 \cdot \log_{10}\left\{ \prod_{k=1}^{N} \frac{1}{\sqrt[N]{A_k \cdot B_k}} \right\}, \tag{2.12}$$

where $A_k = \mathrm{su}\left\{ (\mathbf{h}_k^\top \cdot \mathbf{h}_k) \odot \mathbf{R}_\mathbf{x} \right\}$, $\mathbf{h}_k$ is the kth row vector from $\widehat{\mathbf{T}}_N$, the function $\mathrm{su}(\cdot)$ returns the sum of the elements of its matrix argument, $\odot$ is the Hadamard matrix product operator [127], $\mathbf{R}_\mathbf{x}$ is the autocorrelation matrix of the considered first-order Markovian signal, $B_k = \|\mathbf{g}_k\|^2$ and $\mathbf{g}_k$ is the kth row vector from $\widehat{\mathbf{T}}_N^{-1}$, and $\|\cdot\|$ is the Frobenius norm [127].

2.3.1.2 Transform Efficiency

Another coding related figure of merit is the transform efficiency, given by [125]

$$\eta(\widehat{\mathbf{T}}_N) = 100\frac{\sum_{i=1}^{N}|r_{i,i}|}{\sum_{i=1}^{N}\sum_{j=1}^{N}|r_{i,j}|}, \tag{2.13}$$

where $r_{i,j}$ is the (i,j)th element from $\widehat{\mathbf{T}}_N \cdot \mathbf{R}_\mathbf{x} \cdot \widehat{\mathbf{T}}_N^\top$.

2.3.1.3 Mean-Square Error

The mean-square error (MSE) relative to the KLT is given by [13]:

$$\mathrm{MSE}(\widehat{\mathbf{T}}_N) = \frac{1}{N} \cdot \mathrm{tr}\left\{(\mathbf{K}_N^{(\rho)} - \widehat{\mathbf{T}}_N) \cdot \mathbf{R}_\mathbf{x} \cdot (\mathbf{K}_N^{(\rho)} - \widehat{\mathbf{T}}_N)^\top\right\}, \tag{2.14}$$

where $\mathrm{tr}(\cdot)$ is the trace function [128].

2.3.1.4 Total Error Energy

The total error energy of an approximation relative to the KLT is computed by [72]:

$$\varepsilon(\widehat{\mathbf{T}}_N) = \pi \cdot ||\mathbf{K}_N^{(\rho)} - \widehat{\mathbf{T}}_N||^2. \tag{2.15}$$

2.3.1.5 Proposed Figures of Merit

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

$$\mathrm{C}g_T(\widehat{\mathbf{T}}_N) = \int_0^1 |\mathrm{C}g(\mathbf{K}_N^{(\rho)}) - \mathrm{C}g(\widehat{\mathbf{T}}_N)|\,\mathrm{d}\rho, \tag{2.16}$$

$$\eta_T(\widehat{\mathbf{T}}_N) = \int_0^1 |\eta(\mathbf{K}_N^{(\rho)}) - \eta(\widehat{\mathbf{T}}_N)|\,\mathrm{d}\rho, \tag{2.17}$$

$$\mathrm{MSE}_T(\widehat{\mathbf{T}}_N) = \int_0^1 \mathrm{MSE}(\widehat{\mathbf{T}}_N)\,\mathrm{d}\rho, \tag{2.18}$$

$$\varepsilon_T(\widehat{\mathbf{T}}_N) = \int_0^1 \varepsilon(\widehat{\mathbf{T}}_N)\,\mathrm{d}\rho. \tag{2.19}$$

2.3.2 Optimization Problem

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

$$\widehat{\mathbf{T}}_N^* = \underset{0<\rho<1}{\arg\min}\,\mathrm{error}(\widehat{\mathbf{T}}_N^{(\rho)}), \tag{2.20}$$

where error$(\cdot)$ is one of the proposed measures, C$g_T(\cdot)$, $\eta_T(\cdot)$, M$SE_T(\cdot)$, $\varepsilon_T(\cdot)$, 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 $\widehat{\mathbf{T}}_{N,i}$, which are indexed by the subscript $i = 1,2,\ldots,J$, where J is the number of approximate transforms for the blocklength N.

Table 1 summarizes the results for the optimal SKLT with different transform lengths N and for the intervals of ρ that each transform is defined. DCT, DST, and the SDCT results are included only for comparison purposes. It is important to emphasize that this comparison is not completely fair since the proposed transforms cover different intervals of ρ while the DCT and DST are defined for ρ 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 $\widehat{\mathbf{T}}_{4,1}$, $\widehat{\mathbf{T}}_{8,2}$, and $\widehat{\mathbf{T}}_{16,3}$ are already known in the literature, and coincide with the SDCT. The remaining transforms are, to the best of our knowledge, new ones.

2.3.3 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 [23], 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 [23, 129, 130]. 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.

2.3.3.1 Matrix Factorization for $N = 4$

For $N = 4$, we can factorize $\widehat{\mathbf{T}}_{4,1}$ [71] as follows:

$$\widehat{\mathbf{T}}_{4,1} = \frac{1}{2} \cdot \mathbf{P}_4 \cdot \mathbf{A}_{4,2} \cdot \mathbf{A}_{4,1}, \tag{2.21}$$

Table 1 – Comparison of proposed transforms with KLT

Transform	ρ	$\mathrm{C}g_T(\widehat{\mathbf{T}}_N)$	$\eta_T(\widehat{\mathbf{T}}_N)$	$\mathrm{M}SE_T(\widehat{\mathbf{T}}_N)$	$\varepsilon_T(\widehat{\mathbf{T}}_N)$
		$N=4$			
$\widehat{\mathbf{T}}_{4,1}$ [71]	$(0,1)$	**0.162**	**10.862**	**0.039**	**0.764**
DCT	$\rho \to 1$	0.025	7.071	0.014	0.167
DST	$\rho \to 0$	0.317	11.859	0.016	0.167
		$N=8$			
$\widehat{\mathbf{T}}_{8,1}$	$(0,0.619]$	2.726	**32.907**	**0.110**	**3.670**
$\widehat{\mathbf{T}}_{8,2}$ [71]	$(0.619,1)$	**2.170**	37.192	0.129	3.950
DCT	$\rho \to 1$	0.031	12.173	0.042	0.888
DST	$\rho \to 0$	0.362	18.940	0.032	0.778
		$N=16$			
$\widehat{\mathbf{T}}_{16,1}$	$(0.540,0.550]$	2.825	56.924	0.157	**8.499**
$\widehat{\mathbf{T}}_{16,2}$	$(0.550,0.700]$	2.748	54.668	**0.153**	8.538
$\widehat{\mathbf{T}}_{16,3}$ [71]	$[0.9,1)$	**2.227**	**48.588**	0.169	9.532
DCT	$\rho \to 1$	0.024	15.709	0.078	2.945
DST	$\rho \to 0$	0.307	22.772	0.044	2.283
		$N=32$			
$\widehat{\mathbf{T}}_{32,1}$	$(0.139,0.162]$	**2.352**	66.355	0.195	21.143
$\widehat{\mathbf{T}}_{32,2}$	$(0.487,0.490]$	2.442	65.047	**0.184**	19.806
$\widehat{\mathbf{T}}_{32,3}$	$(0.490,0.528]$	2.477	65.475	0.185	**19.779**
$\widehat{\mathbf{T}}_{32,4}$	$(0.956,0.977]$	2.450	**61.930**	0.244	24.288
SDCT	$\rho \to 1$	2.491	62.550	0.256	24.856
DCT	$\rho \to 1$	0.015	18.005	0.116	7.891
DST	$\rho \to 0$	0.227	24.673	0.052	5.531

Source: Author (2023).

where

$$\mathbf{P}_4 = \begin{bmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{bmatrix}, \quad \mathbf{A}_{4,2} = \begin{bmatrix} 1 & 1 & & \\ 1 & -1 & & \\ & & 1 & 1 \\ & & -1 & 1 \end{bmatrix}, \tag{2.22}$$

and

$$\mathbf{A}_{4,1} = \begin{bmatrix} \mathbf{I}_2 & \bar{\mathbf{I}}_2 \\ \bar{\mathbf{I}}_2 & -\mathbf{I}_2 \end{bmatrix}, \tag{2.23}$$

where $\mathbf{I}_2$ and $\bar{\mathbf{I}}_2$ are, respectively, the identity and counter-identity matrices of order 2. The blank spaces represents zeroes.

2.3.3.2 Matrix Factorization for $N = 8$

For $N = 8$, we have:

$$\widehat{\mathbf{T}}_{8,1} = \frac{1}{\sqrt{8}} \cdot \mathbf{P}_8 \cdot \mathbf{A}'_{8,3} \cdot \mathbf{A}_{8,2} \cdot \mathbf{A}_{8,1}, \tag{2.24}$$

$$\widehat{\mathbf{T}}_{8,2} = \frac{1}{\sqrt{8}} \cdot \mathbf{P}_8 \cdot \mathbf{A}''_{8,3} \cdot \mathbf{A}_{8,2} \cdot \mathbf{A}_{8,1} \text{ [71]}, \tag{2.25}$$

where

$$\mathbf{P}_8 = \begin{bmatrix} 1 & & & & & & & \\ & & & 1 & & & & \\ & & 1 & & & & & \\ & & & & 1 & & & \\ & 1 & & & & & & \\ & & & & & & 1 & \\ & & & 1 & & & & \\ & & & & & & & 1 \end{bmatrix}, \quad \mathbf{A}'_{8,3} = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & & 1 & 1 & & & & \\ & & -1 & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & & -1 & -1 \\ & & & & 1 & & 1 & \\ & & & & & & 1 & -1 \end{bmatrix}, \tag{2.26}$$

$$\mathbf{A}_{8,2} = \begin{bmatrix} \mathbf{A}_{4,1} & \\ & \mathbf{A}_{4,1} \end{bmatrix}, \quad \mathbf{A}_{8,1} = \begin{bmatrix} \mathbf{I}_4 & \bar{\mathbf{I}}_4 \\ \bar{\mathbf{I}}_4 & -\mathbf{I}_4 \end{bmatrix}, \tag{2.27}$$

and

$$\mathbf{A}''_{8,3} = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & & 1 & 1 & & & & \\ & & 1 & -1 & & & & \\ & & & & 1 & 1 & & \\ & & & & & -1 & & -1 \\ & & & & 1 & & 1 & \\ & & & & & & 1 & -1 \end{bmatrix}. \tag{2.28}$$

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,

Table 2 – Computational cost comparison for 4-, 8-, 16-, and 32-point transforms

Transform	Additions	Multiplications	Bit-Shifting	Addition reduction (%)
$\widehat{\mathbf{T}}_{4,1}$ [71]	4	0	0	66%
$\mathbf{K}_4$	12	16	0	-
$\widehat{\mathbf{T}}_{8,1}$	24	0	0	57%
$\widehat{\mathbf{T}}_{8,2}$ [71]	24	0	0	57%
$\mathbf{K}_8$	56	64	0	-
$\widehat{\mathbf{T}}_{16,1}$	80	0	0	67%
$\widehat{\mathbf{T}}_{16,2}$	75	0	0	69%
$\widehat{\mathbf{T}}_{16,3}$ [71]	72	0	0	70%
$\mathbf{K}_{16}$	240	256	0	-
$\widehat{\mathbf{T}}_{32,1}$	288	0	0	71%
$\widehat{\mathbf{T}}_{32,2}$	288	0	0	71%
$\widehat{\mathbf{T}}_{32,3}$	288	0	0	71%
$\widehat{\mathbf{T}}_{32,4}$	232	0	0	77%
$\mathbf{K}_{32}$	992	1024	0	-

Source: Author (2023).

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 [45] and [60] for the 8- and 16-point DCT, respectively; and approximations for the DCT already known in the literature proposed in [71–74, 76, 77, 83–85, 119, 120, 131, 132]. 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 1 and 2, 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 2, 3, 4, 5, respectively. Note that both DCT and DST also require multiplications when implemented, differently of the proposed transforms that are multiplierless.

2.4 IMAGE AND VIDEO CODING

We applied the proposed transforms to two distinct contexts that are commonly explored in the field of approximation: (i) still image compression according to a JPEG-like algorithm [72,

Table 3 – Computational cost comparison for 8-, 16-, and 32-point DCT approximation transforms

Transform	Additions	Multiplications	Bit-Shifting
DCT_8 [45]	29	11	0
DST_8 [133]	29	12	0
$\mathbf{T}_{8,\text{LO}}$ [120]	24	0	2
$\mathbf{T}_{8,\text{RDCT}}$ [72]	22	0	0
$\mathbf{T}_{8,\text{MRDCT}}$ [83]	14	0	0
$\mathbf{T}_{8,\text{BAS-2008a}}$ [74]	18	0	2
$\mathbf{T}_{8,\text{BAS-2008b}}$ [74]	21	0	2
$\mathbf{T}_{8,\text{BAS-2009}}$ [77]	18	0	0
$\mathbf{T}_{8,\text{BAS-2011}}$ [76]	16	0	0
$\mathbf{T}_{8,\text{BAS-2012}}$ [119]	24	0	0
$\mathbf{T}'_{8,1}$ [73]	18	0	0
$\mathbf{T}_{8,4}$ [73]	24	0	0
$\mathbf{T}_{8,5}$ [73]	24	0	4
$\mathbf{T}_{8,6}$ [73]	24	0	6
DCT_{16} [60]	74	44	0
DST_{16} [133]	81	32	0
$\mathbf{T}_{16,\text{BAS-2012}}$ [119]	64	0	0
$\mathbf{T}_{16,\text{BCEM}}$ [131]	72	0	0
$\mathbf{T}_{16,\text{SBCKMK}}$ [132]	60	0	0
$\mathbf{T}_{16,\text{SOBCM}}$ [85]	44	0	0
$\mathbf{T}_{16,\text{JAM}}$ [84]	60	0	0
DCT_{32} [60]	194	116	0
DST_{32} [133]	209	80	0
$\mathbf{T}_{32,\text{BAS-2012}}$ [119]	160	0	0
$\mathbf{T}_{32,\text{JAM}}$ [84]	152	0	0

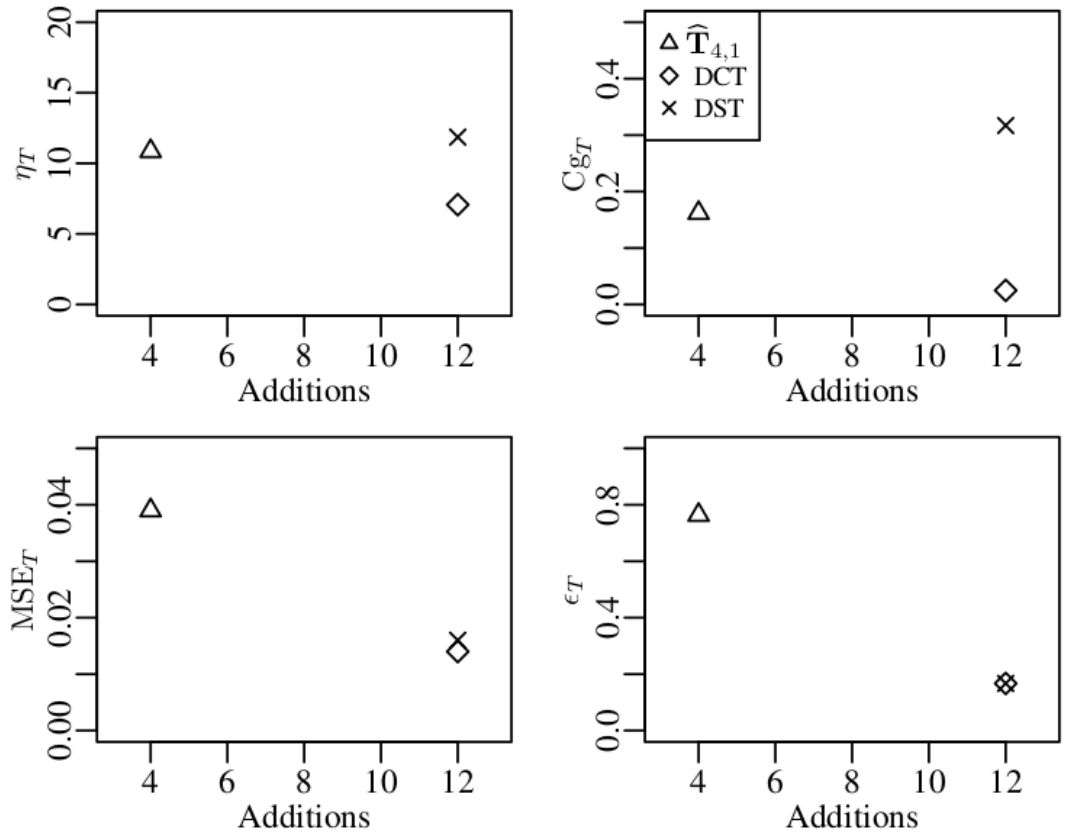
Source: Author (2023).

74, 81], and (ii) video encoding as defined in the HEVC reference software [134]. In this section, we compared the proposed transforms with the exact KLT for $\rho = 0.1$, 0.6 and 0.9 ($\mathbf{K}^{(0.1)}$, $\mathbf{K}^{(0.6)}$ and $\mathbf{K}^{(0.9)}$), and with the exact DCT. The transforms $\mathbf{K}^{(0.1)}$, $\mathbf{K}^{(0.6)}$, and $\mathbf{K}^{(0.9)}$ 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.

2.4.1 Image Compression

In this section, we evaluated the performance of the proposed transforms in image compression, similarly to [72, 74, 81]. If $\mathbf{A}$ is a two-dimensional (2D) image, then the direct and

Figure 2 – Additive complexity versus the proposed figures of merit for each transform for $N = 4$.



Source: Author (2023).

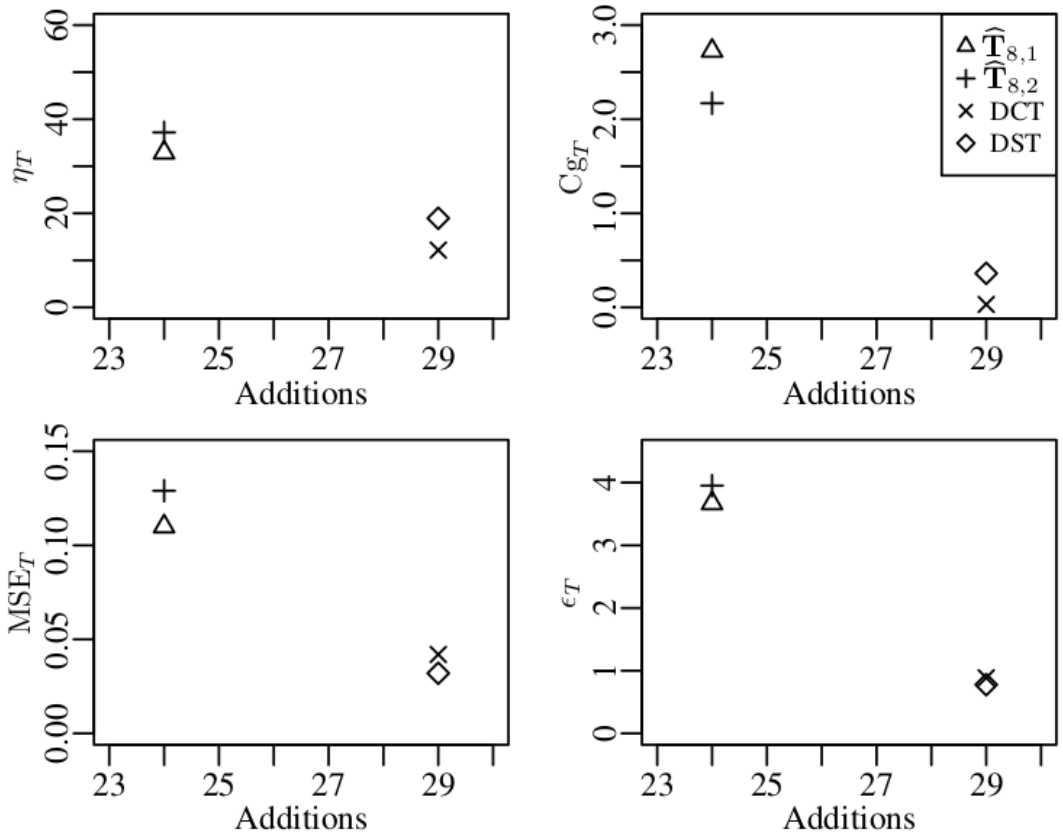
inverse transformations induced by the SKLT are computed, respectively, by

$$\mathbf{B}_j = \widehat{\mathbf{T}}_N^{(\rho)} \cdot \mathbf{A}_j \cdot (\widehat{\mathbf{T}}_N^{(\rho)})^{-1}, \tag{2.29}$$

$$\mathbf{A}_j = (\widehat{\mathbf{T}}_N^{(\rho)})^{-1} \cdot \mathbf{B}_j \cdot \widehat{\mathbf{T}}_N^{(\rho)}. \tag{2.30}$$

The adopted experiment scheme is described as follows [16]: (i) the image is divided into disjoint sub-blocks $\mathbf{A}_j$ of size $N \times N$; (ii) each sub-block is submitted to a selected transform $\widehat{\mathbf{T}}_N^{(\rho)}$ according to (2.29); (iii) using the standard zig-zag sequence [16], only the initial r coefficients in each sub-block $\mathbf{B}_j$ are retained and the remaining $N^2 - r$ coefficients are zeroed, resulting in sub-block $\bar{\mathbf{B}}_j$; (iv) the two-dimensional inverse transform is applied according (2.30), and (v) the reconstructed sub-blocks $\bar{\mathbf{A}}_j$ are adequately rearranged. The final reconstructed image $\bar{\mathbf{A}}$ is compared with the original image $\mathbf{A}$ for assessing the performance of $\widehat{\mathbf{T}}_N$. We adopted $N \in \{4, 8, 16, 32\}$ and used the PSNR [135], and the MSSIM [136] as figures of merit for image quality evaluation. The results were taken individually for 45 512×512 8-bit greyscale images obtained from [137] and averaged. For each transform length, we considered two approaches: (i) a qualitative analysis, based on the compressed *Lena* image with approximately

Figure 3 – Additive complexity versus the proposed figures of merit for each transform for $N = 8$.



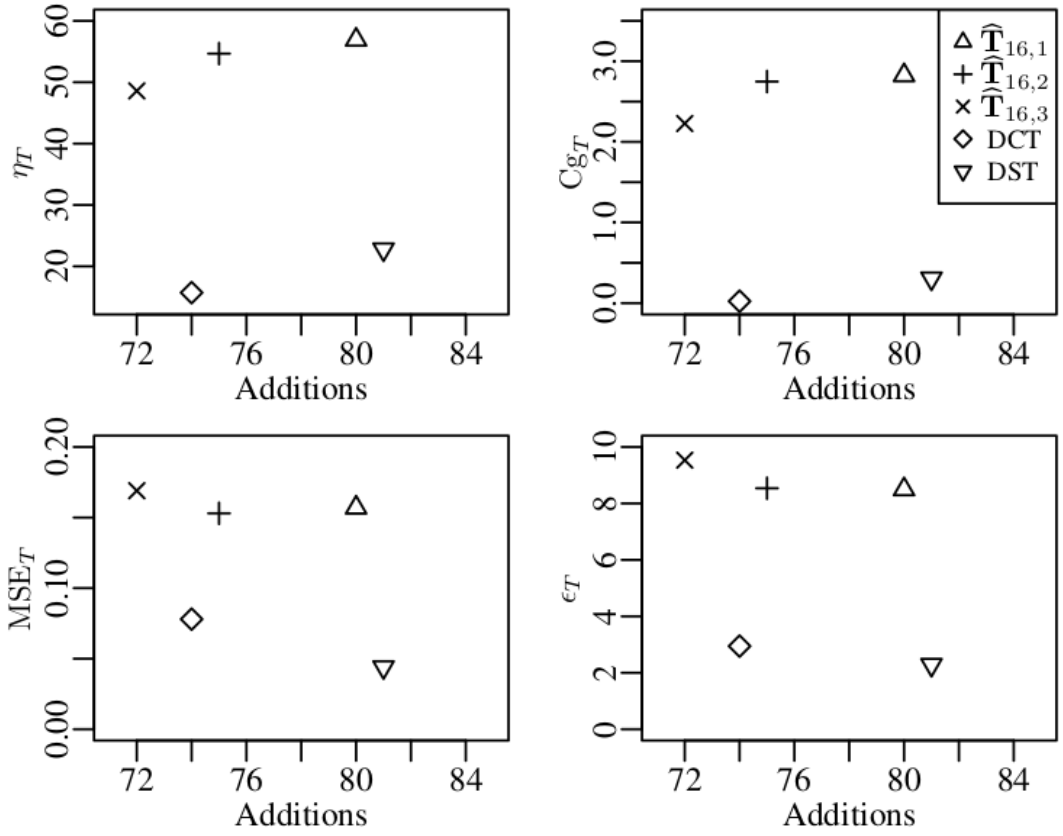
Source: Author (2023).

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 6 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 7, 8, 9, and 10. The corresponding compression ratio (CR) was CR = 81.25%, 84.38%, 84.38%, 84.86%, respectively. Visually, the reconstructed images after compression exhibit quality comparable with the original image.

Figure 11 presents the average image quality measurements for different values of N considering different levels of compression, comparing with the exact KLT for $\rho = 0.1$, 0.6, and 0.9 and the DCT. Figure 11b shows that the average MSSIM of $\widehat{\mathbf{T}}_{4,1}$ is very close to the results from the KLT and DCT. Figure 11a also shows that the values of the average PSNR of $\widehat{\mathbf{T}}_{4,1}$ are close to the values obtained by KLT and DCT. We can also notice that the approximation $\widehat{\mathbf{T}}_{4,1}$ 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 $\widehat{\mathbf{T}}_{8,1}$ and $\widehat{\mathbf{T}}_{8,2}$ behave in a similar way according to the image quality measures. For $N = 16$, the $\widehat{\mathbf{T}}_{16,3}$ transform has considerably closer values to the DCT and KLT than the other approximations. This may be related to the fact that

Figure 4 – Additive complexity versus the proposed figures of merit for each transform for $N = 16$.



Source: Author (2023).

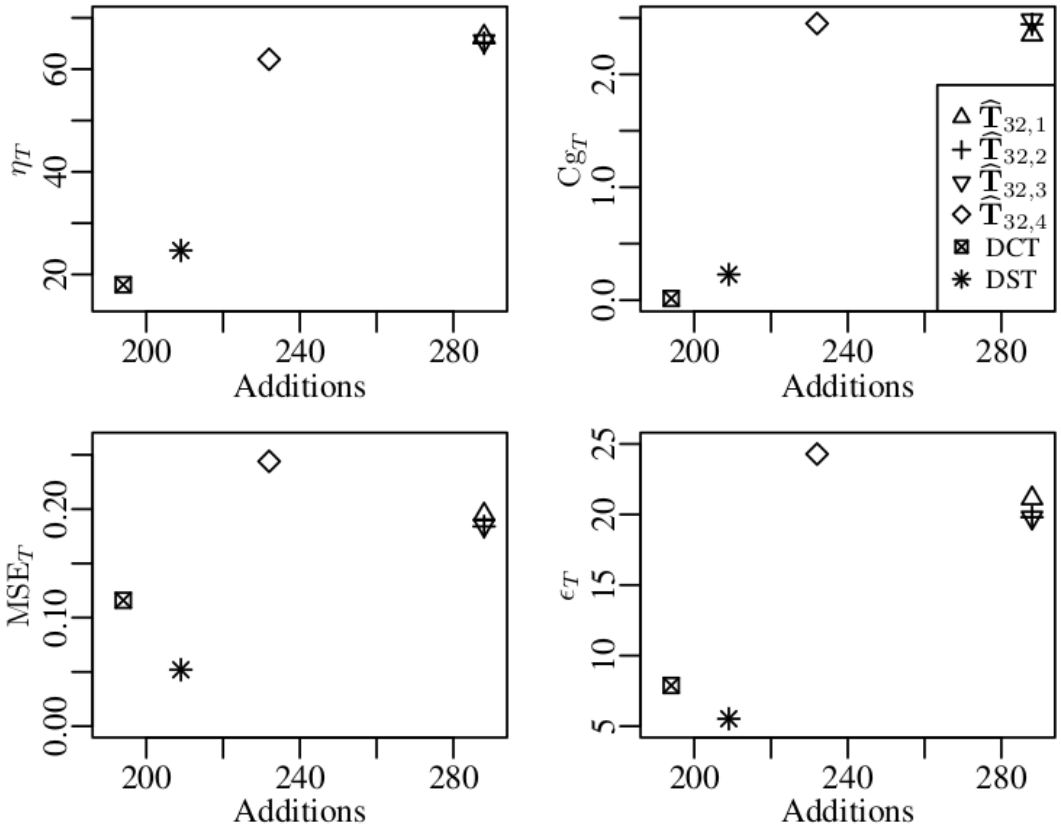
$\widehat{\mathbf{T}}_{16,3}$ is the obtained transform with a higher value of ρ ($\rho \geq 0.9$). One can see that, for $r < 10$, the approximation $\widehat{\mathbf{T}}_{16,3}$ presents better average MSSIM values than KLT itself. Figures 11g and 11h present the image quality measurements for $N = 32$, that poses $\widehat{\mathbf{T}}_{32,4}$ as the best-performing approximate KLT for compression.

2.4.2 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 [134]. The HEVC employs an IDCT of lengths 4, 8, 16, and 32 [138], unlike its predecessors [139–142]. According to [28], 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 [143] 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

Figure 5 – Additive complexity versus the proposed figures of merit for each transform for $N = 32$.



Source: Author (2023).

Figure 6 – Original *Lena* image.

Source: SIPI [137].

approaches the coding capabilities of the DCT [71]. We separated four suits of approximations, relating to the optimality in Equation (2.20):

(i) $\widehat{\mathbf{T}}_{4,1}$, $\widehat{\mathbf{T}}_{8,1}$, $\widehat{\mathbf{T}}_{16,2}$, and $\widehat{\mathbf{T}}_{32,2}$ (Group I);

(ii) $\widehat{\mathbf{T}}_{4,1}$, $\widehat{\mathbf{T}}_{8,1}$, $\widehat{\mathbf{T}}_{16,1}$, and $\widehat{\mathbf{T}}_{32,3}$ (Group II);

(iii) $\widehat{\mathbf{T}}_{4,1}$, $\widehat{\mathbf{T}}_{8,2}$, $\widehat{\mathbf{T}}_{16,3}$, and $\widehat{\mathbf{T}}_{32,1}$ (Group III); and

(iv) $\widehat{\mathbf{T}}_{4,1}$, $\widehat{\mathbf{T}}_{8,1}$, $\widehat{\mathbf{T}}_{16,3}$, and $\widehat{\mathbf{T}}_{32,4}$ (Group IV).

Namely, Groups I, II, III, and IV are optimal regarding total MSE, total error energy, total

Figure 7 – Compressed *Lena* images for $N = 4$ and $r = 3$.

(a) $\widehat{\mathbf{T}}_{4,1}$ [71] , PSNR = 30.758dB, MSSIM = 0.846

(b) $\mathrm{DCT}_4(\rho \to 1)$, PSNR = 32.001dB, MSSIM = 0.871

(c) $\mathrm{DST}_4(\rho \to 0)$, PSNR = 15.491dB, MSSIM = 0.127

Source: Author (2023).

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 one video sequence of each A to F class in accordance with the Common Test Conditions (CTC) document [144]. 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 `BasketbalDrillText` (832×480 at 50 fps). We further considered the `Foreman` (352×288 at 30 fps) [145], a standard 8-bit CIF video sequence adopted in related works like [85, 120]. As done in [84], 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 [138] 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 [144].

Moreover, we measured the Bjøntegaard's delta PSNR (BD-PSNR) [146, 147] 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 12, 13, 14, and 15, which show the RD curves for

Figure 8 – Compressed *Lena* images for $N = 8$ and $r = 10$.

(a) $\widehat{\mathbf{T}}_{8,1}$, PSNR = 27.876dB, MSSIM = 0.854

(b) $\widehat{\mathbf{T}}_{8,2}$ [71] , PSNR = 27.896dB, MSSIM = 0.861

(c) $\mathrm{DCT}_8(\rho \to 1)$, PSNR = 32.081dB, MSSIM = 0.913

(d) $\mathrm{DST}_8(\rho \to 0)$, PSNR = 18.297dB, MSSIM = 0.272

Source: Author (2023).

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 2.4.1.

As a qualitative example, we present in Figure 16 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.

2.5 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

Figure 9 – Compressed *Lena* images for $N = 16$ and $r = 40$.

(a) $\widehat{\mathbf{T}}_{16,1}$, PSNR = 25.717dB, MSSIM = 0.862

(b) $\widehat{\mathbf{T}}_{16,2}$, PSNR = 25.887dB, MSSIM = 0.866

(c) $\widehat{\mathbf{T}}_{16,3}$ [71] , PSNR = 27.200dB, MSSIM = 0.892

(d) $\mathrm{DCT}_{16}(\rho \to 1)$, PSNR = 32.494dB, MSSIM = 0.945

(e) $\mathrm{DST}_{16}(\rho \to 0)$, PSNR = 22.639dB, MSSIM = 0.532

Source: Author (2023).

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, **any approximation derived from this method is a 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 blocklenghts transforms and other applications for fully exploring the potential of the proposed transforms, such as the Versatile Video Coding (VVC)

Figure 10 – Compressed *Lena* images for $N = 32$ and $r = 155$.

(a) $\widehat{\mathbf{T}}_{32,1}$, PSNR = 17.519dB, MSSIM = 0.473

(b) $\widehat{\mathbf{T}}_{32,2}$, PSNR = 17.573dB, MSSIM = 0.476

(c) $\widehat{\mathbf{T}}_{32,3}$, PSNR = 20.632dB, MSSIM = 0.600

(d) $\widehat{\mathbf{T}}_{32,4}$, PSNR = 25.997dB, MSSIM = 0.911

(e) $\mathrm{DCT}_{32}(\rho \to 1)$, PSNR = 32.996dB, MSSIM = 0.969

(f) $\mathrm{DST}_{32}(\rho \to 0)$, PSNR = 26.089dB, MSSIM = 0.781

Source: Author (2023).

standard [148–150], for example.

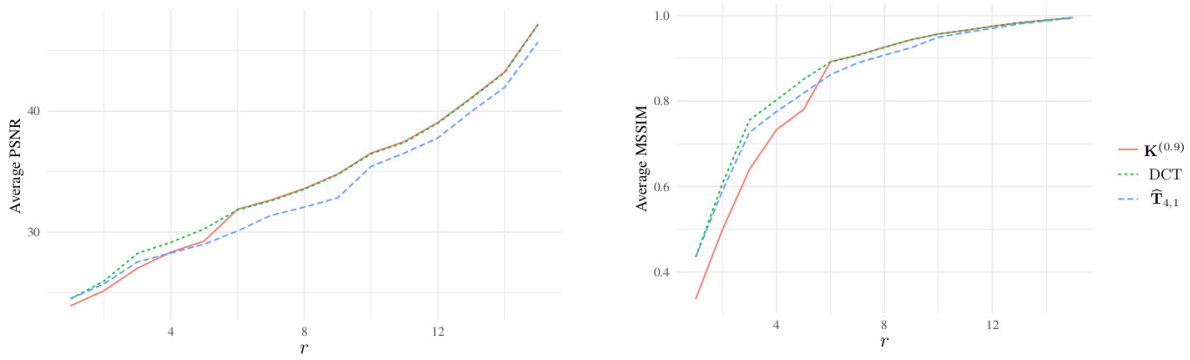
Table 4 – Average BD-PSNR of the modified HEVC reference software for tested video sequences

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
	BlowingBubbles	−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
	BlowingBubbles	−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
	BlowingBubbles	−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
	BlowingBubbles	−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

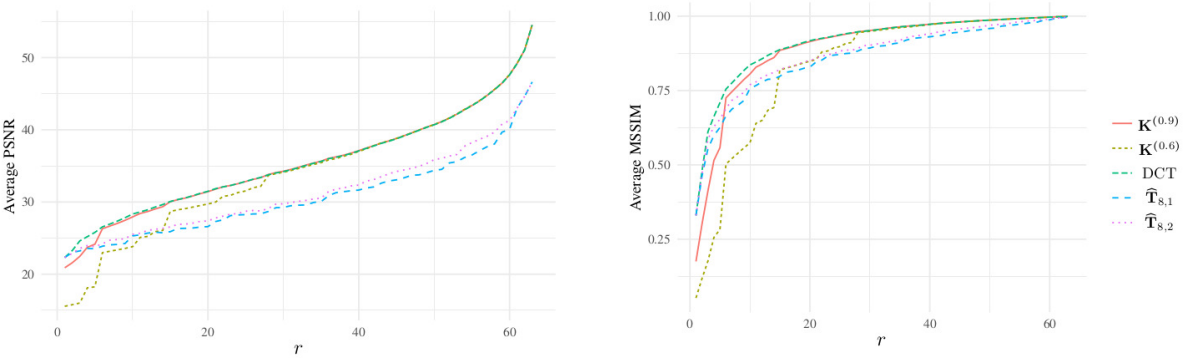
Source: Author (2023).

Figure 11 – Image quality measurements for different levels of compression.

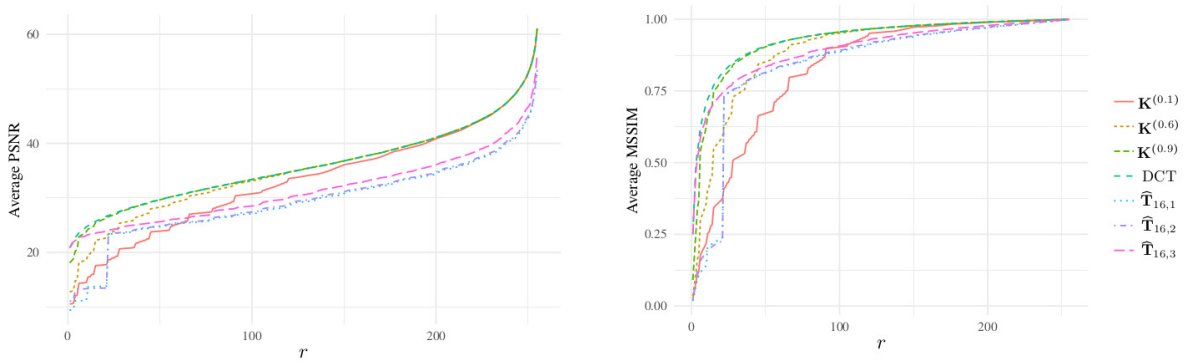
(a) Average PSNR ($N = 4$) **(b)** Average MSSIM($N = 4$)



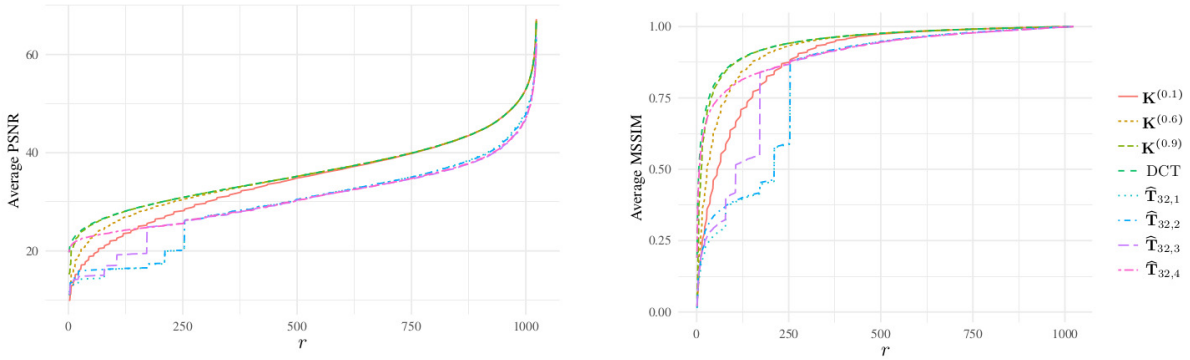
(c) Average PSNR ($N = 8$) **(d)** Average MSSIM ($N = 8$)



(e) Average PSNR ($N = 16$) **(f)** Average MSSIM ($N = 16$)

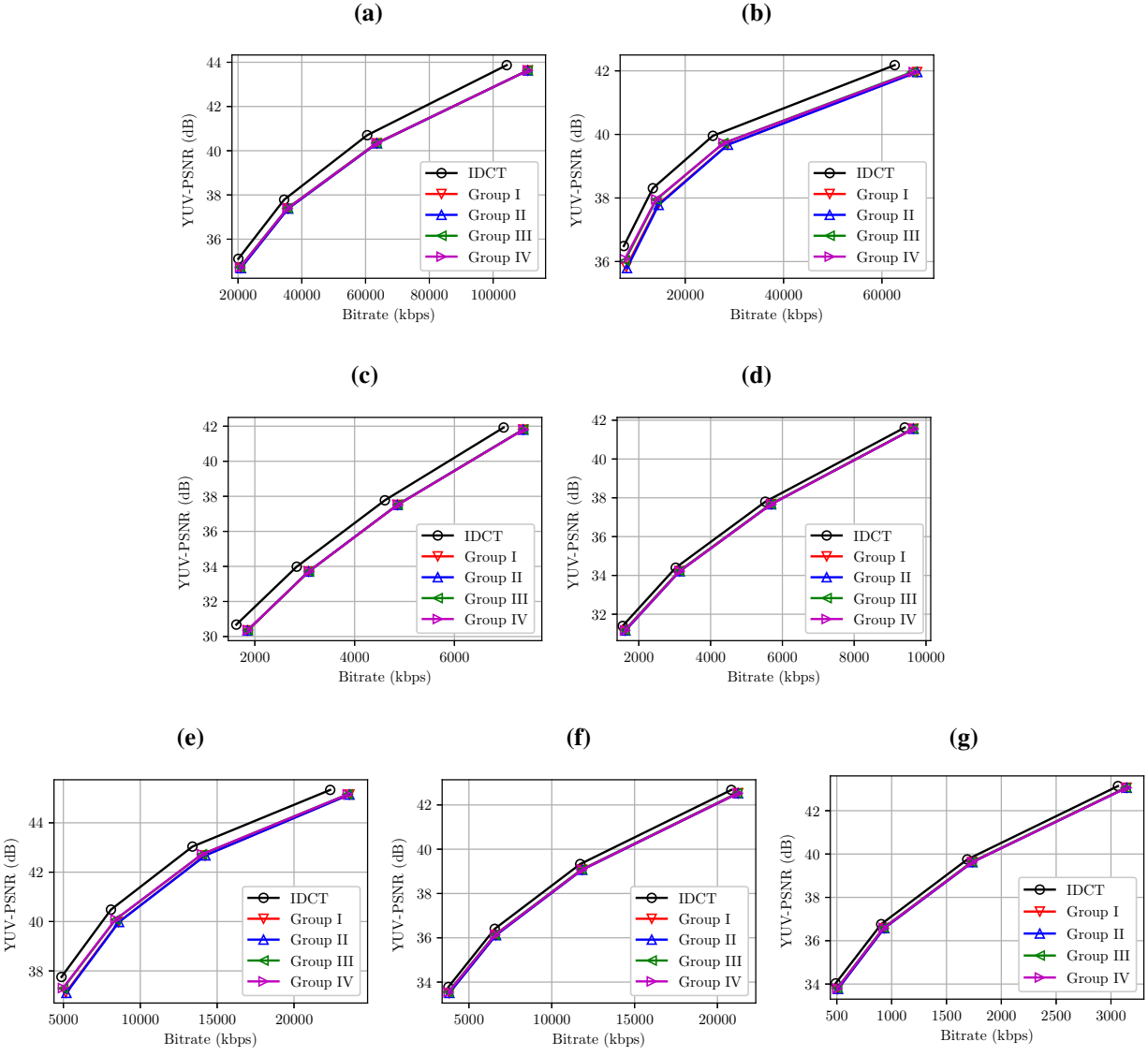


(g) Average PSNR ($N = 32$) **(h)** Average MSSIM ($N = 32$)



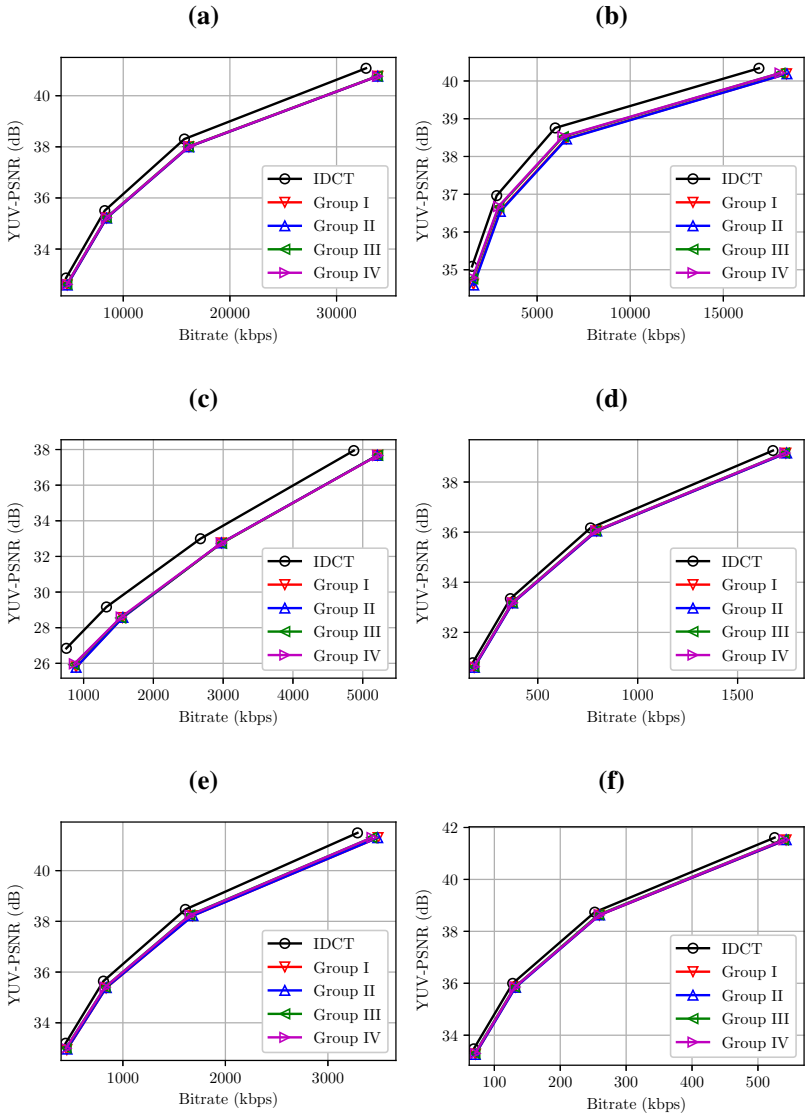
Source: Author (2023).

Figure 12 – 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) `BasketbalDrillText`, and (g) `Foreman`.



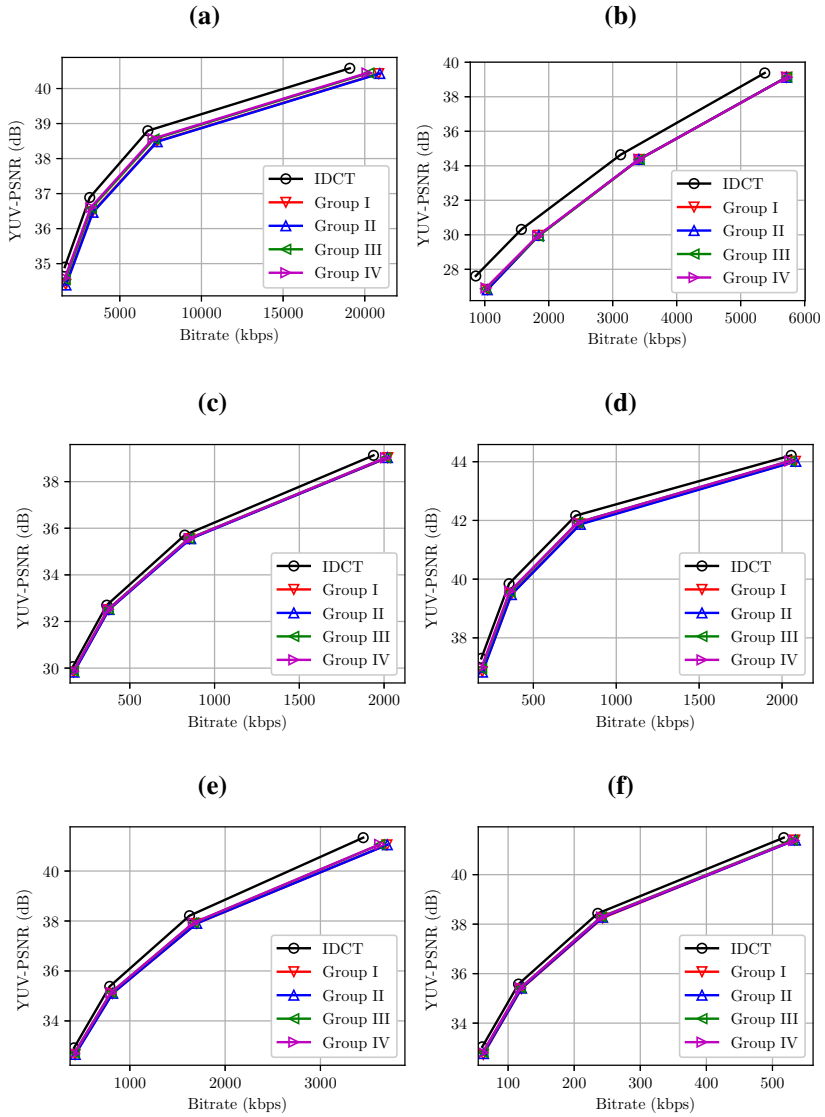
Source: Author (2023).

Figure 13 – 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) `BasketbalDrillText`, and (f) `Foreman`.



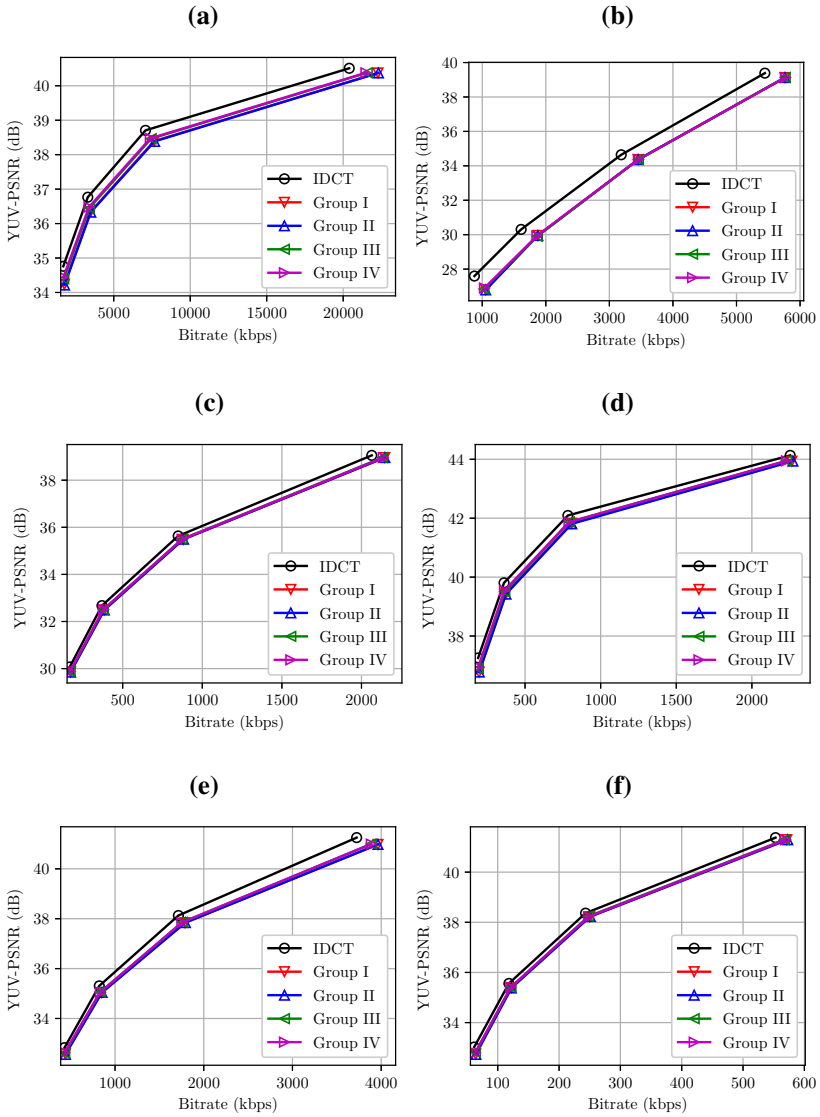
Source: Author (2023).

Figure 14 – 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) `BasketbalDrillText`, and (f) `Foreman`.



Source: Author (2023).

Figure 15 – 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) `BasketbalDrillText`, and (f) `Foreman`.



Source: Author (2023).

Figure 16 – 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.

(a) PSNR-Y = 39.4857dB, PSNR-U = 43.5883dB and PSNR-V = 44.4961dB

(b) PSNR-Y = 39.0457dB, PSNR-U = 43.0431dB and PSNR-V = 43.9074dB

(c) PSNR-Y = 39.0571dB, PSNR-U = 43.0027dB and PSNR-V = 43.8623dB

(d) PSNR-Y = 39.1155dB, PSNR-U = 43.1140dB and PSNR-V = 44.0624dB

(e) PSNR-Y = 39.1320dB, PSNR-U = 43.1120dB and PSNR-V = 44.0912dB

Source: Author (2023).

3 LOW-COMPLEXITY ROUNDED KLT APPROXIMATION FOR IMAGE COMPRESSION

3.1 INTRODUCTION

We are living in the era where colossal amounts of data are generated every day, the Big Data era [3]. With increasing computational demands for analyzing large masses of data in real-time, the development of low-complexity signal processing methods became an area of great interest [151]. In this context, data compression techniques are useful tools for reducing data dimensionality, a necessary requisite for efficient data transmission and storage. There are several methods for data compression [110, 152] and they are all based on the same principle of reducing or removing redundancy from the input data [153]. In particular, transform-based compression methods generally map the input data into smaller output data [16].

Among several transforms found in the literature, the Karhunen-Loève transform (KLT) [13, 21, 22] has the distinction of being capable of completely decorrelate the input signal in the transform domain [13, 14]. In fact, the KLT is the optimal linear transform that minimizes the mean squared error in data compression for maximum energy concentration in a few coefficients of the output signal. Although mathematically optimal, the KLT has limited applicability, because its derivation depends on the covariance matrix of the input data, thus precluding or hindering the development of fast algorithms for its computation. However, if the input data is a first-order Markov process with known correlation coefficient ρ, then the associate covariance matrix is known and fast algorithms are possible [114]. Nonetheless, even when a fast algorithm is possible, it requires multiplications by a significant amount of irrational numbers, increasing the computational cost of the algorithm. To the best of our knowledge, literature is scarce in methods devoted to the efficient implementation of the KLT [25, 154, 155]. Our approach differs from the literature mainly because we consider first-order Markov signals for different values of ρ. In this way, depending on the correlation coefficient of the input signal, we will have a low-complexity transform.

The KLT is mathematically linked to the DCT [13, 115]. In fact, the DCT is itself an asymptotic approximation for the KLT when (i) the input data is first-order Markovian and (ii) the correlation coefficient of the input signal tends to the unity [115]. In contrast to the KLT, the definition of the DCT does not depend on the input signal, which allows the development of fast algorithms computationally efficient. Thus, the DCT became widely adopted in image and video compression standards such as JPEG [26], MPEG [27], and HEVC [28], for example.

However, even considering state-of-the-art algorithms, the computational cost of the DCT can still be prohibitive in scenarios of very low processing power or severe restrictions of energy autonomy [74, 81]. In this context, several multiplierless approximations for the DCT have been proposed [71,72,74,80–91,93]. In particular, a widely used approach is to derive approximations based on integer functions [81], such as the signum and rounding functions. Such methodology is employed to derive the *signed DCT* (SDCT) [71] and the *rounded DCT* (RDCT) [72]. In this thesis chapter, we adopt and expand the round-off-based approximation methodology to the KLT case. We aim, therefore, at the proposition of a new class of KLT approximations. Since the transforms are defined deterministically for values of ρ in a predetermined range, our approach addresses the base exchange problem of the KLT.

This chapter is structured as follows. In Section 3.2, we present the mathematical formulation of the KLT for first-order Markov data, the approximation theory, and the design methodology for the proposed approximations. In Section 3.3, the proposed transforms are presented, as well as its assessment measurements. In Subsection 3.3.2, the fast algorithms of the proposed transforms are displayed. Section 3.4 presents the experiments on image compression and Section 3.5 concludes the chapter.

3.2 KLT AND APPROXIMATE TRANSFORMS

3.2.1 KLT for First-Order Markov Signal

Let $\mathbf{x} = [x_0 \quad x_1 \quad \dots \quad x_{N-1}]^\top$ be an N-point random vector. The KLT can be described as an $N \times N$ matrix denoted by $\mathbf{K}_N^{(\rho)}$ that maps $\mathbf{x}$ into the N-point uncorrelated vector $\mathbf{y} = [y_0 \quad y_1 \quad \dots \quad y_{N-1}]^\top$ given by

$$\mathbf{y} = \mathbf{K}_N^{(\rho)} \cdot \mathbf{x}. \tag{3.1}$$

If $\mathbf{x}$ is a first-order Markov signal, then it was shown in [25] that the (i, j)th entry of the KLT matrix for a given value of the correlation coefficient $\rho \in [0, 1]$ is [13]:

$$k_{i,j} = \sqrt{\frac{2}{N+\lambda_i}} \sin\left[\omega_i \left(i - \frac{N-1}{2}\right) + \frac{(j+1)\pi}{2}\right], \tag{3.2}$$

where $i, j = 0, 1, \dots, N-1$,

$$\lambda_i = \frac{1-\rho^2}{1+\rho^2-2\rho\cos\omega_i}, \tag{3.3}$$

and $\omega_1, \omega_2, \ldots, \omega_N$ are the solutions to

$$\tan(N\omega) = \frac{-(1-\rho^2)\sin\omega}{(1+\rho^2)\cos\omega - 2\rho}. \tag{3.4}$$

Since the implementation of the KLT requires floating-point arithmetic, its use has become impractical in real-time applications. In this context, low-complexity approximations for the KLT are viable solutions to circumvent this problem, since its elements generally require only trivial multiplications and bit-shifting operations.

3.2.2 Approximation Theory

Generally, an approximation is a transform $\widehat{\mathbf{T}}$ that behaves similarly to the exact transform according to some specified figure of merit. The design of approximate transforms often requires the approximations to be orthogonal [13]. Indeed, if a matrix is orthogonal, then its inverse is equal to its transpose, and its inverse is ensured to possess low complexity. However, finding orthogonal low-complexity matrices is not always an easy task. In [81], it was shown that if $\mathbf{T}$ is a low-complexity matrix, we can obtain $\widehat{\mathbf{T}}$ through the polar decomposition [156]. Thus, we have that $\widehat{\mathbf{T}} = \mathbf{S}\cdot\mathbf{T}$, where

$$\mathbf{S} = \begin{cases} \sqrt{(\mathbf{T}\cdot\mathbf{T}^\top)^{-1}}, & \text{if } \mathbf{T} \text{ is orthogonal,} \\ \sqrt{[\mathrm{diag}(\mathbf{T}\cdot\mathbf{T}^\top)]^{-1}}, & \text{if } \mathbf{T} \text{ is non-orthogonal,} \end{cases} \tag{3.5}$$

and $\sqrt{\cdot}$ represents the matrix square root operator [156]. Because $\mathbf{S}$ is a diagonal matrix, the computational complexity of $\widehat{\mathbf{T}}$ is the same as that of $\mathbf{T}$, except for the multipliers contained in $\mathbf{S}$. However, the complexity of $\mathbf{S}$ can be absorbed into other sections of a larger procedure, such as the quantization step in the context of image and video compression [72, 74, 76, 83, 120, 131]. In such cases, $\mathbf{S}$ does not contribute to the computational cost [13, 23]. In the Appendix B, we provide a brief derivation showing how the matrix $\mathbf{S}$ can be absorbed into the quantization matrix.

3.2.3 Design Methodology

In a similar fashion as introduced in [72], the low-complexity matrix associated with the RKLT is proposed according to the following expression:

$$\mathbf{T} \triangleq \mathrm{round}(\alpha\cdot\mathbf{K}_N^{(\rho)}), \tag{3.6}$$

where α is an expansion factor [13], and $\mathbf{K}_N^{(\rho)}$ is the N-point KLT matrix for a first-order Markov signal with a given correlation coefficient $0 < \rho < 1$. The rounding function is given by:

$$\text{round}(x) = \lfloor x + 0.5 \rfloor, \tag{3.7}$$

where

$$\lfloor x \rfloor = \max\{m \in \mathbb{Z} | m \leq x\}. \tag{3.8}$$

When applied to a matrix, the round function operates elementwise. To ensure the low-complexity of $\mathbf{T}$, we restrict its entries to the set $\{0, \pm 1\}$. Therefore, α must satisfy the inequality

$$0 \leq \text{round}(\alpha \cdot \gamma) \leq 1, \tag{3.9}$$

where γ is the absolute value of the largest element of the matrix $\mathbf{K}_N^{(\rho)}$. Thus, we have $\alpha \in [0, 3/2\gamma]$.

3.3 PROPOSED APPROXIMATIONS

To numerically derive the proposed RKLT transforms, we adopted the procedure presented in Algorithm 1.

Algorithm 1: Pseudocode for deriving low-complexity matrices.

Input: N, α, step
Output: Set $\mathscr{C}$ of low-complexity matrices

```
C = {}, T = 0_{N×N};
for ρ = step : step : (1 − step) do
    T' := round(α · K_N^(ρ));
    if T' ≠ T then
        C := C ∪ {T'};
        T := T';
    end if
end for
return C
```

The extreme values $\rho = 0$ and $\rho = 1$ were not considered, because they result, respectively, in a degenerate covariance matrix and in the exact DCT matrix, whose approximation theory is covered in [13]. Such methodology is capable of finding approximations for any blocklength N. In this chapter, we focus on the case $N = 8$ due to its wide significance in the image and video coding. In this case, the range of α is approximately $[0, 3.07]$. Thus, we adopted $\alpha = 2$,

agreeing with the methodology [73] employed to derive the RDCT [72]. Exact KLT matrices were obtained for values of $\rho \in [0.1, 0.9]$ with steps of 10^{-1}. Table 5 presents the obtained transforms and their respective diagonals $\mathbf{S}$, as well as the intervals of ρ. Matrix $\mathbf{T}_4$ coincides with the transformation shown in [72].

Table 5 – RKLT approximations

Transform	ρ	Matrix	$\mathbf{S}$
$\mathbf{T}_1$	$(0, 0.4)$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & 0 & -1 & 1 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\operatorname{diag}\left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right)$
$\mathbf{T}_2$	$[0.4, 0.7)$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\operatorname{diag}\left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}\right)$
$\mathbf{T}_3$	$[0.7, 0.8)$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\operatorname{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}\right)$
$\mathbf{T}_4$ [72]	$[0.8, 1)$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\operatorname{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}\right)$

Source: Author (2023).

3.3.1 Assessment Metrics

To evaluate the performance of the proposed transforms, we considered two types of figures of merit: (i) coding measures, such as the unified coding gain [126] and transform efficiency [125], which measure decorrelation and energy compaction; and (ii) proximity measures, such as the mean square error [13] and total error energy [72], which measure similarities between approximate and exact matrices in a Euclidean distance sense. We detailed each of these measures below.

3.3.1.1 Unified Coding Gain

The unified coding gain quantifies the energy compaction capability of the transform $\widehat{\mathbf{T}}$ and is given by [126]

$$\mathrm{Cg}(\widehat{\mathbf{T}}) = 10 \cdot \log_{10} \left\{ \prod_{k=1}^{N} \frac{1}{\sqrt[N]{A_k \cdot B_k}} \right\}, \tag{3.10}$$

where $A_k = \mathrm{su}\left\{ (\mathbf{h}_k^\top \cdot \mathbf{h}_k) \odot \mathbf{R}_\mathbf{x} \right\}$, $\mathbf{h}_k$ is the kth row vector from $\widehat{\mathbf{T}}$, the function $\mathrm{su}(\cdot)$ returns the sum of the elements of its matrix argument, $\odot$ is the Hadamard matrix product operator [127], $\mathbf{R}_\mathbf{x}$ is the autocorrelation matrix of the considered first-order Markov signal, $B_k = \|\mathbf{g}_k\|^2$, $\mathbf{g}_k$ is the kth row vector from $\widehat{\mathbf{T}}^{-1}$, and $\|\cdot\|$ is the Frobenius norm [127].

3.3.1.2 Transform Efficiency

The transform efficiency is another coding related figure of merit, and is given by [125]

$$\eta(\widehat{\mathbf{T}}) = 100 \cdot \frac{\sum_{i=1}^{N} |r_{i,i}|}{\sum_{i=1}^{N} \sum_{j=1}^{N} |r_{i,j}|}, \tag{3.11}$$

where $r_{i,j}$ is the (i,j)th element from $\widehat{\mathbf{T}} \cdot \mathbf{R}_\mathbf{x} \cdot \widehat{\mathbf{T}}^\top$.

3.3.1.3 Mean Square Error

The mean square error (MSE) relative to the exact KLT is given by [13]

$$\mathrm{MSE}(\widehat{\mathbf{T}}) = \frac{1}{N} \cdot \mathrm{tr}\left\{ (\mathbf{K}_N^{(\rho)} - \widehat{\mathbf{T}}) \cdot \mathbf{R}_\mathbf{x} \cdot (\mathbf{K}_N^{(\rho)} - \widehat{\mathbf{T}})^\top \right\}, \tag{3.12}$$

where $\mathrm{tr}(\cdot)$ is the trace function [128].

3.3.1.4 Total Error Energy

Another error measure is the total error energy, which is given by [72]

$$\varepsilon(\widehat{\mathbf{T}}) = \pi \cdot \|\mathbf{K}_N^{(\rho)} - \widehat{\mathbf{T}}\|^2. \tag{3.13}$$

Strictly there is no competing method in the literature that could allow us to make a fair comparison. Thus, we compared the proposed approximation only with the exact KLT for $\rho = 0.3$, 0.4, 0.7, and 0.8. We considered the exact same values of ρ for computing the unified coding gain and the transform efficiency, which depends on the autocorrelation matrix $\mathbf{R}_\mathbf{x}$ of the considered first-order Markov input signal. Thus, we assessed $\widehat{\mathbf{T}}_1$ compared to the $\mathbf{K}^{(0.3)}$ for the

Table 6 – Coding and similarity measures

	ρ	$\mathrm{Cg}(\widehat{\mathbf{T}})$	$\eta(\widehat{\mathbf{T}})$	$\varepsilon(\mathbf{K}^{(\rho)},\widehat{\mathbf{T}})$	$\mathrm{MSE}(\mathbf{K}^{(\rho)},\widehat{\mathbf{T}})$
$\mathbf{K}^{(0.3)}$	0.3	0.3584	100	0	0
$\widehat{\mathbf{T}}_1$	$(0,0.4)$	0.2829	80.7088	1.6751	0.0659
$\mathbf{K}^{(0.4)}$	0.4	0.6626	100	0	0
$\widehat{\mathbf{T}}_2$	$[0.4,0.7)$	0.5616	70.2996	1.7011	0.0660
$\mathbf{K}^{(0.7)}$	0.7	2.5588	100	0	0
$\widehat{\mathbf{T}}_3$	$[0.7,0.8)$	2.1398	65.8777	1.4716	0.0523
$\mathbf{K}^{(0.8)}$	0.8	3.8824	100	0	0
$\widehat{\mathbf{T}}_4$ [72]	$[0.8,1)$	3.4058	74.4747	1.7715	0.0362

Source: Author (2023).

value of $\rho = 0.3$, $\widehat{\mathbf{T}}_2$ compared to the $\mathbf{K}^{(0.4)}$ for the value of $\rho = 0.4$, $\widehat{\mathbf{T}}_3$ compared to the $\mathbf{K}^{(0.7)}$ for the value of $\rho = 0.7$, and $\widehat{\mathbf{T}}_4$ compared to the $\mathbf{K}^{(0.8)}$ for the value of $\rho = 0.8$.

To the best of our knowledge, the literature lacks efficient KLT-based methods for lowly correlated data. We aim at contributing to filling this gap.

Table 6 presents the coding and similarity measurements. As reference values, the total error energy and mean square error for the RDCT [72] compared to the DCT (for $\rho = 0.95$) are, respectively, 1.7945 and 0.0098. Note that for the proximity measures (mean square error and total error energy), the proposed transforms perform well, even better than the performance of the RDCT compared to the DCT considering the total error energy. In this case, the smaller the measurement is, the more similar the approximate transform is to the exact one. Considering the coding measures (unified coding gain and transform efficiency), the proposed transforms assert their good performances when compared with the measurements from the exact KLT for each value of ρ. The exact KLT coding measurements have been used as a benchmark to evaluate the performance of other transforms since it is the unitary optimal transform in terms of energy compaction and decorrelation [13, 14]. It is notable that the performance of the proposed approximations are similar to the exact KLT and have a greatly reduced computational cost, as detailed in the following.

3.3.2 Fast Algorithm and Computational Complexity

Fast algorithms for the approximate transforms can be derived based on the sparse factorization of the transform matrices and butterfly matrix structures [23]. The factorizations of

the proposed transform are given by

$$\mathbf{T}_i = \mathbf{P}_i \cdot \mathbf{A}_{2,i} \cdot \mathbf{A}_1 \quad \text{for} \quad i = 1,2,3,4, \tag{3.14}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{I}_4 & \bar{\mathbf{I}}_4 \\ \bar{\mathbf{I}}_4 & -\mathbf{I}_4 \end{bmatrix}, \quad \mathbf{A}_{2,i} = \begin{bmatrix} \mathbf{B}_{2,i} & \\ & \mathbf{B}_2 \end{bmatrix}, \tag{3.15}$$

$$\mathbf{B}_2 = \begin{bmatrix} -1 & -1 & 0 & 1 \\ -1 & 1 & -1 & 0 \\ 1 & 0 & -1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \tag{3.16}$$

and $\mathbf{I}_4$ and $\bar{\mathbf{I}}_4$ are, respectively, the identity and counter-identity matrices of order 4. Matrices $\mathbf{B}_{2,i}$ are given by

$$\mathbf{B}_{2,1} = \begin{bmatrix} 1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 0 \\ 1 & 0 & -1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{B}_{2,2} = \begin{bmatrix} 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}, \tag{3.17}$$

$$\mathbf{B}_{2,3} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \tag{3.18}$$

$$\mathbf{B}_{2,4} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}. \tag{3.19}$$

The permutation matrices $\mathbf{P}_i$ are

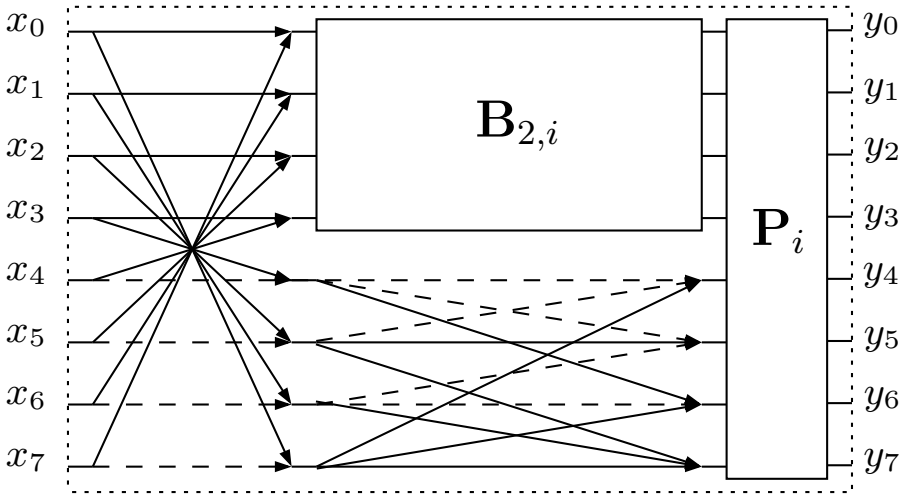
$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \tag{3.20}$$

and

$$\mathbf{P}_3 = \mathbf{P}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{3.21}$$

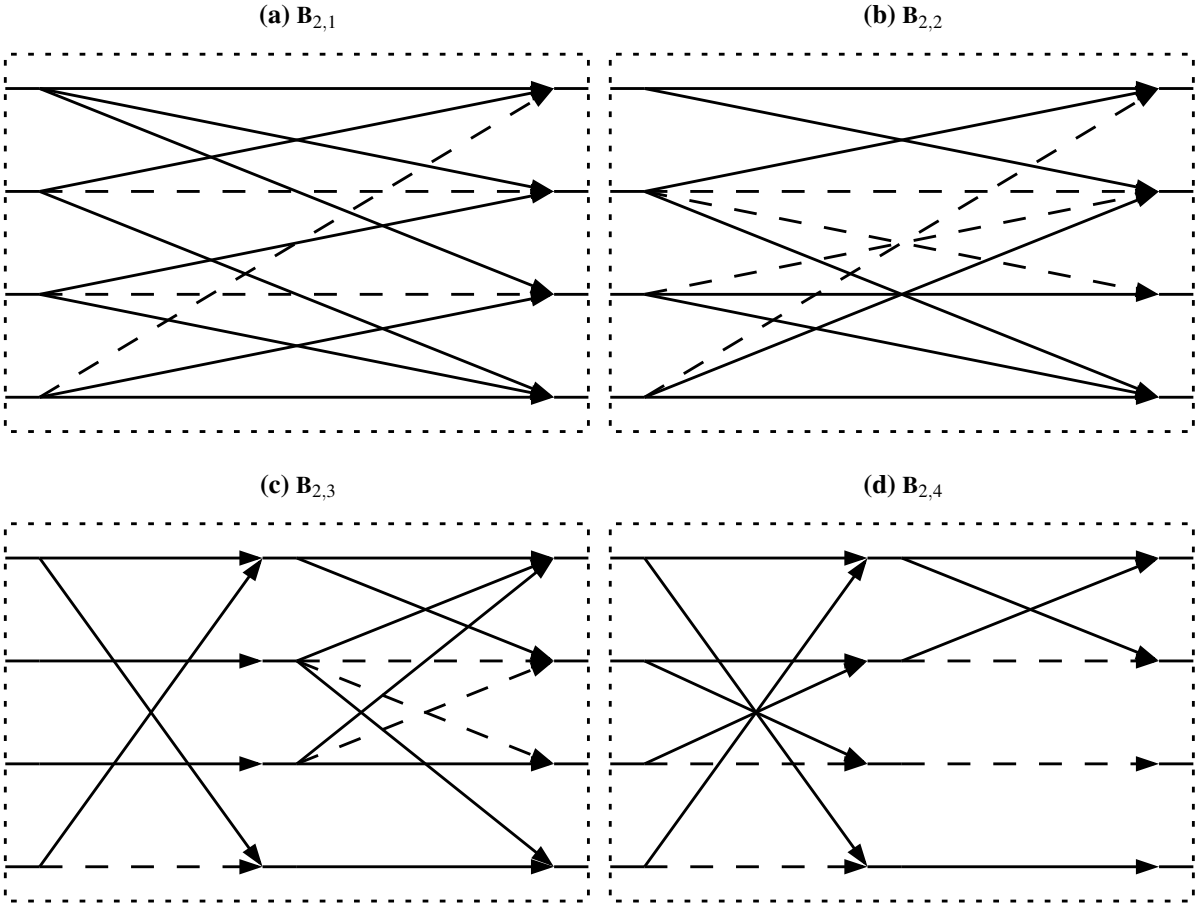
Fig. 17 and 18 show the SFG of the fast algorithms. The dashed arrows represent multiplication by -1.

Figure 17 – SFG of the proposed transforms. Block $\mathbf{B}_{2,i}$ is different for each transform and it is presented in Fig. 18.



Source: Author (2023).

Figure 18 – Block $\mathbf{B}_{2,i}$ of each proposed transform.



Source: Author (2023).

The direct implementation of the exact KLT requires 56 additions and 64 multiplications. The proposed transforms are designed to be multiplierless but still require 56 additions in its direct implementation. Considering the fast algorithms proposed for the transforms, we have a reduction in the addition operations relative to the direct implementation. Matrices $\mathbf{T}_1$, $\mathbf{T}_2$, and $\mathbf{T}_3$ require only 24 additions, causing a reduction of 57.17% in the number of addition operations, and matrix $\mathbf{T}_4$ requires 22 additions, with a reduction of 60.71%.

3.4 IMAGE COMPRESSION

In this section, the performance of the proposed transforms is assessed in the context of image compression [12] as suggested in [71, 72, 74, 80, 81, 84, 85].

3.4.1 JPEG-like Compression

The following compression scheme [16] was applied to standardized images obtained from the public image bank available in [137]. Input images were divided into disjoint sub-

blocks of size 8×8. The 2D direct and inverse transformations induced by $\mathbf{K}_N^\rho$ are computed, respectively, by [53]

$$\mathbf{B} = \mathbf{K}_N^\rho \cdot \mathbf{A} \cdot (\mathbf{K}_N^\rho)^{-1}, \tag{3.22}$$

$$\mathbf{A} = (\mathbf{K}_N^\rho)^{-1} \cdot \mathbf{B} \cdot \mathbf{K}_N^\rho, \tag{3.23}$$

where $\mathbf{A}$ and $\mathbf{B}$ are square matrices of size N. Each sub-block was submitted to the 2D transform computation and the resulting transform domain coefficients were reordered using the standard zig-zag sequence [16]. Only the initial r coefficients in each sub-block were retained and the remaining coefficients were zeroed. The 2D inverse transform was applied and the reconstructed sub-blocks were adequately rearranged. Original and compressed images were then evaluated considering traditional quality assessment measures. The considered figures of merit for image quality evaluation were: (i) the Mean Structural Similarity Index (MSSIM) [136]; (ii) the MSE [13]; (iii) and the Peak Signal-to-Noise Ratio (PSNR) [135]. Even though the PSNR and MSE are very popular figures of merit, it was shown in [157] that they might offer limited results as image quality assessment tools for it poorly correlates with human perception. On the other hand, the MSSIM is capable of closely capturing the image quality as understood by the human visual system model [136].

The image compression experiments were divided into two analysis: (i) a qualitative one, based on compressed *Lena*, *Baboon*, and *Moon* images with approximately 77% of compression rate; (ii) and a quantitative analysis, based on the average measures of 45 standardized 8-bit compressed images [137] for a wide range of retained coefficients (r). The results are presented below.

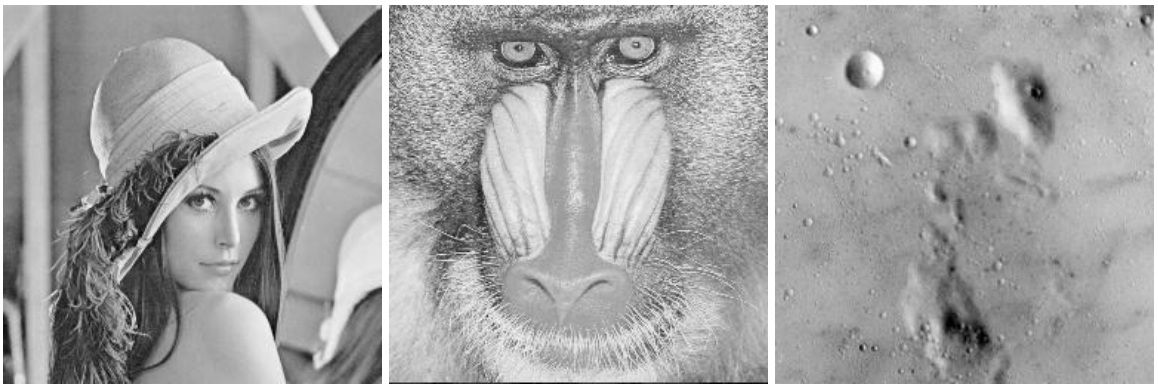
3.4.2 Results

For the qualitative analysis, we considered three known public images available on [137]. The original grayscale images are presented in Fig. 19.

Figs. 20, 21, and 22 illustrate qualitatively the reconstruction of the *Lena*, *Baboon*, and *Moon* images, after the application of the compression scheme for $r = 15$ (compression rate of approximately 77%) using the proposed RKLT, the exact KLT for $\rho = 0.3$, 0.4, 0.7, 0.8, and the exact DCT. The assessment metrics from each compressed image are presented in Table 7. We highlighted the results from the proposed transforms which performed better than the exact KLT for the value of ρ associated with the interval of which the approximate transform was derived. Approximations $\widehat{\mathbf{T}}_2$ and $\widehat{\mathbf{T}}_3$ outperformed the exact KLT, $\mathbf{K}^{(0.4)}$ and $\mathbf{K}^{(0.7)}$ respectively,

Figure 19 – Original images.

(a) *Lena* **(b)** *Baboon* **(c)** *Moon*



Source: SIPI [137].

Table 7 – Quality image assessment measurements for *Lena*, *Baboon*, and *Moon* compressed images

Image	Lena			Baboon			Moon		
Transform	MSE	PSNR	MSSIM	MSE	PSNR	MSSIM	MSE	PSNR	MSSIM
$\widehat{\mathbf{T}}_1$	3198.18	13.082	0.154	3505.908	12.683	0.220	3114.159	13.197	0.096
$\mathbf{K}^{(0.3)}$	95.674	28.323	0.660	322.113	23.051	0.685	112.060	27.636	0.598
$\widehat{\mathbf{T}}_2$	48.729	31.253	**0.913**	313.852	23.164	**0.761**	**57.287**	**30.550**	**0.781**
$\mathbf{K}^{(0.4)}$	70.215	29.666	0.720	296.260	23.414	0.716	87.816	28.695	0.653
$\widehat{\mathbf{T}}_3$	49.071	31.222	**0.913**	313.788	23.164	0.761	57.276	30.551	**0.781**
$\mathbf{K}^{(0.7)}$	30.464	33.293	0.884	257.115	24.030	0.783	50.058	31.136	0.776
$\widehat{\mathbf{T}}_4$ [72]	44.593	31.638	0.917	286.597	23.558	0.766	52.478	30.931	0.789
$\mathbf{K}^{(0.8)}$	25.907	33.997	0.916	253.406	24.093	0.793	45.705	31.531	0.796
DCT	23.867	34.353	0.938	254.233	24.078	0.796	43.543	31.742	0.807

Source: Author (2023).

according to the values of MSE, PSNR, and MSSIM. The approximations may outperform the exact KLT because we measure the overall performance of the entire image compression system, which includes particular nonlinearities that are better suited for the approximate computation.

For the quantitative analysis, we considered the average image quality measurements of 45 compressed standardized images [137] considering different levels of compression ($r \in (0,45)$). Fig. 23 presents the average image quality measurements from the compressed images considering the approximate transforms and the exact KLT for values of $\rho = 0.3$, 0.4, 0.7, and 0.8. The approximate transforms perform similarly to the exact KLT, mainly when we retain more than $r = 15$ retained coefficients, except for $\widehat{\mathbf{T}}_1$. Approximation $\widehat{\mathbf{T}}_4$ outperformed the exact KLT ($\rho = 0.8$) for $r \in [1,11]$ considering PSNR values and for $r \in [1,14]$ considering the MSSIM values. Considering the performance in JPEG-like compression the proposed approximations exhibited relevant results, showing a good balance between performance and computational cost.

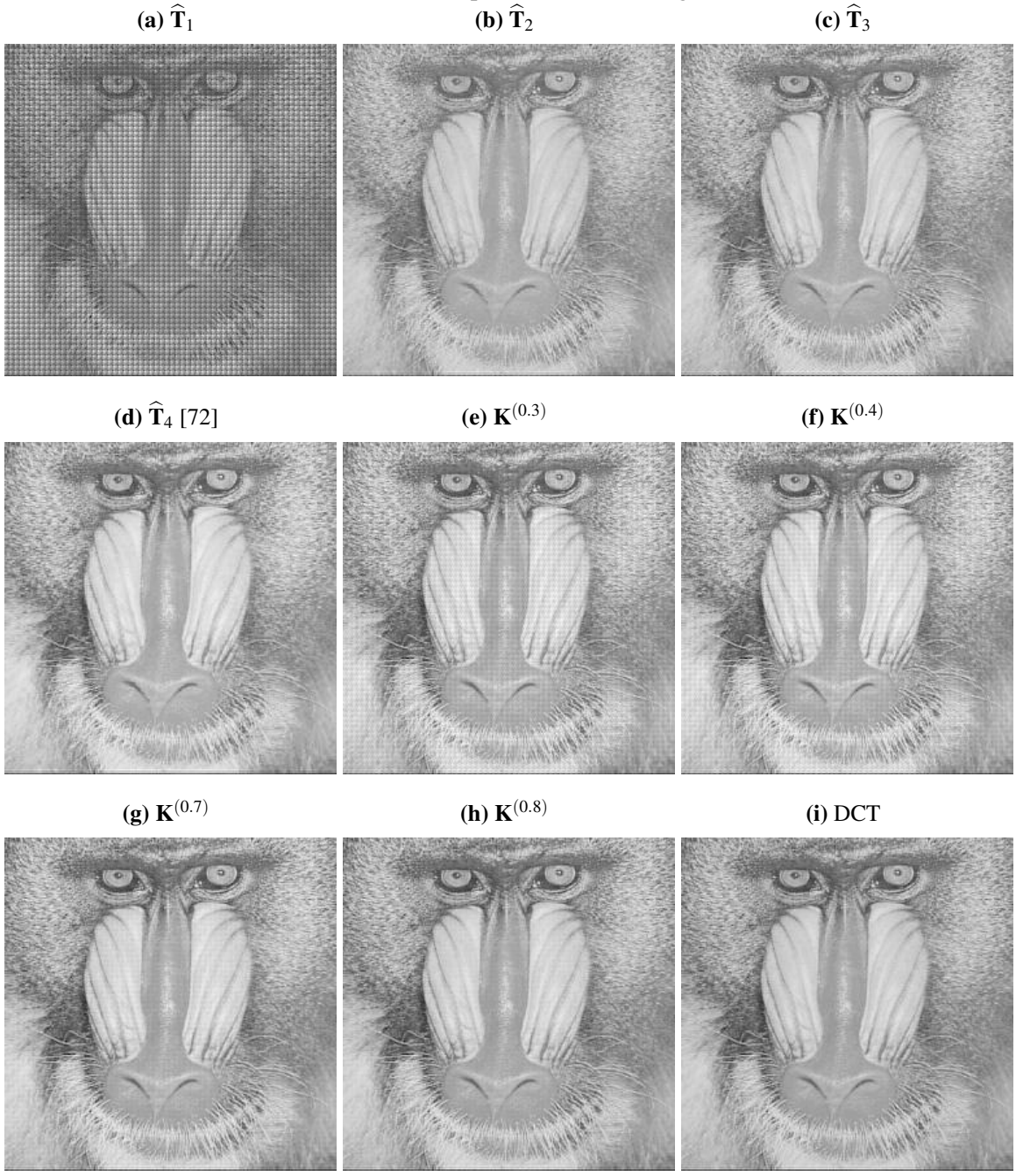
Figure 20 – Compressed *Lena* images.

Source: Author (2023).

3.5 CONCLUSIONS

In this chapter, we introduced a methodology based on the rounding-off function to design low-complexity approximations for the Karhunen-Loève transform (KLT). Due to its relevance in practical image coding systems, the special case $N = 8$ was comprehensively examined. According to qualitative and quantitative computational experiments, the proposed transforms were shown to be good approximations for the KLT as measured by the adopted quality measures: MSE, PSNR, and MSSIM. The low-complexity matrices are natural candidates for the design

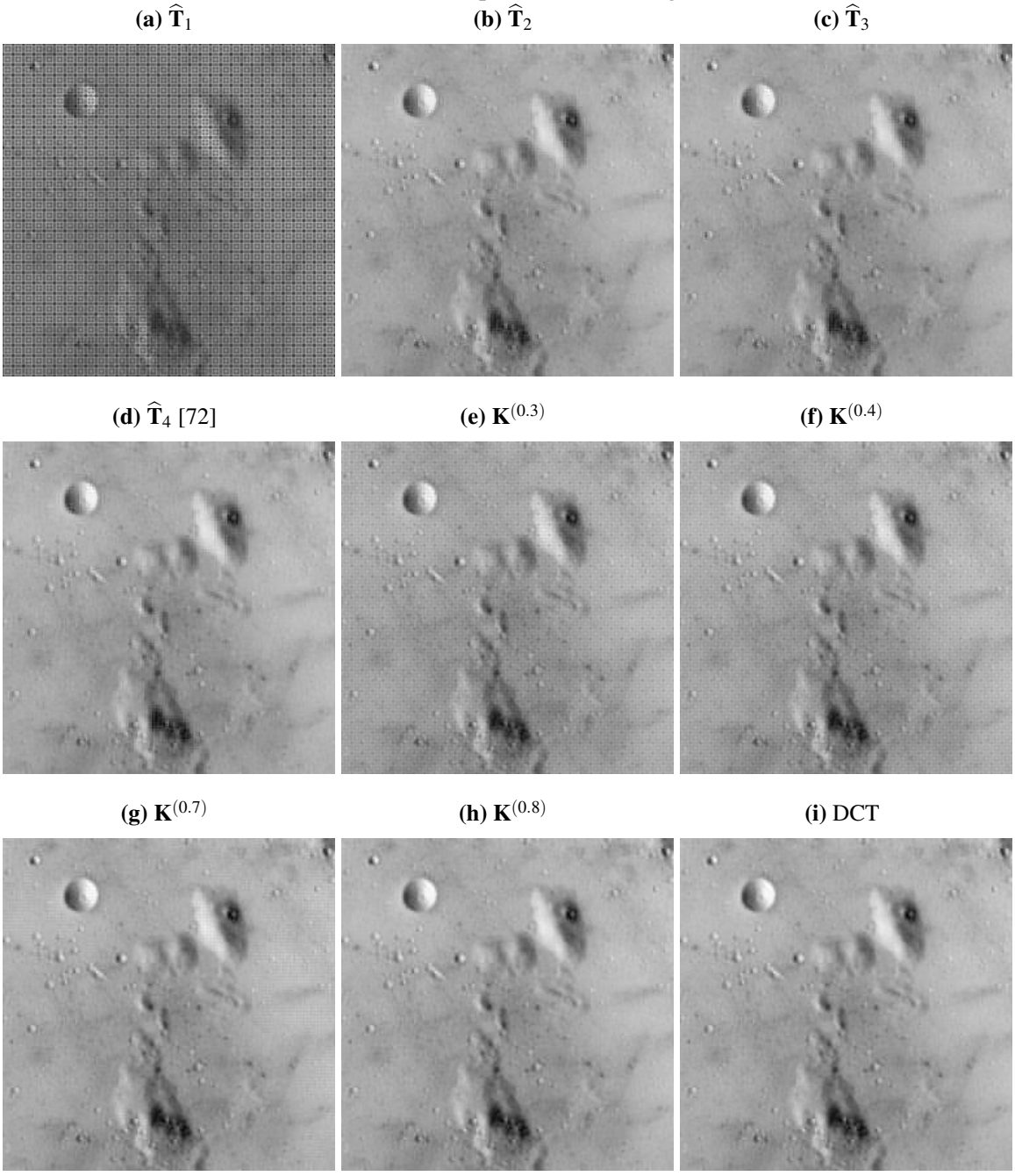
Figure 21 – Compressed *Baboon* images.



Source: Author (2023).

of efficient hardware implementation capable of operating at low power consumption and high performance.

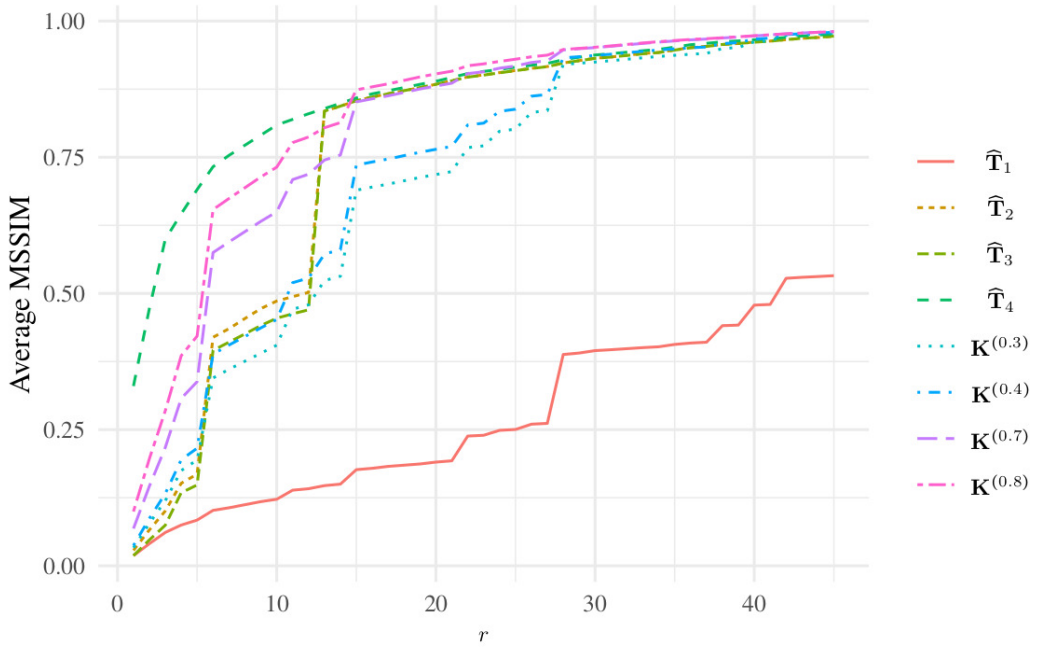
Figure 22 – Compressed *Moon* images.



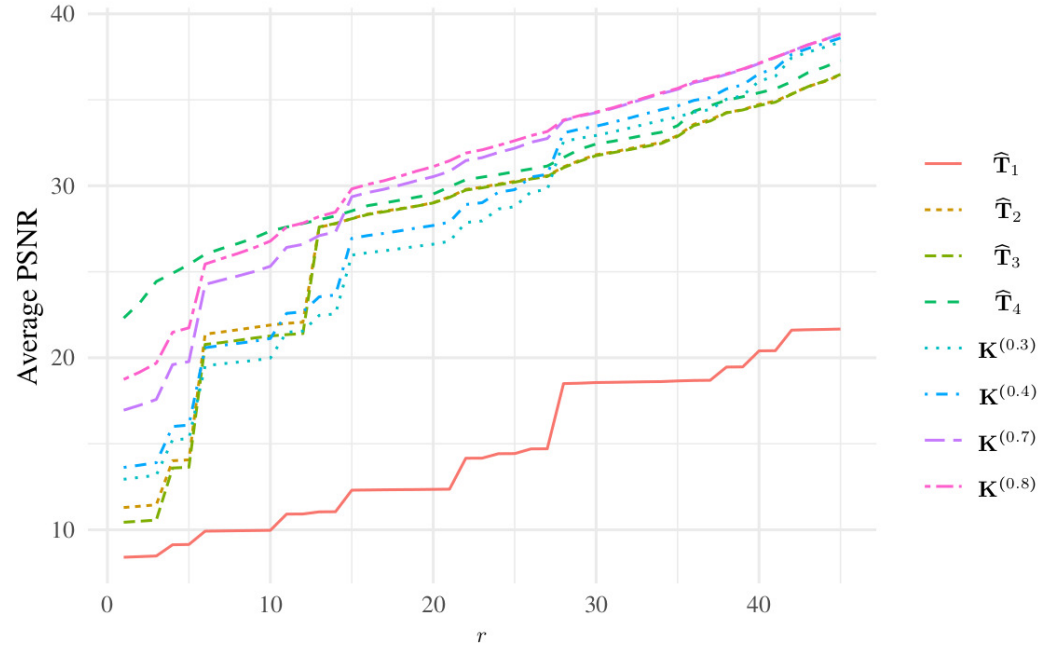
Source: Author (2023).

Figure 23 – Image quality measurements for different levels of compression.

(a)



(b)



Source: Author (2023).

4 FAST DATA-INDEPENDENT KLT APPROXIMATIONS BASED ON INTEGER FUNCTIONS

4.1 INTRODUCTION

Among the discrete tranforms, the Karhunen-Loève transform (KLT) is an optimal linear tool for data decorrelation, being capable of concentrating the signal energy in few transform-domain coefficients [14]. Because the KLT minimizes the mean square error of compressed data, it can greatly reduce data dimensionality and can be regarded as the ideal transform for image compression [13, 14]. Despite such good properties, the KLT finds few practical applications [25, 154, 158–161], mainly due to the fact that the definition of the KLT matrix relies on the covariance matrix of the input signal. Therefore, in general, two different input signals would effect two different transformation matrices, thus, precluding the design of efficient approaches for computing the transformed signal. Moreover, because of the data dependency, generally, the KLT suffers from the dictionary exchange problem [14, 94], i.e., the transformation basis is not known *a priori* by the decoder.

For specific classes of signals, such as first-order Markovian processes with known correlation coefficient ρ, the KLT matrix can be expressed simply in terms of ρ [114]. Nevertheless, the complexity of the resulting transformation matrix remains in $\mathcal{O}(N^2)$, where N is the signal length [14] and, in general, there are no efficient fast algorithms available for its computation [23, 25, 95, 162, 163]. . In this context, several KLT approximations [94–99] and fast algorithms for the KLT [25, 41–44] have been proposed so they have a lower computational cost. However, these methods still suffer the problem of data-dependency.

For the particular—but very relevant—case where $\rho \to 1$, the KLT assumes the mathematical definition of the DCT [115]. In other words, the DCT is the KLT for highly autocorrelated first-order Markovian data [13, 14, 164]. Such model fairly captures the structure of natural images, which is typically assumed to admit $\rho = 0.95$ [12]. Being fully independent of the data [13, 14], efficient methods for computing the DCT can be derived [45, 60], such as the Loeffler algorithm and the Chen algorithm, turning it into a central tool for image and video coding [26–28]. However, even at the reduced computational cost offered by fast algorithms, the residual complexity of the DCT might still be sufficiently large to preclude its application in contexts where severe restrictions on computational processing power and/or energy autonomy are present, such as in wireless and satellite communication systems and in portable computing applications [13]. This reality opened the path for the design of extremely low-complexity

methods for the DCT estimation based on approximate integer transforms. Hence, several approximations for the DCT have been proposed, generally being multiplierless transforms that require addition and bit-shifting operations only [71, 72, 74, 80–93]. In particular, we cite the following approximations for the DCT based on integer functions: the signed DCT (SDCT) [71], the rounded DCT (RDCT) [72], and the collection of integer DCT approximations detailed in [81]. Despite the very low computational requirements, such approximations can still offer good coding performance and constitute realistic alternatives to the exact DCT.

Considering the above discussion and taking into account the following major aspects:

- the current literature lacks specific methodologies for the low-complexity computation of the KLT, mainly when considering low-complexity approximation transforms for mid- and low-correlated signals;
- the methods for the KLT evaluation [25,41–44,94–98,154,155,165] exhibit data-dependency which entail severe difficulties in designing fast algorithms based on matrix factorization;
- the dictionary exchange problem presented in the KLT and its approximations [13, 14];
- the proven success of matrix approximation theory for deriving low-complexity DCT methods found in the literature;
- and the fact that the KLT is the optimal linear transform in terms of decorrelation of first-order Markovian signals;

we aim at proposing matrix approximations for the KLT with the following properties: (i) data independence and closed-form expression; (ii) symmetrical structure that leads to sparse matrix factorizations and fast algorithms, and (iii) suitability for first-order Markovian processes at a wide range of correlation coefficient. **To the best of our knowledge, there are no approximations for the KLT in the literature that fill these gaps.** To obtain the sought KLT approximations, we consider integer-based approximation methods as shown in [71–73, 81].

This chapter is structured as follows. In Section 4.2, we present the mathematical formulation of the KLT for first-order Markovian data, a brief review of approximation theory for discrete transforms, and the assessment metrics used for the evaluation of approximate transforms. Section 4.3 introduces the optimization problem, search space, objective function, and the methodology used to obtain the proposed transforms. The proposed transforms are presented in Section 4.4 and the fast algorithms and their computational complexities are displayed in Section 4.5. Section 4.6 presents the experiments on image compression. In Section 4.7 a FPGA design is proposed and compared with competing methods. Finally, Section 4.8 concludes the chapter.

4.2.1 KLT for First-Order Markovian Signal

The KLT maps an N-point input signal

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \dots & x_{N-1} \end{bmatrix}^\top \tag{4.1}$$

into an N-point uncorrelated signal

$$\mathbf{y} = \begin{bmatrix} y_0 & y_1 & \dots & y_{N-1} \end{bmatrix}^\top \tag{4.2}$$

according to

$$\mathbf{y} = \mathbf{K} \cdot \mathbf{x}, \tag{4.3}$$

where $\mathbf{K}$ is the KLT matrix [21, 22]. If $\mathbf{x}$ is a first-order Markovian signal, then its covariance matrix is given by [13]

$$\mathbf{R}_\mathbf{x} = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \dots & \rho & 1 \end{bmatrix}, \tag{4.4}$$

where $0 \leq \rho \leq 1$ is the autocorrelation parameter of the Markovian process. Therefore, the (i, j)th entry of the KLT matrix is [13]

$$k_{i,j} = \sqrt{\frac{2}{N+\lambda_i}} \sin\left[\omega_i \left(i - \frac{N-1}{2}\right) + \frac{(j+1)\pi}{2}\right], \tag{4.5}$$

where $i, j = 0, 1, \dots, N-1$,

$$\lambda_i = \frac{1-\rho^2}{1+\rho^2-2\rho\cos\omega_i}, \tag{4.6}$$

and $\omega_1, \omega_2, \dots, \omega_N$ are the solutions to [13]

$$\tan(N\omega) = \frac{-(1-\rho^2)\sin\omega}{(1+\rho^2)\cos\omega - 2\rho}. \tag{4.7}$$

4.2.2 Approximation Theory

Generally, an approximation $\widehat{\mathbf{K}}$ is based on a low-complexity matrix $\mathbf{T}$, such that [80, 81, 123, 166]

$$\widehat{\mathbf{K}} = \mathbf{S} \cdot \mathbf{T}, \tag{4.8}$$

and

$$\mathbf{S} = \begin{cases} \sqrt{(\mathbf{T} \cdot \mathbf{T}^{\top})^{-1}}, & \text{if } \mathbf{T} \text{ is orthogonal,} \\ \sqrt{[\text{diag}(\mathbf{T} \cdot \mathbf{T}^{\top})]^{-1}}, & \text{if } \mathbf{T} \text{ is non-orthogonal,} \end{cases} \tag{4.9}$$

where $\text{diag}(\cdot)$ is the diagonal matrix generated by its arguments.

Thus, we focus our search on matrix $\mathbf{T}$. The low-complexity matrix $\mathbf{T}$ can be obtained by restricting its elements over sets whose entries possess very low multiplicative complexity, such as $\{0, \pm 1, \pm 2, \ldots\}$, $\{0, \pm 1/2, \pm 1, \pm 2, \ldots\}$, $\{0, \pm 1, \pm 2, \pm 3, \ldots\}$, among others. As a matter of fact, multiplications by powers of two require only bit-shifting operations and a multiplication by 3 can be implemented by means of one addition and one bit-shifting operation. One possible way of restricting the entries of matrix $\mathbf{T}$ is applying integer functions to the elements of the exact transform, as described in [71, 72, 81].

Common integer functions employed to derive new transform approximations are the floor, ceiling, truncation (round towards zero), and round-away-from-zero functions, defined respectively as

$$\text{floor}(x) = \lfloor x \rfloor = \max\{m \in \mathbb{Z} | m \leq x\}, \tag{4.10}$$

$$\text{ceil}(x) = \lceil x \rceil = \min\{n \in \mathbb{Z} | n \geq x\}, \tag{4.11}$$

$$\text{trunc}(x) = \text{sign}(x) \cdot \lfloor |x| \rfloor, \tag{4.12}$$

$$\text{round}_{\text{AFZ}}(x) = \text{sign}(x) \cdot \lceil |x| \rceil, \tag{4.13}$$

where $|\cdot|$ is the absolute value of its argument. Besides the above integer functions, we also consider the variants of the round-off function as defined next [167]:

$$\text{round}_{\text{HU}}(x) = \left\lfloor x + \frac{1}{2} \right\rfloor, \tag{4.14}$$

$$\text{round}_{\text{HD}}(x) = \left\lceil x - \frac{1}{2} \right\rceil, \tag{4.15}$$

$$\text{round}_{\text{HAFZ}}(x) = \text{sign}(x) \cdot \left\lfloor |x| + \frac{1}{2} \right\rfloor, \tag{4.16}$$

$$\text{round}_{\text{HTZ}}(x) = \text{sign}(x) \cdot \left\lceil |x| - \frac{1}{2} \right\rceil, \tag{4.17}$$

$$\text{round}_{\text{EVEN}}(x) = \begin{cases} \left\lfloor x - \frac{1}{2} \right\rfloor, & \text{if } \frac{2x-1}{4} \in \mathbb{Z}, \\ \left\lfloor x + \frac{1}{2} \right\rfloor, & \text{otherwise}, \end{cases} \tag{4.18}$$

$$\text{round}_{\text{ODD}}(x) = \begin{cases} \left\lfloor x + \frac{1}{2} \right\rfloor, & \text{if } \frac{2x-1}{4} \in \mathbb{Z}, \\ \left\lfloor x - \frac{1}{2} \right\rfloor, & \text{otherwise}. \end{cases} \tag{4.19}$$

Hereafter we denote this group of rounding functions as round_{NI}.

4.2.3 Assessment Metrics

The performance measures usually employed for assessing approximate transforms can be categorized in two types: (i) coding measures, such as the coding gain [126] and transform efficiency [125], which measure the power of energy decorrelation and compaction; and (ii) proximity measures relative to the exact transform, such as mean square error [13] and total energy error [72], which quantify similarities between the approximate matrices and the exact transform in a Euclidean distance sense. Such figures of merit are presented next.

4.2.3.1 Unified Coding Gain

The unified coding gain measures the energy compaction capacity and is given by [126]:

$$\text{C}_g(\widehat{\mathbf{K}}) = 10 \cdot \log_{10} \left\{ \prod_{k=1}^{N} \frac{1}{\sqrt[N]{A_k \cdot B_k}} \right\}, \tag{4.20}$$

where $A_k = \text{su}\left\{ (\mathbf{h}_k^\top \cdot \mathbf{h}_k) \odot \mathbf{R}_\mathbf{x} \right\}$, $\mathbf{h}_k$ is the kth row vector from $\widehat{\mathbf{K}}$, function $\text{su}(\cdot)$ gives the sum of the elements of its matrix argument, $\odot$ is the Hadamard matrix product [127], $\mathbf{R}_\mathbf{x}$ is the autocorrelation matrix from a first-order Markovian signal, $B_k = \|\mathbf{g}_k\|_\text{F}^2$, $\mathbf{g}_k$ is the kth row vector from $\widehat{\mathbf{K}}^{-1}$, and $\|\cdot\|_\text{F}$ is the Frobenius norm [127].

4.2.3.2 Transform Efficiency

The transform efficiency is given by [125]:

$$\eta(\widehat{\mathbf{K}}) = 100\frac{\sum_{i=1}^{N}|r_{i,i}|}{\sum_{i=1}^{N}\sum_{j=1}^{N}|r_{i,j}|}, \tag{4.21}$$

where $r_{i,j}$ is the (i,j)th element from $\widehat{\mathbf{K}}\cdot\mathbf{R}_{\mathbf{x}}\cdot\widehat{\mathbf{K}}^{\top}$.

4.2.3.3 Mean square Error

The mean square error between the exact and approximate transforms is defined as [13]:

$$\mathrm{MSE}(\mathbf{K},\widehat{\mathbf{K}}) = \frac{1}{N}\cdot\mathrm{tr}\left\{(\mathbf{K}-\widehat{\mathbf{K}})\cdot\mathbf{R}_{\mathbf{x}}\cdot(\mathbf{K}-\widehat{\mathbf{K}})^{\top}\right\}, \tag{4.22}$$

where $\mathrm{tr}(\cdot)$ is the trace function [128].

4.2.3.4 Total Error Energy

The total error energy measures the similarity between the approximate and the exact transform matrix, according to [72]:

$$\varepsilon(\mathbf{K},\widehat{\mathbf{K}}) = \pi\cdot\|\mathbf{K}-\widehat{\mathbf{K}}\|_{\mathrm{F}}^{2}. \tag{4.23}$$

4.3 OPTIMAL PROPOSED TRANSFORMS

In this section, we detail a systematic computational procedure to obtain the sought approximations. The methodology consists of solving optimization problems according to the measures discussed in the previous section.

4.3.1 Search Space

For the computational search, we set the elements of the matrices to be in the set of low-complexity entries $\mathscr{C} = \{0, \pm 1, \pm 2, \pm 3\}$, because multiplying these elements only requires addition and bit-shifting operations. For the blocklength we considered $N = 8$, due to its importance in image compression. Thus, we have $7^8 = 5764801$ candidate matrices to be considered in the optimization problem for each value of ρ of the KLT matrix.

The transform search space can be formally defined as follows. Let

$$\widehat{\mathbf{K}} = \sqrt{[\mathrm{diag}(\mathbf{T}\cdot\mathbf{T}^{\top})]^{-1}}\cdot\mathbf{T}, \tag{4.24}$$

where $\mathbf{T} \in \mathscr{M}_{\mathscr{C}}(8)$ and $\mathscr{M}_{\mathscr{C}}(8)$ is the 8×8 matrix space with elements in the set $\mathscr{C} = \{0, \pm 1, \pm 2, \pm 3\}$. We propose to search a subset of $\mathscr{M}_{\mathscr{C}}(8)$

$$\mathscr{E}_{\alpha} = \{\mathbf{T} \in \mathscr{M}_{\mathscr{C}}(8) : \mathbf{T} = \text{int}(\alpha \cdot \mathbf{K}^{(\rho)})\}, \tag{4.25}$$

where

$$\text{int} \in \{\text{floor}, \text{ceil}, \text{trunc}, \text{round}_{\text{AFZ}}, \text{round}_{\text{NI}}\}, \tag{4.26}$$

and α is the expansion factor [13, 81, 168]. The ranges of α must satisfy the inequality

$$0 \leq \text{int}(\alpha \cdot \gamma) \leq 3, \tag{4.27}$$

where γ is the absolute value of the largest element of the matrix $\mathbf{K}^{(\rho)}$. Considering the integer functions floor, ceil, trunc, $\text{round}_{\text{AFZ}}$, round_{NI}, the ranges of α ($\mathscr{A}$) are given, respectively, by $(1/\gamma, 4/\gamma)$, $(0, 3/\gamma)$, $(1/\gamma, 4/\gamma)$, $(0, 3/\gamma)$, and $(0, 3/\gamma)$. Therefore, the search space is

$$\mathscr{E} = \bigcup_{\alpha \in \mathscr{A}} \mathscr{E}_{\alpha}. \tag{4.28}$$

4.3.2 Objective Function

In order to search for the optimal transforms according to the considered metrics, the following optimization problem was proposed:

$$\widehat{\mathbf{K}}^{*} = \arg \underset{\widehat{\mathbf{K}}}{\text{opt}}\, f(\widehat{\mathbf{K}}), \tag{4.29}$$

where $\widehat{\mathbf{K}}$ is a candidate matrix for solving the problem, and

$$f \in \{\text{C}_g(\cdot), \eta(\cdot), \text{MSE}(\mathbf{K}^{(\rho)}, \cdot), \varepsilon(\mathbf{K}^{(\rho)}, \cdot)\}, \tag{4.30}$$

which are the figures of merit to be optimized. For $\text{C}_g(\cdot)$ and $\eta(\cdot)$, the optimization problem is of the maximization type; whereas, for the $\text{MSE}(\cdot)$ and $\varepsilon(\cdot)$, it is of the minimization type.

4.3.3 Methodology

Once the optimization problem, restrictions, search space, and the objective function are established, we exhaustively compute (4.29) for the specific values of α within the intervals defined by each integer function, with steps of 10^{-2}. To compute the exact KLT matrix, $\mathbf{K}^{(\rho)}$, we considered $0 < \rho < 1$ with steps of 10^{-1}. The pseudo-code for the adopted procedure is presented in Algorithm 2.

Algorithm 2: Pseudocode for deriving low-complexity matrices.

Input: ρ-step, α-step, int
Output: Set $\mathscr{C}$ of low-complexity matrices
$\mathscr{C} = \{\}$, $\mathscr{A}$ (range of α), $\widehat{\mathbf{K}} = \mathbf{0}_{N\times N}$, $\mathbf{M} = \mathbf{0}_{4\times N}$;
for $\rho \in (0,1)$, ρ-step **do**
  for $\alpha \in \mathscr{A}$, α-step **do**
    $\mathbf{T}' := \text{int}(\alpha \cdot \mathbf{K}^{(\rho)})$;
    $\widehat{\mathbf{K}}' = \sqrt{[\text{diag}(\mathbf{T}' \cdot \mathbf{T}'^{\top})]^{-1}} \cdot \mathbf{T}'$;
    $\mathbf{M} = [\text{C}_g(\widehat{\mathbf{K}}'), \eta(\widehat{\mathbf{K}}'), \text{MSE}(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}}'), \varepsilon(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}}')]$;
  end for
  if $\widehat{\mathbf{K}} = \arg\text{opt}_{\widehat{\mathbf{K}}'}(\mathbf{M})$ **then**
    $\mathscr{C} := \mathscr{C} \cup \widehat{\mathbf{K}}'$;
  end if
end for
return $\mathscr{C}$

This search results in 144 matrices. Here, we are considering the discussed four figures of merit and evaluating each one separately, so we can have up to four optimal transforms for each fixed interval of ρ and integer function. Among the 144 obtained matrices, it is expected that several of them show similar performance. Therefore, we aim now at refining the set of 144 matrices so we could identify a reduced set of matrices that are representative over the range $\rho \in (0,1)$.

In this sense, we propose a two-step procedure. In the first step, we only consider, among the 144 transforms, those that obtained the best performance, for the values of $\rho \in (0,1)$ with steps of 10^{-1}, according to each figure of merit. This procedure caused a reduction of 86.11% in the number of transforms. Table 8 presents the 20 optimal transforms found in the first stage of refinement. The similarity measurements were obtained considering the exact KLT for the value of ρ of the upper limit of each interval for which the approximation was derived, i.e., $\widehat{\mathbf{K}}_1$ were compared to the exact KLT for $\rho = 0.1$.

In the second stage of the refinement, we aim to group intervals of ρ in which the transforms exhibit similar performance according to the unified coding gain, since this metric presents information about the coding capacity of the orthogonal transformation for applications of data compression. These groups can be obtained according to a clustering procedure, such as the *k*-means [169]. Using a clustering method can result in a reduced number of groups in which only one matrix can be chosen as representative of the group.

Hence, applying the *k*-means clustering method considering the values of the unified coding gain of each transform, we obtained two distinct groups, C_1 and C_2. The first group

Table 8 – Coding and similarity measures from the obtained optimal transforms

Transform	Function	ρ interval	$C_g(\widehat{\mathbf{K}})$	$\eta(\widehat{\mathbf{K}})$	$\varepsilon(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$	$\text{MSE}(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$
$\widehat{\mathbf{K}}_1$	$\text{round}_{\text{AFZ}}$	$(0, 0.1]$	0.0308	93.4298	1.5331	0.0608
$\widehat{\mathbf{K}}_2$	$\text{round}_{\text{AFZ}}$	$(0, 0.1]$	0.1325	79.5971	0.3173	0.0128
$\widehat{\mathbf{K}}_3$	trunc	$(0, 0.1]$	0.0588	88.3104	0.093	0.0036
$\widehat{\mathbf{K}}_4$	trunc	$(0.1, 0.2]$	0.1754	83.7756	0.2265	0.0094
$\widehat{\mathbf{K}}_5$	trunc	$(0.2, 0.3]$	0.3461	80.8238	0.2999	0.0132
$\widehat{\mathbf{K}}_6$	$\text{round}_{\text{AFZ}}$	$(0.3, 0.4]$	0.6725	83.0728	0.3104	0.0095
$\widehat{\mathbf{K}}_7$	trunc	$(0.3, 0.4]$	0.7618	63.712	2.1348	0.0785
$\widehat{\mathbf{K}}_8$	trunc	$(0.3, 0.4]$	0.6532	83.3729	0.2823	0.0115
$\widehat{\mathbf{K}}_9$	$\text{round}_{\text{AFZ}}$	$(0.4, 0.5]$	1.1063	87.2737	0.3487	0.0094
$\widehat{\mathbf{K}}_{10}$	trunc	$(0.4, 0.5]$	1.153	77.5984	0.6439	0.0163
$\widehat{\mathbf{K}}_{11}$	$\text{round}_{\text{AFZ}}$	$(0.5, 0.6]$	1.7572	82.7462	0.9273	0.0197
$\widehat{\mathbf{K}}_{12}$	$\text{round}_{\text{AFZ}}$	$(0.5, 0.6]$	1.6743	86.4929	0.275	0.0089
$\widehat{\mathbf{K}}_{13}$	$\text{round}_{\text{AFZ}}$	$(0.6, 0.7]$	2.5736	84.7636	0.7505	0.0153
$\widehat{\mathbf{K}}_{14}$	$\text{round}_{\text{AFZ}}$	$(0.6, 0.7]$	2.5308	89.7579	0.2299	0.0065
$\widehat{\mathbf{K}}_{15}$	$\text{round}_{\text{AFZ}}$	$(0.7, 0.8]$	3.8534	84.1782	0.6043	0.0087
$\widehat{\mathbf{K}}_{16}$	$\text{round}_{\text{AFZ}}$	$(0.7, 0.8]$	3.8484	87.7103	0.2418	0.0043
$\widehat{\mathbf{K}}_{17}$	$\text{round}_{\text{AFZ}}$	$(0.7, 0.8]$	3.8146	86.6308	0.1884	0.0049
$\widehat{\mathbf{K}}_{18}$	$\text{round}_{\text{AFZ}}$	$(0.8, 1)$	6.2462	88.1734	0.6746	0.0102
$\widehat{\mathbf{K}}_{19}$	$\text{round}_{\text{AFZ}}$	$(0.8, 1)$	6.1727	85.8301	0.1948	0.0055
$\widehat{\mathbf{K}}_{20}$	$\text{round}_{\text{AFZ}}$	$(0.8, 1)$	6.2335	86.827	0.4439	0.005

Source: Author (2023).

presented transforms for the values of ρ ranging from $(0, 0.7]$ and the second group transforms for values of $\rho \in (0.7, 1)$. In each group, we considered the transforms which presented the best values of the discussed figures of merit. The optimal transforms chosen are presented in the next section.

4.4 PROPOSED APPROXIMATE KLT AND EVALUATION

For the group C_1, which represents the transforms obtained for values of $\rho \in (0, 0.7]$, the optimal transforms are $\widehat{\mathbf{K}}_1$, $\widehat{\mathbf{K}}_3$, and $\widehat{\mathbf{K}}_{13}$. For C_2, the transforms $\widehat{\mathbf{K}}_{16}$, $\widehat{\mathbf{K}}_{17}$, and $\widehat{\mathbf{K}}_{18}$ were the ones which performed better considering the values of $\rho \in (0.7, 1)$. Table 9 presents the low-complexity matrices that generate the respective transforms.

The proposed transforms are not necessarily orthogonal, but we can quantify the deviation from orthogonality, which measures how close a matrix is from its diagonal. This measure is

Table 9 – Low-complexity matrices of the KLT Approximations

Transform	Matrix
$\mathbf{T}_1$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & 0 & -1 & 1 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$
$\mathbf{T}_3$	$\begin{bmatrix} 1 & 2 & 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 3 & 3 & 1 & -1 & -3 & -3 & -2 \\ 3 & 3 & 0 & -3 & -3 & 0 & 3 & 3 \\ 3 & 1 & -3 & -2 & 2 & 3 & -1 & -3 \\ 3 & -1 & -3 & 2 & 2 & -3 & -1 & 3 \\ 3 & -3 & 0 & 3 & -3 & 0 & 3 & -3 \\ 2 & -3 & 3 & -1 & -1 & 3 & -3 & 2 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}$
$\mathbf{T}_{13}$	$\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 & -1 & -2 & -2 \\ 2 & 1 & 0 & -2 & -2 & 0 & 1 & 2 \\ 2 & 0 & -2 & -1 & 1 & 2 & 0 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 2 & -2 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix}$
$\mathbf{T}_{16}$	$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 2 & -1 & -3 & -3 & -1 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 1 & 2 & -2 & -1 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}$
$\mathbf{T}_{17}$	$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 2 & -1 & -3 & -3 & -1 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 1 & 3 & -3 & -1 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}$
$\mathbf{T}_{18}$	$\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 & -1 & -2 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 0 & -2 & -1 & 1 & 2 & 0 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 2 & -2 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix}$

Source: Author (2023).

given by [170]

$$\delta(\mathbf{A}) = 1 - \frac{||\operatorname{diag}(\mathbf{A})||_{\mathrm{F}}}{||\mathbf{A}||_{\mathrm{F}}}, \tag{4.31}$$

where $\mathbf{A}$ is a square matrix. Table 10 presents each matrix, its diagonal matrix $\mathbf{S}$, and the deviation from diagonality measurement. In this case, only $\mathbf{T}_1$ is orthogonal. The remaining matrices have low diagonal deviation. As a reference value we used the deviation from diagonality from the SDCT [71], which is

$$\delta_{\mathrm{SDCT}} = 1 - \frac{2}{\sqrt{5}} \approx 0.1056. \tag{4.32}$$

All derived matrices have deviation smaller than δ_{SDCT}, thus regarded as quasi-orthogonal.

Table 10 – Proposed transforms and their respective diagonals

Matrix	Orthogonal?	$\mathbf{S}$	$\delta(\mathbf{K}\cdot\mathbf{K}^{\top})$
$\mathbf{T}_1$	Yes	$\operatorname{diag}\left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right)$	-
$\mathbf{T}_3$	No	$\operatorname{diag}\left(\frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{54}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{54}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}\right)$	0.0056
$\mathbf{T}_{13}$	No	$\operatorname{diag}\left(\frac{1}{\sqrt{14}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{18}}\right)$	0.0395
$\mathbf{T}_{16}$	No	$\operatorname{diag}\left(\frac{1}{\sqrt{32}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{44}}, \frac{1}{\sqrt{32}}, \frac{1}{\sqrt{36}}, \frac{1}{\sqrt{40}}, \frac{1}{\sqrt{46}}\right)$	0.0133
$\mathbf{T}_{17}$	No	$\operatorname{diag}\left(\frac{1}{\sqrt{32}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{44}}, \frac{1}{\sqrt{32}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{40}}, \frac{1}{\sqrt{46}}\right)$	0.0094
$\mathbf{T}_{18}$	No	$\operatorname{diag}\left(\frac{1}{\sqrt{14}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{18}}\right)$	0.0360

Source: Author (2023).

Table 11 presents the coding and similarities measurements of the exact KLT for a given value of ρ and the proposed approximate transforms. 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. In this case, we only compare the proposed transforms to the exact KLT for specific values of ρ. We divided Table 11 into the two groups, C_1 and C_2, and compared the approximate transforms from group C_1 to the exact KLT for $\rho = 0.2$ ($\mathbf{K}_8^{(0.2)}$), and the approximate transforms from group C_2 to the exact KLT for $\rho = 0.8$ ($\mathbf{K}_8^{(0.8)}$).

4.5 FAST ALGORITHMS AND COMPUTATIONAL COMPLEXITY

In this section, we introduce fast algorithms for the proposed approximations.

Table 11 – Comparison of coding and similarity measures between the exact KLT and the proposed approximate transforms

Transform	$C_g(\widehat{\mathbf{K}})$	$\eta(\widehat{\mathbf{K}})$	$\varepsilon(\mathbf{K}^{(\rho)},\widehat{\mathbf{K}})$	$\text{MSE}(\mathbf{K}^{(\rho)},\widehat{\mathbf{K}})$
$\mathbf{K}_8^{(0.2)}$	0.1551	100	0	0
$\widehat{\mathbf{K}}_1$	0.0308	93.4298	1.5331	0.0608
$\widehat{\mathbf{K}}_3$	0.0588	88.3104	0.093	0.0036
$\widehat{\mathbf{K}}_{13}$	2.5736	84.7636	0.7505	0.0153
$\mathbf{K}_8^{(0.8)}$	3.8824	100	0	0
$\widehat{\mathbf{K}}_{16}$	3.8484	87.7103	0.2418	0.0043
$\widehat{\mathbf{K}}_{17}$	3.8146	86.6308	0.1884	0.0049
$\widehat{\mathbf{K}}_{18}$	6.2462	88.1734	0.6746	0.0102

Source: Author (2023).

4.5.1 Proposed Fast Algorithms

By factoring the matrices of the proposed optimal transforms, $\mathbf{T}_1$, $\mathbf{T}_3$, $\mathbf{T}_{13}$, $\mathbf{T}_{16}$, $\mathbf{T}_{17}$, and $\mathbf{T}_{18}$ into sparse matrices, considering butterfly-based structures [23], we obtain the following decomposition:

$$\mathbf{T}_i = \mathbf{P}\cdot\mathbf{M}\cdot\mathbf{A}_1, \qquad i = 1,3,13, \tag{4.33}$$

$$\mathbf{T}_j = \mathbf{P}\cdot\mathbf{M}\cdot\mathbf{A}_2'\cdot\mathbf{A}_1, \qquad j = 16,17, \tag{4.34}$$

$$\mathbf{T}_{18} = \mathbf{P}\cdot\mathbf{M}\cdot\mathbf{A}_2''\cdot\mathbf{A}_1, \tag{4.35}$$

where $\mathbf{P}$ is a permutation matrix, $\mathbf{A}_1$, $\mathbf{A}_2'$, and $\mathbf{A}_2''$ are additive matrices, and $\mathbf{M}$ is a multiplicative matrix. For the factorization of $\mathbf{T}_1$, $\mathbf{T}_3$, and $\mathbf{T}_{13}$, we have

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.36}$$

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}. \tag{4.37}$$

The multiplicative matrix $\mathbf{M}$ can be written as

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \\ & \mathbf{M}_2 \end{bmatrix}, \tag{4.38}$$

where

$$\mathbf{M}_1 = \mathbf{M}_2 = \begin{bmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix}, \tag{4.39}$$

and the constants m_k, $k = 0, 1, \ldots, 15$, depend on the choice of the matrix $\mathbf{T}$ and are presented in Table 12. For the factorization of $\mathbf{T}_{16}$ and $\mathbf{T}_{17}$ we have

$$\mathbf{A}_2' = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.40}$$

and, for th $\mathbf{T}_{18}$ factorization,

$$\mathbf{A}_2'' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.41}$$

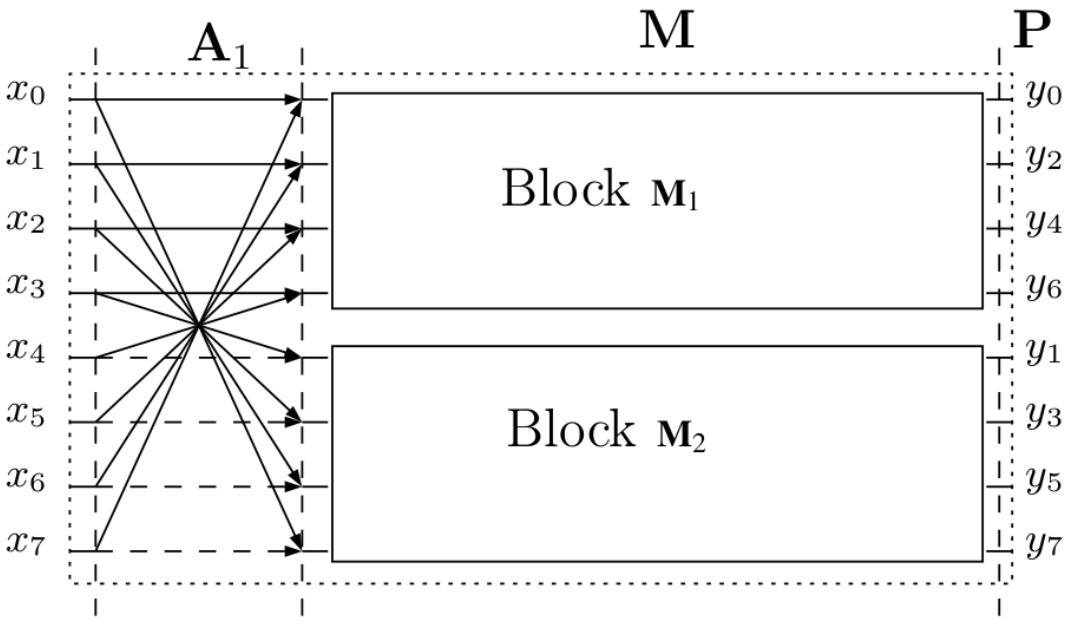
Table 12 – Constants required for the fast algorithm for blocks $\mathbf{M}_1$ and $\mathbf{M}_2$

Constants	$\mathbf{T}_1$		$\mathbf{T}_3$		$\mathbf{T}_{13}$		$\mathbf{T}_{16}$		$\mathbf{T}_{17}$		$\mathbf{T}_{18}$	
	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_1$	$\mathbf{M}_2$	$\mathbf{M}_1$	$\mathbf{M}_2$
m_0	0	0	1	1	1	0	2	1	2	1	1	0
m_1	1	1	2	3	1	1	2	2	2	2	1	1
m_2	1	1	3	3	1	2	2	3	2	3	0	2
m_3	1	1	3	2	2	2	0	3	0	3	2	2
m_4	1	−1	3	−2	2	−1	0	−2	0	−2	2	−1
m_5	1	−1	3	−3	1	−2	2	−3	2	−3	0	−2
m_6	0	0	0	1	0	0	−1	0	−1	0	1	0
m_7	−1	1	−3	3	−2	2	3	3	3	3	−2	2
m_8	1	1	3	3	1	2	2	2	2	3	1	2
m_9	0	0	−1	0	−1	0	−2	1	−2	1	−1	0
m_{10}	−1	−1	−3	−3	−1	−2	−2	−3	−2	−3	0	−2
m_{11}	1	1	2	3	1	1	0	2	0	2	1	1
m_{12}	1	−1	2	−3	1	−2	0	−3	0	−3	1	−2
m_{13}	−1	1	−3	3	−2	2	−3	3	−3	3	0	2
m_{14}	1	−1	3	−2	2	−1	3	−2	3	−2	−2	−1
m_{15}	0	0	−1	1	−1	0	1	1	1	1	−1	0

Source: Author (2023).

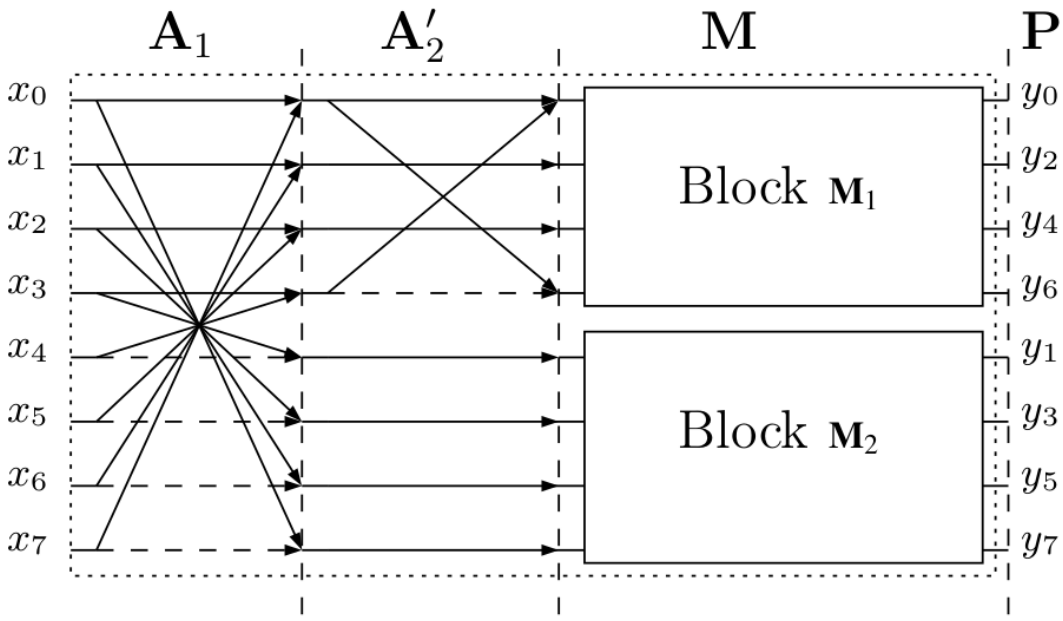
Figs. 24, 25, and 26 present the signal flow graphs of the fast algorithms. Diagrams relate the input data x_n, $n = 0, 1, \ldots, 7$, to the output data y_k, $k = 0, 1, \ldots, 7$, resulting in $\mathbf{y} = \mathbf{T} \cdot \mathbf{x}$. Here, dashed arrows represent multiplications by -1. When two or more arrows meet, their values are added [23]. Blocks $\mathbf{M}_1$ and $\mathbf{M}_2$ share the same structure in all diagrams except for the value of the constants presented in Table 12 and are displayed in Fig. 27.

Figure 24 – Signal flow graph for $\mathbf{T}_1$, $\mathbf{T}_3$, and $\mathbf{T}_{13}$.



Source: Author (2023).

Figure 25 – Signal flow graph for $\mathbf{T}_{16}$ and $\mathbf{T}_{17}$.

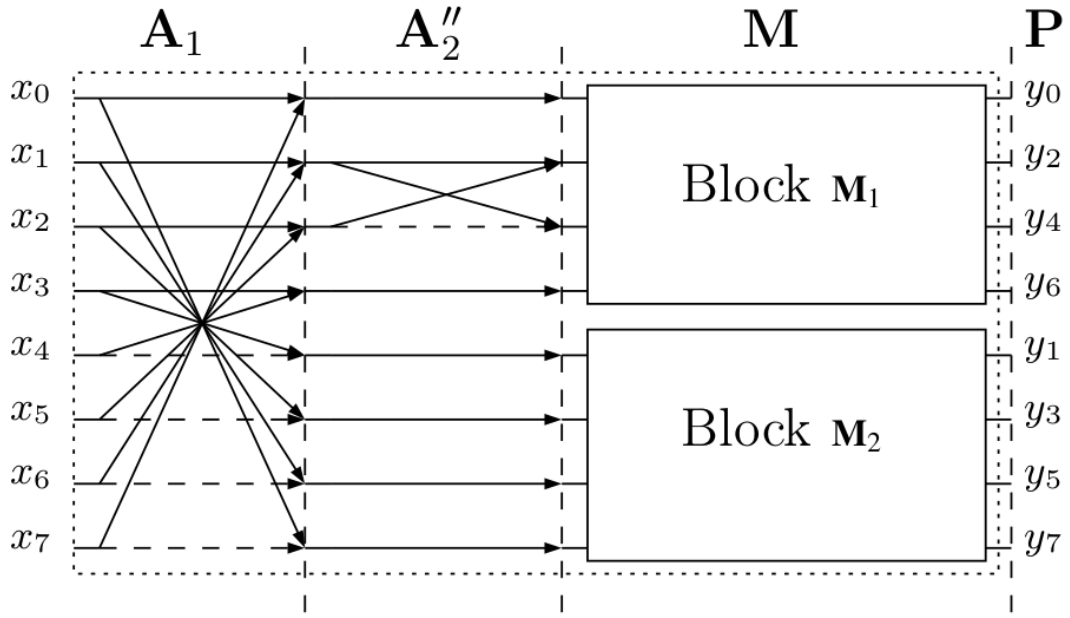


Source: Author (2023).

4.5.2 Computational Complexity

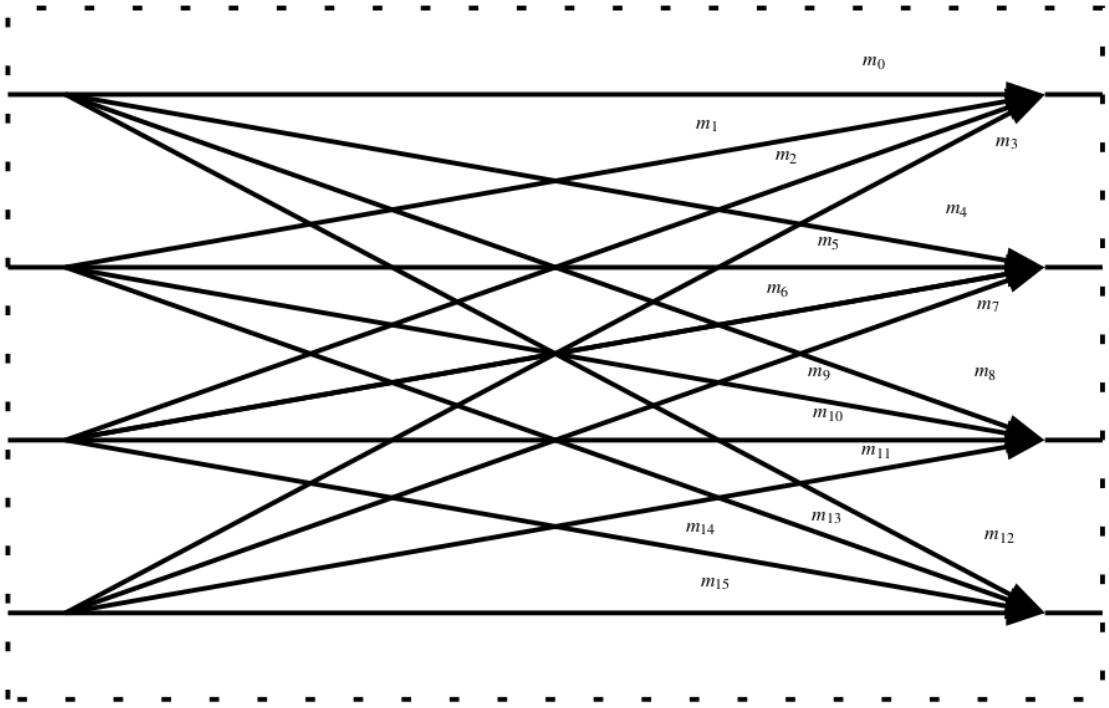
The computational complexity of the proposed transforms can be estimated by the arithmetic complexity, given by the number of multiplications, additions and bit-shifting operations required for its implementation. Table 13 presents the arithmetic complexity of the discussed fast algorithms. We also present the cost of direct computation of the 8-point KLT and the 8-point DCT considering the fast algorithms proposed by Loeffler [45]. The proposed transforms have a

Figure 26 – Signal flow graph for $\mathbf{T}_{18}$.



Source: Author (2023).

Figure 27 – Blocks $\mathbf{M}_1$ and $\mathbf{M}_2$ from the signal flow graphs.



Source: Author (2023).

much lower cost when compared with the KLT and DCT. From the proposed transforms, we can highlight $\mathbf{T}_1$, $\mathbf{T}_{13}$, and $\mathbf{T}_{18}$ as the ones that have the lower arithmetic cost. These transforms perform well when compared with the fast algorithms for classical 8-point low-complexity approximations such as the SDCT [71], which requires only 24 addition operations for its implementation.

Table 13 – Comparison of the arithmetic complexity of the 8-point transforms

Transform	Additions	Multiplications	Bit-shifts
KLT	56	64	0
DCT [45]	29	11	0
$\mathbf{T}_1$	24	0	0
$\mathbf{T}_3$	48	0	24
$\mathbf{T}_{13}$	26	0	13
$\mathbf{T}_{16}$	38	0	22
$\mathbf{T}_{17}$	38	0	22
$\mathbf{T}_{18}$	26	0	12

Source: Author (2023).

4.6 EXPERIMENTS ON IMAGE COMPRESSION

4.6.1 JPEG-like Compression

The compression performance of the proposed transforms can be evaluated when applied to image coding experiments, as well as in [72, 74, 81]. For simplicity, but without loss of generality, 8-bit images in gray scale were considered. The JPEG-like compression methodology used in this experiment is presented as follows [16]. The input image was divided into disjoint 8×8 sub-blocks. Let $\mathbf{A}$ be a sub-block. The direct two-dimensional (2D) transform was applied to each sub-block, resulting in [53]

$$\mathbf{B} = \widehat{\mathbf{K}} \cdot \mathbf{A} \cdot \widehat{\mathbf{K}}^{\top}. \tag{4.42}$$

Considering the zig-zag pattern [16], the initial r coefficients from $\mathbf{B}$ were retained, resulting in truncated sub-blocks $\bar{\mathbf{B}}$. The 2D inverse transform was applied to each sub-block $\bar{\mathbf{B}}$, resulting in

$$\bar{\mathbf{A}} = \widehat{\mathbf{K}}^{-1} \cdot \bar{\mathbf{B}} \cdot (\widehat{\mathbf{K}}^{-1})^{\top}. \tag{4.43}$$

The compressed sub-blocks $\bar{\mathbf{A}}$ were recomposed in the place of the originals sub-blocks $\mathbf{A}$. Finally, the compressed image was compared to the original image to evaluate the loss of quality imposed by compression using transform $\widehat{\mathbf{K}}$.

For assessing the quality of compressed images, we used as figures of merit the peak signal-to-noise ratio (PSNR) [135] and the mean structural similarity index (MSSIM) [136]. Even though it is a very popular metric, it was shown in [157] that the PSNR is not the best measure when it comes to predict the human perception of image quality [136,157]. Nevertheless, we considered this figure of merit for comparison purposes. On the other hand, the MSSIM was

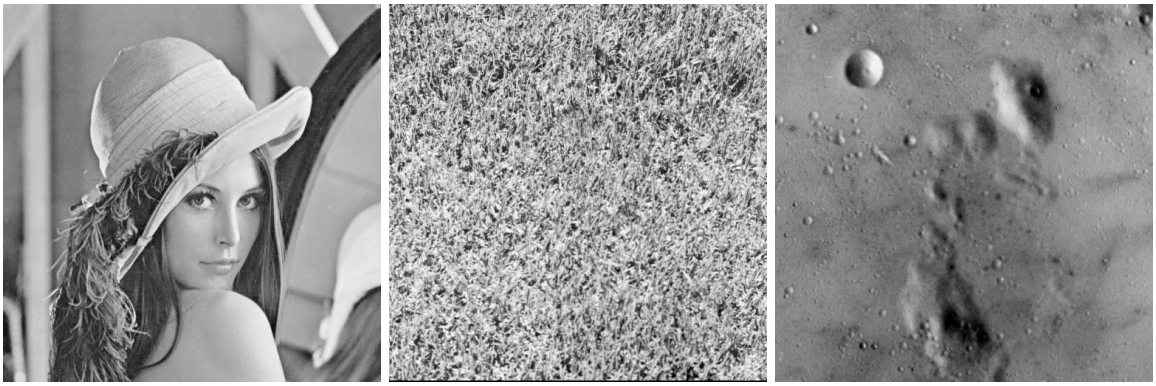
shown to be capable of closely capturing the image quality as understood by the human visual system model [136].

4.6.2 Results and Discussion

Fig. 28 presents the original *Lena*, *Grass*, and *Moon* images [137] used in the qualitative analysis. In this step, each image was submitted to a compression rate (CR) of 85%, $r = 10$. Fig. 29, 30, and 31 present the compressed images using the proposed transforms and the exact KLT for $\rho = 0.2$ and 0.8.

Figure 28 – Original Images.

(a) *Lena* **(b)** *Grass* **(c)** *Moon*

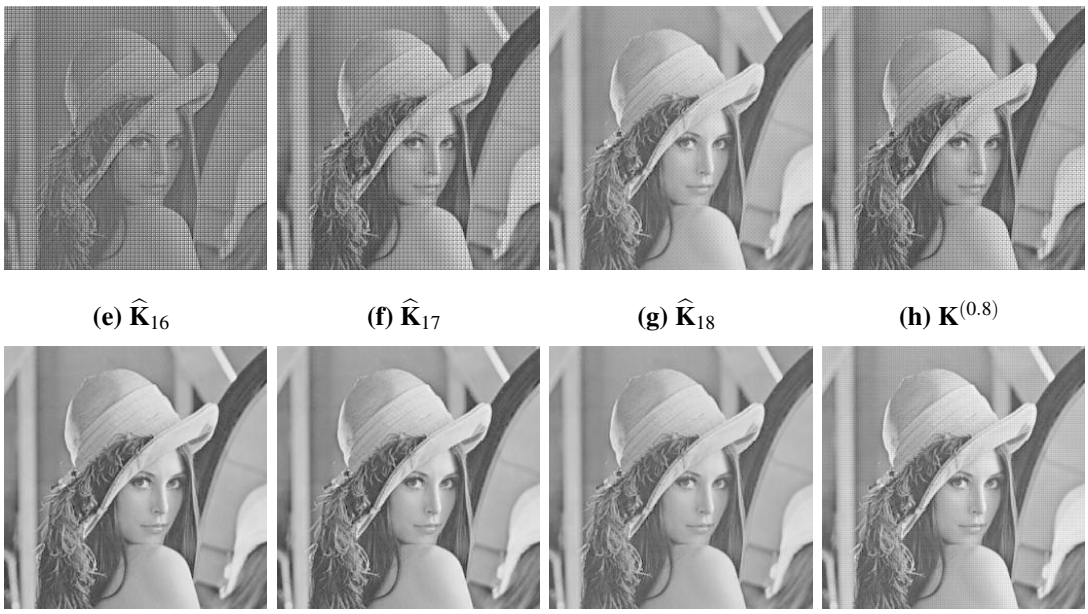


Source: SIPI [137].

Figure 29 – Compressed *Lena* Images.

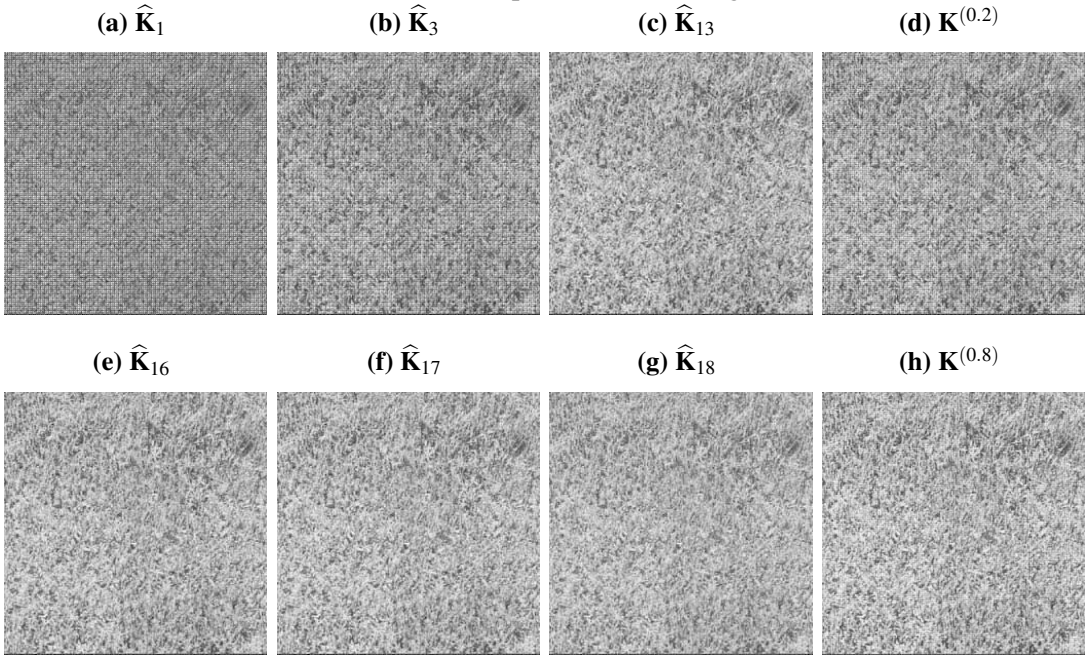
(a) $\widehat{\mathbf{K}}_1$ **(b)** $\widehat{\mathbf{K}}_3$ **(c)** $\widehat{\mathbf{K}}_{13}$ **(d)** $\mathbf{K}^{(0.2)}$

(e) $\widehat{\mathbf{K}}_{16}$ **(f)** $\widehat{\mathbf{K}}_{17}$ **(g)** $\widehat{\mathbf{K}}_{18}$ **(h)** $\mathbf{K}^{(0.8)}$



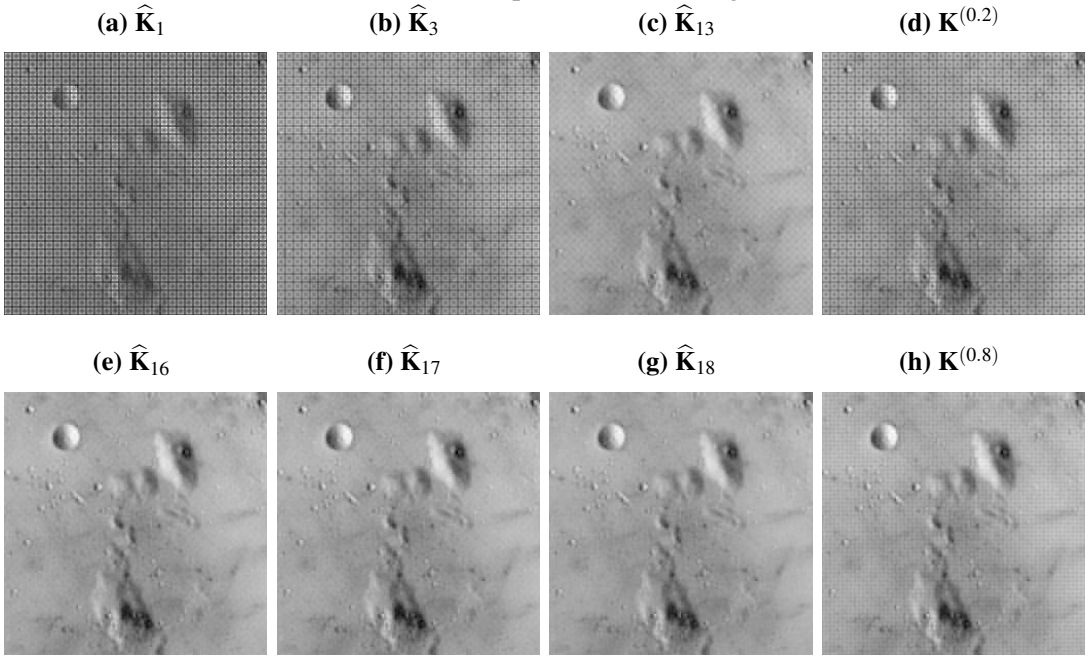
Source: Author (2023).

Figure 30 – Compressed *Grass* Images.



Source: Author (2023).

Figure 31 – Compressed *Moon* Images.



Source: Author (2023).

Table 14 presents the PSNR and MSSIM values for the compressed images. The proposed transforms perform well, and in some cases even better than the exact KLT, for a given value of ρ within the interval of each group of transforms. We highlighted the values of the best measurements for each group of the approximate transforms. The proposed transforms $\widehat{\mathbf{K}}_1$,

$\widehat{\mathbf{K}}_3$, and $\widehat{\mathbf{K}}_{13}$ were derived considering low values of ρ. As the pixels of a natural image are highly correlated [24], image compression using these transforms does not show the best results, as expected. However, considering the other proposed transforms, and even $\widehat{\mathbf{K}}_{13}$, we can see that, qualitatively, there is no visually perceptible differences between the compressed images considering the approximate transforms and the exact KLT.

Table 14 – Image quality measures

Image	*Lena*		*Grass*		*Moon*	
Transform	PSNR	MSSIM	PSNR	MSSIM	PSNR	MSSIM
$\widehat{\mathbf{K}}_1$	10.7230	0.1086	10.2617	0.3440	10.837	0.062
$\widehat{\mathbf{K}}_3$	18.2075	0.2698	16.1770	0.6216	18.238	0.186
$\widehat{\mathbf{K}}_{13}$	**30.5265**	**0.8093**	**19.6360**	**0.7797**	**29.374**	**0.687**
$\mathbf{K}^{(0.2)}$	20.14173	0.3302	17.3274	0.6759	20.112	0.240
$\widehat{\mathbf{K}}_{16}$	**31.8353**	0.8942	**19.9568**	**0.7884**	**30.388**	0.744
$\widehat{\mathbf{K}}_{17}$	31.7447	0.8934	19.9213	0.7874	30.355	0.743
$\widehat{\mathbf{K}}_{18}$	31.6908	**0.9091**	19.59472	0.7776	30.364	**0.748**
$\mathbf{K}^{(0.8)}$	29.9278	0.7584	19.8954	0.7861	28.896	0.653
$\mathbf{K}^{(0.95)}$	31.9935	0.9019	19.9384	0.7864	30.473	0.748
DCT	32.0814	0.9136	19.893	0.7839	30.566	0.754

Source: Author (2023).

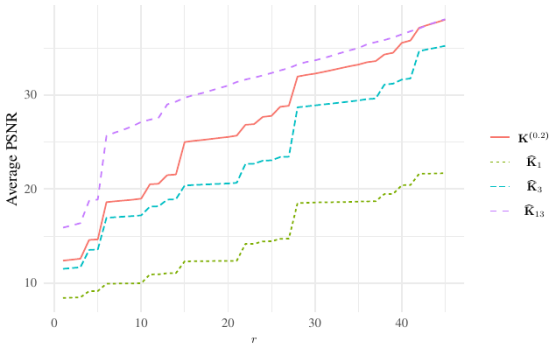
We extended the experiment to a group of 45 512×512 8-bit greyscale images, obtained from [137], considering different rates of compression ($1 \leq r \leq 45$). The PSNR and MSSIM measures were computed for each compressed image, and the average of these values were taken. Fig. 32 presents the plots of the average values of these measures. There are two graphs for each figure of merit, one for each group of the approximate transforms, C_1 and C_2. In order to compare the approximate transforms we also calculated this measurements for the exact KLT considering the values of $\rho = 0.2$ and 0.8. The proposed transforms demonstrate satisfactory performance when compared to the exact KLT and the transforms from group C_2 even outperformed the exact KLT for $0 < r < 15$ approximately.

4.7 HARDWARE IMPLEMENTATION

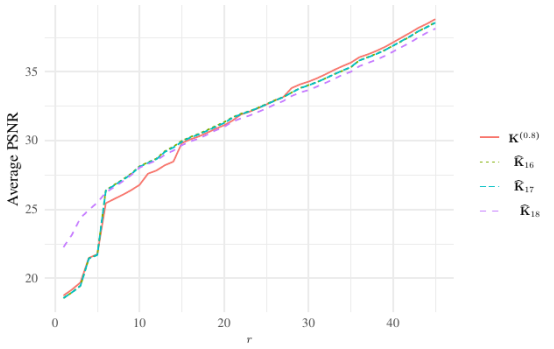
The 8-point low-complexity transforms outlined in Table 9 were implemented on an FPGA. The platform adopted for the hardware implementation was the Xilinx Artix-7 XC7A35T-1CPG236C. Notice we do not implement the diagonal elements in Table 10. This is because

Figure 32 – Quality measures of the considered approximations for several values of r according to the figures of merit.

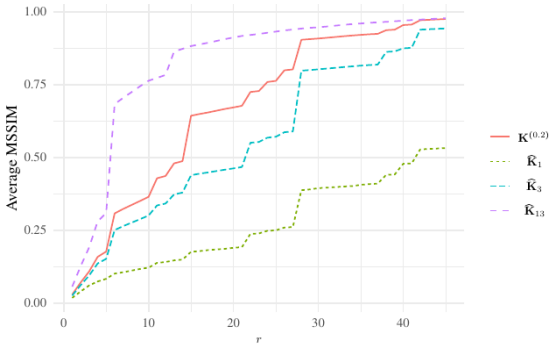
(a) Average PSNR considering C_1 group approximate transforms



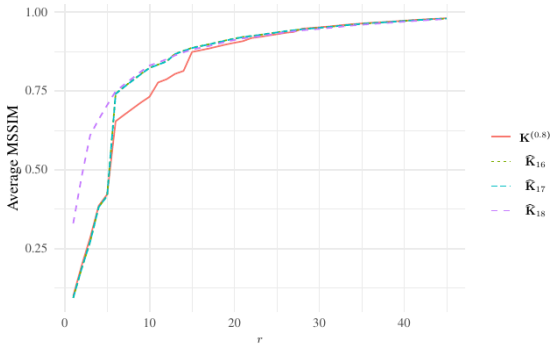
(b) Average PSNR considering C_2 group approximate transforms



(c) Average MSSIM considering C_1 group approximate transforms



(d) Average MSSIM considering C_2 group approximate transforms



Source: Author (2023).

they can be easily incorporated into the quantization step in image and video compression schemes [26].

The designs were implemented using a pipelined systolic architecture for each of the transforms [171, 172]. Each transform implementation is split in different sub-blocks. Each sub-block implements a different matrix in its corresponding fast algorithm as in (4.33), (4.34), and (4.35) and is displayed in Figs. 24, Fig. 25, and Fig. 26, respectively. Each sub-block that requires an arithmetic operation expands the wordlength in one bit in order to avoid overflow. The sub-block implementing the permutation matrix $\mathbf{P}$ in (4.36) contains only combinational logic as it only requires re-routing of the transform coefficients and does not possess any arithmetic operation. The kernel $\mathbf{M}$ of all the transforms in (4.38) are implemented with two clock cycles of latency. This is because each row of each of the transform kernel possesses at least three nonzero entries (cf. Table 12). The intermediary matrices $\mathbf{A}_1$, $\mathbf{A}'_2$, and $\mathbf{A}''_2$ in the factorizations in (4.33), (4.34), and (4.34), respectively, require at most an addition of two elements per row, so that only one clock cycle is enough to implement each one.

The designs were implemented and tested according to the scheme shown in Fig. 33, along with a state-machine serving as controller and connected to a Universal Asynchronous Receiver-Transmitter (UART) block. The UART core interfaces with the controller through an ARM Advanced Microcontroller Bus Architecture Advanced eXtensible Interface 4 (AMBA AXI-4) protocol.

Figure 33 – Testbed architecture for testing the implemented designs.

Source: Author (2023).

The Personal Computer (PC) communicates with the controller through the UART by sending a set of eight 8-bit coefficients, which corresponds to an input for the transform block under test. The values of the 8-bit coefficients are drawn from a uniform distribution in the interval $[-10, 10]$. The set of the eight coefficients are then sent to the design and processed. After processed, the controller sends the eight output coefficients back to the PC, which is compared with the output of a software model used to ensure the hardware design is accurately implemented.

Table 15 shows the hardware resources utilization and metrics for the transforms in Table 9. The considered figures of merit are the number of occupied slices, number of Look-Up Tables (LUT), Flip-Flop (FF) count, wordlength increase (Δ #bits), latency (L) in terms of clock cycles, critical path delay (T_{cpd}), maximum operating frequency $F_{\text{max}} = T_{\text{cpd}}^{-1}$, and dynamic power ($D_p$) normalized by F_{max}.

Table 15 – FPGA measures of the implemented architectures new and competing transforms

Transform	Metrics							
	Slices	LUT	FF	Δ #bits	L (cycles)	T_{cpd} (ns)	F_{max} (MHz)	D_p (μW/MHz)
$\mathbf{T}_1$	**75**	**217**	**279**	**3**	**3**	**3.691**	**270.929**	**33.219**
$\mathbf{T}_3$	150	471	370	5	**3**	4.961	201.572	54.571
$\mathbf{T}_{13}$	93	277	334	4	**3**	4.203	237.925	37.827
$\mathbf{T}_{16}$	143	406	444	6	4	4.926	203.004	54.186
$\mathbf{T}_{17}$	148	426	444	6	4	5.072	197.161	55.792
$\mathbf{T}_{18}$	110	287	401	5	4	4.580	218.341	41.220

Source: Author (2023).

Among the considered transforms, $\mathbf{T}_1$ is the one requiring the least amount of resources such as FFs, LUT, and, consequently, slices. This is due to two factors: (i) the latency L, and

(ii) wordlength increase Δ #bits. Smaller latency means less registers are needed for storing information, directly reducing the need for FFs and LUTs. Also, with reduced wordlength increase Δ #bits, less routing resources are needed inside the device to attain the desired computation.

The latency is a direct consequence of the fast algorithm that is found for the considered transforms, as outlined in (4.33), (4.34), and (4.35). The transform $\mathbf{T}_1$ is factorized with the simplest of the fast algorithms, involving only three matrices — in fact only two that requires arithmetic operations — as compared to the other algorithms that requires one more matrix — in fact only three that requires arithmetic operations. An inspection of Table 12 along with (4.38) also shows that the transform kernel $\mathbf{M}$ for $\mathbf{T}_1$ requires only additions of at most three elements per row. It does not require any constant multiplication by two (a bit-shifting operation) or by three (a bit-shifting plus addition) like the other transforms in Table 9 (cf. Table 12), which then renders a transform requiring a smaller bit increment when compared to other proposed transforms. Because of the reduced amount of resources when compared to the other transforms in Table 15, $\mathbf{T}_1$ is also the transform with the least critical path delay, and therefore the highest maximum operating frequency and normalized dynamic power.

The transform $\mathbf{T}_{13}$ is the second most economical in terms of hardware resources considering FFs, LUTs, and slices. It shares in common with $\mathbf{T}_1$ its fast algorithm, however, the $\mathbf{T}_{13}$ kernel requires multiplications by constants as shown in Table 12, demanding a higher wordlength increment and therefore more resources than $\mathbf{T}_1$. The transform $\mathbf{T}_{17}$ is the one that requires the most amount of resources and possessing the highest critical path delay, resulting in the lowest maximum operating frequency and normalized dynamic power in comparison to the other transforms in Table 15.

4.8 CONCLUSIONS

In this chapter, we proposed a new class of data-independent low-complexity KLT approximations. The obtained approximations were derived applying a set of rounding functions to the elements of the exact KLT, varying the value of the correlation coefficient ρ. An optimization problem was solved aiming at the proposition of optimal transforms according to defined figures of merit. The k-means clustering method was used to classify the optimal transforms into groups to certain ρ values intervals. Fast algorithms were derived for the optimal approximations proposed by factorizing the transforms into sparse matrices. Only addition and bit-shifting

operations were necessary for the implementation of the proposed transforms. **To the best of our knowledge, there is no low-complexity approximation transforms in the literature that covers the whole correlation scenario** ($0 < \rho < 1$) **as proposed in this thesis.** Thus, there is no competing method to do a fair comparison with the proposed transforms. The applicability of the proposed approximation in the context of image compression was demonstrated. Our experiments showed that the proposed transforms performed very well when compared to the exact KLT and, in the cases of $\widehat{\mathbf{K}}_{16}$, $\widehat{\mathbf{K}}_{17}$, and $\widehat{\mathbf{K}}_{18}$, even outperformed the exact KLT and DCT. We also provided FPGA hardware implementation, showing a trade-off between performance and resource usage, where the $\mathbf{T}_1$ requires the least amount of resources and the transform $\mathbf{T}_{17}$ higher amounts of FFs and LUTs.

5 EXTENSIONS ON LOW-COMPLEXITY DCT APPROXIMATIONS FOR LARGER BLOCKLENGTHS BASED ON MINIMAL ANGLE SIMILARITY

5.1 INTRODUCTION

Current technological trends suggest an ever-increasing demand for efficient, low-power, low-complexity digital signal processing methods [173–176]. In this context, many important discrete transforms have become useful tools for signal coding and data decorrelation [12–14, 16, 177] such as the discrete Fourier transform (DFT), the discrete Hartley transform (DHT), the Walsh-Hadamard transform (WHT), the discrete Tchebichef transform (DTT), the discrete cosine transform (DCT), among others. Data compression techniques [110, 152, 178, 179] address the problem of removing redundancy from data [16, pg. 2]. Such removal can be accomplished by using the Karhunen-Loève transform (KLT) [13, 14], which is the optimal tool in terms of energy compaction. Indeed, the KLT packs the energy of the input signal in few transform domain coefficients and diagonalizes the covariance or correlation matrix of the data, resulting in a completely decorrelated signal [14, pg. 10]. In practice, the KLT is not widely adopted because it is a data-dependent transformation, which severely precludes the development of fast algorithms. Fast algorithms can significantly reduce the arithmetic cost of the transform by searching for a computationally efficient implementation method [23, pg. 2].

However, if the input signal is a first-order Markovian process, then the KLT matrix depends only on the correlation coefficient ρ of the process. Natural images constitute a representative class of such kind of data, which often present high values of ρ. When $\rho \to 1$, the KLT becomes the type-II DCT which is independent of the input data or any other parameter [13, pg. 56]. Although natural images do not necessarily present $\rho = 1$, their correlation coefficient is sufficiently high [12] to make the DCT extremely efficient and popular in image and video coding, such as JPEG [26], MPEG [27], and HEVC [28]. Since its introduction in [115], the DCT has been shown to be a superior tool when compared to the other transforms known in the literature in this context of image and video coding. This fact was confirmed in a number of independent works such as [12–14, 24, 26, 116, 153, 164, 178, 180]. The fact that the DCT is data-independent allows the development of low-complexity fast algorithms for its calculation [23].

Nevertheless, under scenarios of severe restrictions on processing power or energy autonomy [81, 118], the arithmetic cost of computing the DCT by traditional algorithms might still be a hindrance. Thus, several multiplierless low-complexity DCT approximations have been developed as detailed in [46, 71, 72, 74, 78, 80, 81, 83, 181, 182]. Particularly, we separate

the method presented in [80], where a DCT approximation is derived by minimizing the angle between each corresponding rows of the exact and approximate transforms. The resulting 8-point DCT approximation introduced in [80] outperforms well-known DCT approximations in the literature according to classical figures of merit as the mean squared error, total energy error, coding gain, and transform efficiency. Furthermore, the associated fast algorithm requires only 24 additions and 6 bit-shifting operations.

The search for 8-point DCT approximations is a relatively mature area of research since the 8-point DCT is a key block in many image and video processing applications. However, with the advancement of image and video encoding technology, there is a demand for larger blocklengths that could be employed in modern codecs [183]. For instance, there is a new video standard coding, the Versatile Video Coding (VVC) [184], that requires 64-point transforms. In this chapter, we extended the method proposed in [80] for larger blocklengths, i.e., for $N = 16$, 32, and 64. To the best of our knowledge, the current literature archives only a few works [78,85,131] addressing low-complexity approximations for the DCT of the above-mentioned sizes. We aim at proposing high-performing low-complexity DCT approximations for such blocklengths.

This chapter is structured as follows. In Section 5.2, we present a brief review of the relevance of the DCT and approximations in the context of hardware implementation and its applications. In Section 5.3, we review the formulation of the DCT and low-complexity approximations. Section 5.4 presents the methodology for deriving the proposed transforms. In Section 5.5, the introduced DCT approximations are presented along with performance assessment and fast algorithms derivation. Section 5.6 presents computational experiments in image compression that demonstrate the suitability of the suggested tools. Section 5.7 concludes the chapter.

5.2 HARDWARE REVIEW

The energy packing and high decorrelation properties for signals modeled after highly-correlated first-order Markovian process makes the DCT a widely employed method for image and video coding [164]. In this context, the DCT is the main tool [80, 86, 91, 122, 148, 185–188], finding implementations in JPEG [87], motion JPEG [189], MPEG [190], and HEVC [191–193]. Such efficient encoders can provide a significant reduction in computational complexity, enabling the development of high-speed computing architectures [194]. Popular architectures that benefited from transform-based encoding include HD and UHD videos [195, 196], smart

antenna applications [197, 198], secure image processing [199, 200], image fusion [201] and defusion [202], biomedical signal processing [203, 204], to cite but a few.

Therefore, the design of DCT-based Very Large-Scale Integration (VLSI) structures has been an essential task for decreasing chip area, power, and time consumption [205–208]. Thus, resource-constrained platforms which require hardware designs capable of larger autonomy, increased storage capacity, extended battery life, and data transmission are prime beneficiaries of low-complexity methods. This is illustrated in the case of low-powered devices in real-time applications [209] and nanotechnologies [210], which require ultralow power consumption. In addition, sensor architectures that present stringent limitations on memory and processing speed [88, 211, 212] and sub-branches like approximate memory, which focus on the trade off between perfect data fidelity and storage density [213], received technical contributions from DCT approximations. The requirements for low-cost computation can be easily noticed in the context of Internet of Things (IoT) [214–217] or 5G technologies [218, 219], for example.

The assumption that approximate methods lead to hardware implementations of low resource consumption was corroborated in a detailed comprehensive study [87].

Motivated by this increasing demand, we focus our work on further reducing the arithmetic complexity of the DCT computation, which directly affects hardware-oriented measures such as chip area, dynamic power consumption, critical path delay, gate-count, area-time, and maximum clock frequency (throughput) [33, 101, 186, 220–223].

5.3 EXACT AND APPROXIMATE DCT

The N-point DCT is represented by an $N \times N$ matrix $\mathbf{C}_N$ the elements of which are given by [13, pg. 61]

$$c_{i,j} = \sqrt{\frac{2}{N}} u_i \cos\left\{\frac{i(2j+1)\pi}{2N}\right\}, \quad i,j = 0,1,\ldots,N-1, \tag{5.1}$$

where the quantities u_i are defined as

$$u_i = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } i = 0, \\ 1, & \text{if } i \neq 0. \end{cases} \tag{5.2}$$

Let

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \ldots & x_{N-1} \end{bmatrix}^\top \tag{5.3}$$

be an N-point input vector. The DCT transformation of $\mathbf{x}$ is the output vector

$$\mathbf{X} = \begin{bmatrix} X_0 & X_1 & \dots & X_{N-1} \end{bmatrix}^\top, \tag{5.4}$$

given by

$$\mathbf{X} = \mathbf{C}_N \cdot \mathbf{x}. \tag{5.5}$$

Since the DCT matrix is orthogonal, its inverse transformation can be written as [13, pg. 41]

$$\mathbf{x} = \mathbf{C}_N^\top \cdot \mathbf{X}. \tag{5.6}$$

Generally, DCT approximations are transformations $\widehat{\mathbf{C}}_N$ that behave similarly to the exact DCT according to a relevant criterion depending on the context at hand. An approximate transform is usually based on a low-complexity transform $\mathbf{T}_N$, i.e., a transformation matrix whose entries possess lower multiplicative complexity [13, 23]. Typical examples of low-complexity multipliers are: $\{0, \pm 1, \pm 2, \pm 4, \dots\}$, $\{0, \pm\frac{1}{2}, \pm 1, \pm 2, \pm 3\}$, and $\{0, \pm\frac{1}{4}, \pm\frac{1}{2}, \pm 1\}$; in all these cases, the multiplication of a given number by any set element requires no or a few additions and bit-shifting operations. Once a low-complexity matrix is obtained, we can derive the associate approximate transform according to the next formalism [81, 156]:

$$\widehat{\mathbf{C}}_N = \mathbf{S}_N \cdot \mathbf{T}_N, \tag{5.7}$$

where

$$\mathbf{S}_N = \sqrt{[\mathrm{diag}(\mathbf{T}_N \cdot \mathbf{T}_N^\top)]^{-1}}, \tag{5.8}$$

being $\mathrm{diag}(\cdot)$ the diagonal matrix generated by its arguments and $\sqrt{\cdot}$ the matrix square root operator [156]. If $\mathbf{T}_N$ is almost orthogonal [81], then the matrix $\widehat{\mathbf{C}}_N$ can represent a meaningful approximation for $\mathbf{C}_N$. The concept of almost orthogonality stems from almost diagonality property as defined in [81, 170].

Notice that if $\mathbf{T}_N$ presents the property of diagonality, i.e.,

$$\mathbf{T}_N \cdot \mathbf{T}_N^\top = [\text{diagonal matrix}], \tag{5.9}$$

than (5.7) provides an orthogonal approximation $\widehat{\mathbf{C}}_N$. In addition, if $\mathbf{T}_N$ is orthogonal then $\mathbf{S}_N$ results in the identity matrix $\mathbf{I}_N$ and $\widehat{\mathbf{C}}_N = \mathbf{T}_N$.

However, finding a good low-complexity matrix $\mathbf{T}_N$ can be a hard task, because of the large search space. For instance, the matrix space of 16×16 low-complexity matrices defined

over the set $\{-1,0,+1\}$ possesses $3^{16^2} \approx 1.39 \times 10^{122}$ candidate matrices. An exhaustive search over this space would take approximately 4.41×10^{105} years of computation assuming that each matrix could be generated and assessed in 1 nanosecond.

Therefore, a crucial step in deriving approximations is the reduction of the search space by restricting the search to potentially good matrices only. Literature describes several methods to accomplish such reduction: (i) matrix quantization [16]; (ii) matrix parametrization [76, 120]; (iii) algorithm parametrization [62, 181]; and (iv) visual inspection [74, 75, 77, 78, 83, 121, 224]. Depending on the particular search space reduction approach, the number of candidate approximations can be as small as just one matrix (e.g., SDCT [71] and RDCT [72]) or as large as classes of various matrices, as shown in [80–82, 181] for the 8-point case.

5.4 DCT APPROXIMATIONS WITH MINIMAL ANGULAR ERROR

In [80], a search space reduction based on the vector direction analysis [225, 226] of the transformation basis vectors was proposed. In the following, we summarize the method.

The exact DCT matrix $\mathbf{C}_N$ can be understood as a stack of row vectors

$$\mathbf{c}_k^\top, \quad k = 0, 1, \ldots, N-1. \tag{5.10}$$

The goal of the method is to find a low-complexity matrix $\mathbf{T}_N$, whose rows are denoted by

$$\mathbf{t}_k^\top, \quad k = 0, 1, \ldots, N-1, \tag{5.11}$$

such that a prescribed error measure between the corresponding rows of the exact DCT and the approximation is minimized. The entries of $\mathbf{t}_k^\top$ are selected from

$$\mathscr{D} = \{d_0, d_1, \ldots, d_{D-1}\}, \tag{5.12}$$

where d_i, $i = 0, 1, \ldots, D-1$, are low-complexity (trivial [23]) multipliers. Departing from the usual measures in the approximate transform literature, such as Euclidean distance, in [80], the angle between vectors is adopted as the error function. Thus, the following optimization problem is defined [80]:

$$\mathbf{t}_k = \underset{\mathbf{p} \in \mathscr{D}^N}{\arg\min} \operatorname{angle}(\mathbf{p}, \mathbf{c}_k), \quad k = 0, 1, \ldots, N-1, \tag{5.13}$$

where $\mathbf{p}^\top$ is a candidate row defined over the N-dimensional discrete space

$$\mathscr{D}^N = \mathscr{D} \times \mathscr{D} \times \cdots \times \mathscr{D}, \tag{5.14}$$

the angle between two vectors, $\mathbf{p}$ and $\mathbf{c}_k$, is given by

$$\text{angle}(\mathbf{p},\mathbf{c}_k) = \arccos\left(\frac{\langle \mathbf{p},\mathbf{c}_k\rangle}{||\mathbf{p}||\cdot||\mathbf{c}_k||}\right), \tag{5.15}$$

the symbol $\langle\cdot,\cdot\rangle$ denotes the usual inner product, and $||\cdot||$ is the norm induced by the inner product [227]. The resulting low-complexity matrix $\mathbf{T}_N$ from the above optimization problem is said to have minimal angular error relative to the exact DCT.

The search space implied by the above-described procedure contains D^N rows. Therefore, the computation described in (5.13) requires $N\cdot D^N$ angle evaluations at most. In [80], it was adopted $\mathscr{D} = \{-1,0,1\}$ ($D = 3$) and $N = 8$, implying in only $3^8 = 6561$ candidate rows. For larger values of N, the number of candidate rows are presented in Table 16, also considering $\mathscr{D} = \{-1,0,1\}$. Notice that any other discrete space with three elements will generate the same number of candidate rows. Hereafter the collection of all possible rows is called $\mathscr{P}$.

Table 16 – The relation between matrix size and the number of candidate rows, considering a discrete space of three elements (*e.g.* $\mathscr{D} = \{-1,0,1\}$).

N	Number of candidate rows
8	6561
16	43046721
32	$\approx 1.85\times 10^{15}$
64	$\approx 3.43\times 10^{30}$

Source: Author (2023).

The above procedure does not ensure the orthogonality of the resulting matrix. Although orthogonality is not strictly a necessary condition for the derivation of good DCT approximations (e.g., SDCT [71]), it is often a desirable design feature. To address such specific need, the optimization problem in (5.13) can be extended by the inclusion of an orthogonality constraint such that each new candidate row is compared against the previously obtained rows ensuring that their inner products are null [127]. However, the orthogonality constraint increases significantly the search time, since this restriction is sensitive to the order in which the rows are approximated, being feasible for small blocklengths only, as successfully demonstrated in [80] for $N = 8$. Algorithm 3 presents the pseudocode that contains the unconstrained procedure.

5.5 PROPOSED DCT APPROXIMATIONS

In this section, we report and assess the proposed approximations obtained from the approach based on the minimal angle error, presented in the Algorithm 3, for $N \in \{16,32,64\}$.

Algorithm 3: Pseudocode for the unconstrained angle based method.

Input: $\mathbf{C}_N, \mathscr{P}$
Output: $\mathbf{T}_N$
  for $k \leftarrow 0, 1, \ldots, N-1$ **do**
    $\theta_{\min} \leftarrow 2\pi$;
    for $i \leftarrow 0, 1, \ldots, D^N - 1$ **do**
      $\mathbf{p} \leftarrow \mathscr{P}_i$;
      $\theta \leftarrow \text{angle}(\mathbf{p}, \mathbf{c}_k)$;
      if $(\theta < \theta_{\min})$ **then**
        $\theta_{\min} \leftarrow \theta$;
        $\mathbf{t}_k \leftarrow \mathbf{p}$;
      end if
    end for
    $\mathbf{T}_N(k,:) \leftarrow \mathbf{t}_k$;
  end for
  return $\mathbf{T}_N$;

For this search, we have used a machine with the following specifications: hexa-core 4.5 GHz Intel(R) Core(R) I7-9750H, with 32 GB RAM running Ubuntu 20.04 LTS 64-bit and GPU GeForce RTX 2060. In parallel, we also used a virtual machine from Google Cloud Platform with the following specifications: 8 cores 3.8 GHz Intel(Cascade Lake) with 32 GB RAM running Ubuntu 20.04 LTS 64-bit. Besides the approximations obtained from this approach, we have also scaled the best transforms according to the Jridi-Alfalou-Meher (JAM) scaling method [84] and derived fast algorithms for the best performing approximations.

Next, we present the considered design parameters to obtain the new approximate transforms, their performance assessment and the proposed fast algorithms. To the best of our knowledge, all of the obtained transforms are new in the literature.

5.5.1 Low-complexity matrices for 16-, 32-, and 64-point DCT approximations

We solved the optimization problem described in (5.13) considering the following parameters: (i) $N \in \{16, 32, 64\}$ and (ii) the following sets of low-complexity multipliers:

$$\mathscr{D}_1 = \{0, \pm 1\}, \tag{5.16}$$

$$\mathscr{D}_2 = \{0, \pm \tfrac{1}{2}, \pm 1\}, \tag{5.17}$$

$$\mathscr{D}_3 = \{0, \pm 1, \pm 2\}, \tag{5.18}$$

$$\mathscr{D}_4 = \{0, \pm \tfrac{1}{4}, \pm \tfrac{1}{2}, \pm 1\}, \tag{5.19}$$

$$\mathscr{D}_5 = \{0, \pm \tfrac{1}{2}, \pm 1, \pm 2\}, \tag{5.20}$$

$$\mathscr{D}_6 = \{0, \pm \tfrac{1}{4}, \pm \tfrac{1}{2}, \pm 1, \pm 2\}. \tag{5.21}$$

Such sets were separated because of the low-complexity nature of their elements. In fact, multiplications by such elements require only bit-shifting operations.

The mapping from $\mathbf{T}_N$ to $\hat{\mathbf{C}}_N$ shown in (5.7) is not one-to-one, i.e., distinct low-complexity matrices might result in the same approximation. For instance, the following 16-point low-complexity matrices, $\mathbf{T}'_{16}$ and $\mathbf{T}''_{16}$, lead to a single approximation,

$$\mathbf{T}'_{16} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & & & & & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & & & -1 & -1 & -1 & -1 & -1 & -1 & & & 1 & 1 & 1 \\
1 & 1 & & -1 & -1 & -1 & -1 & & & 1 & 1 & 1 & 1 & & -1 & -1 \\
1 & & & -1 & -1 & & & 1 & 1 & & & -1 & -1 & & & 1 \\
1 & & -1 & -1 & & 1 & 1 & 1 & -1 & -1 & -1 & & 1 & 1 & & -1 \\
1 & & -1 & -1 & 1 & 1 & & -1 & -1 & & 1 & 1 & -1 & -1 & & 1 \\
1 & -1 & -1 & & 1 & & -1 & -1 & 1 & 1 & & -1 & & 1 & 1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & & 1 & & -1 & 1 & 1 & -1 & -1 & 1 & & -1 & & 1 & -1 \\
1 & -1 & & 1 & -1 & & 1 & -1 & -1 & 1 & & -1 & 1 & & -1 & 1 \\
1 & -1 & 1 & & -1 & 1 & & -1 & 1 & & -1 & 1 & & -1 & 1 & -1 \\
 & -1 & 1 & & & 1 & -1 & & & -1 & 1 & & & 1 & -1 & \\
 & -1 & 1 & -1 & 1 & & -1 & 1 & -1 & 1 & & -1 & 1 & -1 & 1 & \\
 & -1 & 1 & -1 & 1 & -1 & 1 & & & 1 & -1 & 1 & -1 & 1 & -1 & \\
 & & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & &
\end{bmatrix}, \tag{5.22}$$

$$\mathbf{T}''_{16} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
2 & 2 & 2 & 2 & 2 & & & & & & -2 & -2 & -2 & -2 & -2 & -2 \\
1 & 1 & & & & -1 & -1 & -1 & -1 & -1 & -1 & & & 1 & 1 & 1 \\
2 & 2 & & -2 & -2 & -2 & -2 & & & 2 & 2 & 2 & & & -2 & -2 \\
2 & 2 & -2 & -2 & -2 & -2 & 2 & 2 & 2 & 2 & -2 & -2 & -2 & -2 & & 2 \\
2 & & -2 & -2 & & 2 & 2 & & -2 & -2 & -2 & & 2 & 2 & & -2 \\
2 & & -2 & -2 & & 2 & & -2 & -2 & & 2 & 2 & -2 & -2 & & 2 \\
1 & -1 & -1 & & 1 & & -1 & -1 & 1 & 1 & & -1 & & 1 & & -1 \\
\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\
2 & -2 & & 2 & & -2 & & 2 & -2 & -2 & 2 & & -2 & & 2 & -2 \\
 & -2 & & 2 & -2 & & 2 & -2 & -2 & 2 & & -2 & 2 & & -2 & 2 \\
 & -2 & 2 & & -2 & 2 & & -2 & 2 & & -2 & 2 & & -2 & 2 & -2 \\
2 & -2 & 2 & -2 & -2 & 2 & -2 & 2 & 2 & -2 & 2 & -2 & -2 & 2 & -2 & \\
 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & & & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \\
 & -2 & 2 & -2 & 2 & -2 & & & & 2 & -2 & 2 & -2 & 2 & -2 & \\
 & & & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & &
\end{bmatrix}, \tag{5.23}$$

since

$$\left(\sqrt{[\mathrm{diag}(\mathbf{T}'_{16}\cdot\mathbf{T}'^{\top}_{16})]^{-1}}\right)\cdot\mathbf{T}'_{16} = \left(\sqrt{[\mathrm{diag}(\mathbf{T}''_{16}\cdot\mathbf{T}''^{\top}_{16})]^{-1}}\right)\cdot\mathbf{T}''_{16}. \tag{5.24}$$

Such matrices are referred to as equivalent [127] and can be grouped into equivalent classes.

Table 17 summarizes the number of matrices, equivalence classes obtained for 16-, 32-, and 64-point low-complexity matrices, and the sets considered for each blocklength. We do not consider all sets for the 32- and 64-point because of the computation time. The 16-, 32-, and

Table 17 – Total matrices and classes of equivalence obtained for the 16-, 32-, and 64-point DCT

Sets considered	N	Number of matrices obtained	Number of classes of equivalence
$\mathscr{D}_1$, $\mathscr{D}_2$, $\mathscr{D}_3$, $\mathscr{D}_4$, $\mathscr{D}_5$, $\mathscr{D}_6$	16	156	5
$\mathscr{D}_1$, $\mathscr{D}_2$, $\mathscr{D}_3$	32	3	2
$\mathscr{D}_1$	64	1	1

Source: Author (2023).

64-point low-complexity matrices obtained are denoted here as $\mathbf{T}_{N,i}$, where i and N refers to the equivalence class and the length of the low-complexity matrix, respectively. For the case $N = 16$, we have five new approximations obtained directly from the proposed method; for $N = 32$, two new matrices; and, considering $N = 64$, we have only one new matrix. The best performing matrices (as discussed in Section 5.5.3) are numerically shown in Appendix C.

Due to the size of the search space for some settings, we had to reduce the number of matrices according to the following procedure:

1. Split the exact DCT into two matrices as follows:

$$\mathbf{C}_N = \text{abs}(\mathbf{C}_N) \circ \text{sign}(\mathbf{C}_N), \tag{5.25}$$

   where $\text{abs}(\cdot)$ [228, pg. 10] returns the absolute value of its input, $\text{sign}(\cdot)$ is the entry-wise signum function [13, pg. 286], and $\circ$ represents the elementwise multiplication [127, pg. 251];
2. Select $\mathscr{D}$ and remove its negative elements, *e.g.*: $\mathscr{D}_3^+ = \{0,1,2\}$;
3. Rewrite (5.13) replacing $\mathbf{c}_k$ and $\mathscr{D}$ with $\text{abs}(\mathbf{c}_k)$ and $\mathscr{D}^+$, respectively, as follows:

$$\mathbf{t}_k^* = \underset{\mathbf{p}\in(\mathscr{D}^+)^N}{\arg\min}\, \text{angle}(\mathbf{p}, \text{abs}(\mathbf{c}_k)), \quad k = 0,1,\ldots,N-1; \tag{5.26}$$

4. Use (5.26) to approximate the rows of $\text{abs}(\mathbf{C}_N)$ individually and obtain the low-complexity matrix $\mathbf{T}_N^*$;
5. Obtain the low-complexity $\mathbf{T}_N$ by

$$\mathbf{T}_N = \mathbf{T}_N^* \circ \text{sign}(\mathbf{C}_N). \tag{5.27}$$

To further reduce the search space, the symmetries of $\text{abs}(\mathbf{C}_N)$ were used. Notice that the jth and the $(N-j-1)$th column of $\text{abs}(\mathbf{C}_N)$ are equals. Thus, we only need to find approximations for half of the values in each row. Using this property in (5.26), the computation requires $N \cdot (D^+)^{\frac{N}{2}}$ angle evaluations at most. This search space reduction was only adopted for $N = 32$ considering $\mathscr{D}_2$ and $\mathscr{D}_3$; and for $N = 64$ considering $\mathscr{D}_1$.

5.5.2 JAM scaling method

In addition to the obtained matrices, we also derive scaled approximations according to the JAM scaling method described in [84]. A scaled approximation derived from the JAM method consists of a $2N$-point transformation based on an N-point approximation related as

follows:

$$\mathbf{T}_{2N} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{T}_N & \\ & \mathbf{T}_N \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_N & \bar{\mathbf{I}}_N \\ \bar{\mathbf{I}}_N & \mathbf{I}_N \end{bmatrix}, \tag{5.28}$$

where $\mathbf{T}_N$ is an N-point approximation, $\mathbf{I}_N$ and $\bar{\mathbf{I}}_N$ are, respectively, the N-point identity and counter-identity matrices. Matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ are permutations of $2N \times N$ dimension. Matrix $\mathbf{P}_1$ contains ones in positions $(2i,i), i = 0,\ldots,N-1$, and zeros elsewhere; whereas $\mathbf{P}_2$ presents ones in positions $(2i+1,i), i = 0,\ldots,N-1$, and zeros elsewhere.

Thus, we scaled the low-complexity matrices that generate the best performing transforms for N =16 and 32, according to the assessment measurements displayed in Table 18, through the JAM method. The scaling method is a function f defined according to:

$$f: \quad \mathbb{R}^{N\times N} \longmapsto \mathbb{R}^{2N\times 2N} \tag{5.29}$$

$$\mathbf{T}_N \longmapsto \mathbf{T}_{2N} = f(\mathbf{T}_N). \tag{5.30}$$

We denote $\mathbf{T}_N^{(j)}$ as the $2^j \cdot N \times 2^j \cdot N$ matrix based on $\mathbf{T}_N$,

$$\mathbf{T}_N^{(j)} \triangleq f(f(\ldots f(\mathbf{T}_N))), \tag{5.31}$$

where the function f is applied j times.

The proposed method in [84] has a term of $\frac{1}{\sqrt{2}}$ that multiplies (5.28) which was previously omitted in the scaling to be integrated in the final quantization. Thus, the approximation $\widehat{\mathbf{C}}_N^{(j)}$ is obtained by

$$\widehat{\mathbf{C}}_N^{(j)} = \left(\frac{1}{\sqrt{2}}\right)^j \cdot \mathbf{S}_N^* \cdot \mathbf{T}_N^{(j)}, \tag{5.32}$$

where $\mathbf{S}_N^*$ is computed from $\mathbf{T}_N^{(j)}$ as detailed in (5.8). For instance, $\widehat{\mathbf{C}}_{16}^{(2)}$ is the result of two JAM applications in a 16-point matrix, resulting in a 64-point matrix.

5.5.3 Performance assessment

We evaluated the proposed approximations according to similarity measures and coding measures. For similarity measures, we considered the mean square error (MSE) [13, 14] and the total energy error (ε) [72]; whereas for coding measures, we adopted the unified coding gain (C_g) [229] and the transform efficiency (η) [13]. Based on the deviation from diagonality [170], we can also quantify the deviation from orthogonality ($\delta(\cdot)$) of the discussed approximations;

this measure informs how close to orthogonality a matrix is and is given by

$$\delta(\mathbf{A}) = 1 - \frac{||\operatorname{diag}(\mathbf{A})||_F}{||\mathbf{A}||_F}, \tag{5.33}$$

where $\mathbf{A}$ is a square matrix and $||\cdot||_F$ is the Frobenius norm for matrices [127].

For comparison purposes, we considered a comprehensive set of approximations archived in literature. For $N = 16$, the following approximations were separated: (i) the approximation proposed by Haweel ($\widehat{\mathbf{C}}_{16,\text{SDCT}}$) [71] (ii) the approximation proposed by Jridi, Alfalou, and Meher ($\widehat{\mathbf{C}}_{16,\text{JAM}}$) [84]; (iii) the approximation proposed by Bouguezel, Ahmad, and Swamy, ($\widehat{\mathbf{C}}_{16,\text{BAS}}$), in [78]; (iv) the approximation proposed in [131], ($\widehat{\mathbf{C}}_{16,\text{BCEM}}$); (v) the approximation introduced in [85], ($\widehat{\mathbf{C}}_{16,\text{SOBCM}}$), and (vi) the scaled approximation proposed in [80] ($T_{(16)}$), ($\widehat{\mathbf{C}}_{16,\text{OCBSML}}$).

For $N = 32$, we considered the following methods: (i) the approximation proposed by Haweel ($\widehat{\mathbf{C}}_{32,\text{SDCT}}$) [71]; (ii) the approximation proposed by Jridi, Alfalou, and Meher ($\widehat{\mathbf{C}}_{32,\text{JAM}}$) [84]; (iii) the approximation proposed by Bouguezel, Ahmad, and Swamy, ($\widehat{\mathbf{C}}_{32,\text{BAS}}$), in [78]; (vi) the scaled approximation proposed in [80] ($T_{(32)}$), ($\widehat{\mathbf{C}}_{32,\text{OCBSML}}$).

For $N = 64$, we considered the following approximations: (i) the approximation proposed by Haweel ($\widehat{\mathbf{C}}_{64,\text{SDCT}}$) [71]; (ii) the scaled version of the approximation proposed in [80], ($\widehat{\mathbf{C}}_{64,\text{OCBSML}}$).

Table 18 displays the results for the performance measures of the proposed approximations along with the competing approximations and the exact N-point DCT ($\mathbf{C}_N$). For each value of N, we highlighted the best measurements of each metric. We identify $\widehat{\mathbf{C}}_{16,5}$, $\widehat{\mathbf{C}}_{32,2}$, $\widehat{\mathbf{C}}^{(1)}_{16,5}$, $\widehat{\mathbf{C}}_{64,1}$, and $\widehat{\mathbf{C}}^{(2)}_{16,5}$ as the best transforms for $N = 16$, 32, and 64, respectively. Approximations $\widehat{\mathbf{C}}_{16,5}$, $\widehat{\mathbf{C}}_{32,2}$, and $\widehat{\mathbf{C}}_{64,1}$ are the low-complexity matrices obtained from the fifth, second, and first class of equivalence for $N = 16$, 32, and 64, respectively. The approximations $\widehat{\mathbf{C}}^{(1)}_{16,5}$ and $\widehat{\mathbf{C}}^{(2)}_{16,5}$ are the scaled approximations based on $\widehat{\mathbf{C}}_{16,5}$. These proposed approximations outperformed the DCT approximations already known in the literature.

5.5.4 Fast algorithms

To reduce the arithmetic cost of implementing the proposed transforms, we factorized the optimal transforms into sparse matrices using common decimation-based techniques [23, pg. 74]. The factorization for the proposed transforms was developed using butterfly-based structures, such as in [55, 72, 81, 164, 230]. The complexity of the fast algorithms was evaluated in terms of the number of arithmetic operations. The arithmetic complexity does not depend on the

Table 18 – Performance measures for the DCT approximations in literature and the new approximations proposed

Approximation	$\varepsilon(\widehat{\mathbf{C}})$	MSE$(\widehat{\mathbf{C}})$	$C_g(\widehat{\mathbf{C}})$	$\eta(\widehat{\mathbf{C}})$	$\delta(\widehat{\mathbf{C}})$
		$N = 16$			
$\mathbf{C}_{16}$	0	0	9.4555	88.4518	0
$\widehat{\mathbf{C}}_{16,1}$	3.7043	0.0172	7.7474	70.5034	0.0828
$\widehat{\mathbf{C}}_{16,2}$	3.7043	0.0172	8.2190	70.6902	0.0270
$\widehat{\mathbf{C}}_{16,3}$	1.0227	0.0054	8.9653	78.4016	0.0472
$\widehat{\mathbf{C}}_{16,4}$	0.6337	0.0035	9.0922	80.1145	0.0234
$\widehat{\mathbf{C}}_{16,5}$	**0.5748**	**0.0031**	**9.1268**	**80.4401**	0.0118
$\widehat{\mathbf{C}}_{16,\text{SDCT}}$	8.2537	0.0429	6.0297	64.9653	0.2
$\widehat{\mathbf{C}}_{16,\text{JAM}}$	14.7402	0.0506	8.4285	72.2296	0
$\widehat{\mathbf{C}}_{16,\text{BAS}}$	16.4071	0.0564	8.5208	73.6345	0
$\widehat{\mathbf{C}}_{16,\text{BCEM}}$	8.0806	0.0465	7.8401	65.2789	0
$\widehat{\mathbf{C}}_{16,\text{SOBCM}}$	40.9996	0.0947	7.8573	67.6078	0
$\widehat{\mathbf{C}}_{16,\text{OCBSML}}$	13.7032	0.0474	8.8787	76.8108	0
		$N = 32$			
$\mathbf{C}_{32}$	0	0	9.7736	81.6962	0
$\widehat{\mathbf{C}}_{32,1}$	7.6403	0.0287	7.4624	52.5455	0.1138
$\widehat{\mathbf{C}}_{32,2}$	**2.3525**	**0.0100**	9.0983	64.9265	0.0376
$\widehat{\mathbf{C}}^{(1)}_{16,5}$	30.0539	0.0829	**9.1939**	**64.9983**	0.0118
$\widehat{\mathbf{C}}_{32,\text{SDCT}}$	18.2386	0.0748	5.5623	41.6653	0.2727
$\widehat{\mathbf{C}}_{32,\text{JAM}}$	48.0956	0.1124	8.5010	56.9700	0
$\widehat{\mathbf{C}}_{32,\text{BAS}}$	57.1260	0.1171	8.4971	58.1727	0
$\widehat{\mathbf{C}}_{32,\text{OCBSML}}$	46.2658	0.1104	8.9505	61.0272	0
		$N = 64$			
$\mathbf{C}_{64}$	0	0	9.9366	75.55406	0
$\widehat{\mathbf{C}}_{64,1}$	**15.5707**	**0.0434**	7.2436	36.4275	0.1152
$\widehat{\mathbf{C}}^{(2)}_{16,5}$	103.2435	0.1833	**9.2144**	**51.6925**	0.0118
$\widehat{\mathbf{C}}^{(1)}_{32,2}$	66.8310	0.1355	9.1164	51.2582	0.0376
$\widehat{\mathbf{C}}_{64,\text{SDCT}}$	38.2630	0.1141	5.2192	27.9725	0.2809
$\widehat{\mathbf{C}}_{64,\text{OCBSML}}$	125.2247	0.2015	8.9748	48.4443	0

Source: Author (2023).

available technology, an issue that occurs in measures such as computation time [130, 231]. Table 19 presents the arithmetic complexity of the proposed transforms before and after the matrix factorization and also the percentage of complexity reduction.

Although the matrices $\mathbf{T}_{16,\text{OCBSML}}$, $\mathbf{T}_{32,\text{OCBSML}}$, and $\mathbf{T}_{64,\text{OCBSML}}$ [80] are more benefited by factorization, with a few more operations, the proposed approximations can be used, providing better performance. According to Table 18 and 19, when comparing $\mathbf{T}_{16,\text{OCBSML}}$ and $\mathbf{T}_{16,5}$,

Table 19 – The arithmetic complexity of the proposed transforms before and after the matrix factorization.

	Before factorization			After factorization		Reduction (%)	
Matrix	Mult	Adds	Bit-shifting	Adds	Bit-shifting	Adds	Bit-shifting
			$N = 16$				
$\mathbf{C}_{16}$	256	240	0	-	-	-	-
$\mathbf{T}_{16,1}$	0	184	0	82	0	55.44	0
$\mathbf{T}_{16,2}$	0	192	0	80	0	58.33	0
$\mathbf{T}_{16,3}$	0	208	80	88	30	57.69	62.50
$\mathbf{T}_{16,4}$	0	224	96	92	34	58.93	73.44
$\mathbf{T}_{16,5}$	0	240	160	100	62	58.33	64.77
$\mathbf{T}_{16,\text{OCBSML}}$	0	208	96	64	12	69.23	79.17
			$N = 32$				
$\mathbf{C}_{32}$	1024	992	0	-	-	-	-
$\mathbf{T}_{32,1}$	0	752	0	287	0	61.83	0
$\mathbf{T}_{32,2}$	0	864	320	328	110	62.04	65.62
$\mathbf{T}^{(1)}_{16,5}$	0	992	640	232	124	76.61	80.62
$\mathbf{T}_{32,\text{OCBSML}}$	0	864	384	160	24	81.48	89.58
			$N = 64$				
$\mathbf{C}_{64}$	4096	4032	0	-	-	-	-
$\mathbf{T}_{64,1}$	0	3040	0	1087	0	64.24	0
$\mathbf{T}^{(2)}_{16,5}$	0	4032	2560	528	248	86.90	90.31
$\mathbf{T}^{(1)}_{32,2}$	0	3520	1280	720	220	79.54	82.81
$\mathbf{T}_{64,\text{OCBSML}}$	0	3520	1536	384	48	89.09	94.79

Source: Author (2023).

the proposed approximation requires only 36 more additions but it presents a reduction of approximately 96% and 93% in terms of energy error and MSE, respectively; and a gain of approximately 2.8% in coding gain and 4.7% in transform efficiency.

Considering $N = 32$, the proposed approximation $\mathbf{T}_{32,1}$ needs 127 extra additions than $\mathbf{T}_{32,\text{OCBSML}}$ and it presents a reduction of approximately 95% and 93% of energy error and MSE, respectively; and a improvement of 1.6% in coding gain and 6.4% in transform efficiency. In addition, the proposed approximation $\mathbf{T}^{(1)}_{16,5}$ requires 72 more additions than $\mathbf{T}_{32,\text{OCBSML}}$ and it presents a reduction of 54% and 33% of energy error and MSE, respectively; and a gain of approximately 2.7% in coding gain and 6.5% in transform efficiency. Approximations for $N = 64$ (except the SDCT) were first presented in this chapter.

For better understanding and reproducibility, each sparse matrix used to obtain low-complexity matrices associated with the optimal transforms $\mathbf{T}_{16,5}$, $\mathbf{T}_{32,2}$, and $\mathbf{T}_{64,1}$ are detailed in the Appendix C.

5.6 IMAGE COMPRESSION EXPERIMENTS

To evaluate the performance of the proposed 16-, 32-, and 64-point DCT approximations we performed JPEG-like image compression experiments, as in [72, 81, 219]. We considered 45 8-bit images of 512×512 obtained from [137]. All images were subdivided into $N \times N$ sub-blocks and were submitted to the following 2D transformation. Let $\mathbf{A}$ be an $N \times N$ sub-block. The direct and inverse transformation induced by $\widehat{\mathbf{C}}_N$ are given, respectively, by [53, 72, 81]

$$\mathbf{B} = \widehat{\mathbf{C}}_N \cdot \mathbf{A} \cdot \widehat{\mathbf{C}}_N^{-1} \quad \text{and} \quad \mathbf{A} = \widehat{\mathbf{C}}_N^{-1} \cdot \mathbf{B} \cdot \widehat{\mathbf{C}}_N, \tag{5.34}$$

where $\mathbf{A}$ and $\mathbf{B}$ are matrices of size $N \times N$. To evaluate the exact DCT, the transformation matrix $\widehat{\mathbf{C}}_N$ and $\widehat{\mathbf{C}}_N^{-1}$ are replaced by $\mathbf{C}_N$ and $\mathbf{C}_N^{-1}$, respectively. Considering the zig-zag pattern [16, p. 30], we retained the initial r coefficients from each sub-block $\mathbf{B}$. Finally, we applied the inverse 2D transform in each sub-block, and then the compressed images are obtained.

Original and compressed images were evaluated considering usual quality assessment measures: (i) the mean square error (MSE) [13], (ii) the peak signal-to-noise ratio (PSNR) [16], and (iii) the mean structural similarity index (MSSIM) [136]. Notice that the MSE measures are computed considering the original and the compressed images; not to be confused with the MSE calculation described in Section 5.5.3. Although the MSE and PSNR measures are popular in the context of image compression, it was shown in [157] that they might offer limited results as image quality tools. On the other hand, the MSSIM was shown to be a better measure when it comes to capturing the image quality as understood by the human visual system model [136, 157].

The image compression experiments were divided into two steps: (i) a qualitative analysis where we considered the compressed *Peppers* image with a compression rate of approximately 80%; (ii) and a quantitative analysis, where we considered the average measurements from 45 standardized images. Both analysis are presented next.

5.6.1 Qualitative analysis

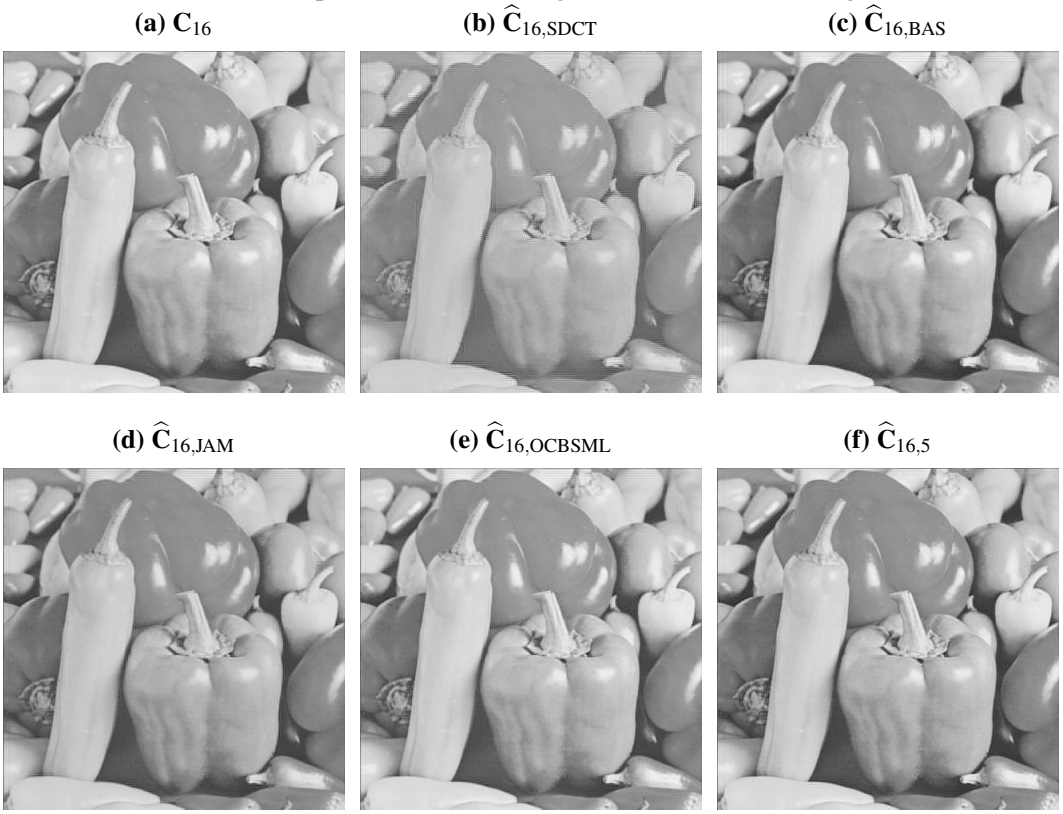
For the qualitative analysis, we considered the compressed *Peppers* image [137], with a compression rate (CR) of approximately 80%. The compression rate is defined by

$$\text{CR} = 1 - \frac{r}{N^2}. \tag{5.35}$$

Figs. 34, 35, and 36 display the compressed *Peppers* images with the DCT and approximations, for $N = 16$, 32, and 64, respectively. Visually, the reconstructed images after the compression

with the proposed transforms exhibit quality comparable to the ones compressed using the exact DCT and known approximations. For a better representation, we present the MSE, PSNR, and MSSIM of these images in Table 20. The best results for each sample image and measure are highlighted. **The proposed approximations for the 16-, 32-, and 64-point DCT show better results than the approximations in literature.** To the best of our knowledge, there is no 64-point DCT approximation in the literature for comparison, except for the $\widehat{\mathbf{C}}_{64,\mathrm{SDCT}}$ [71].

Figure 34 – Compressed *Peppers* image for $N = 16$ considering $r = 50$.



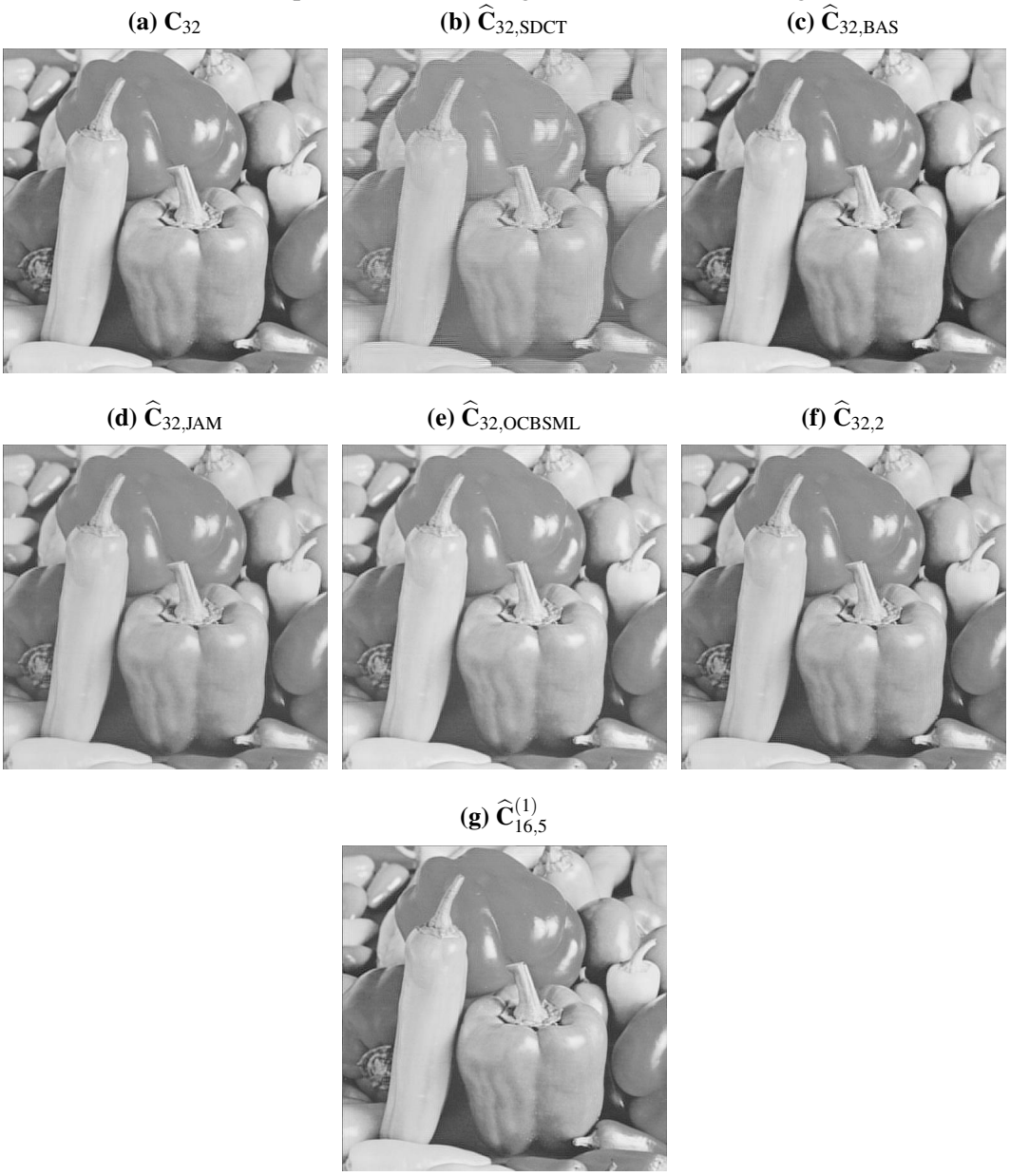
Source: Author (2023).

5.6.2 Quantitative analysis

In this analysis, we considered the measurements of the selected 45 8-bit images obtained from a public image bank [137]. Each image was compressed considering the initial r coefficients (matrix elements ordered according to the zig-zag pattern [16, pg. 30]), $r \in \{0, 1, \ldots N^2\}$, and assessed by the quality measures.

Fig. 37, 38, and 39 show the average curves from the 45 images for the MSE, PSNR, and MSSIM, respectively. For better visualization, we adopted the Absolute Percentage Error (APE)

Figure 35 – Compressed *Peppers* image for $N = 32$ considering $r = 205$.



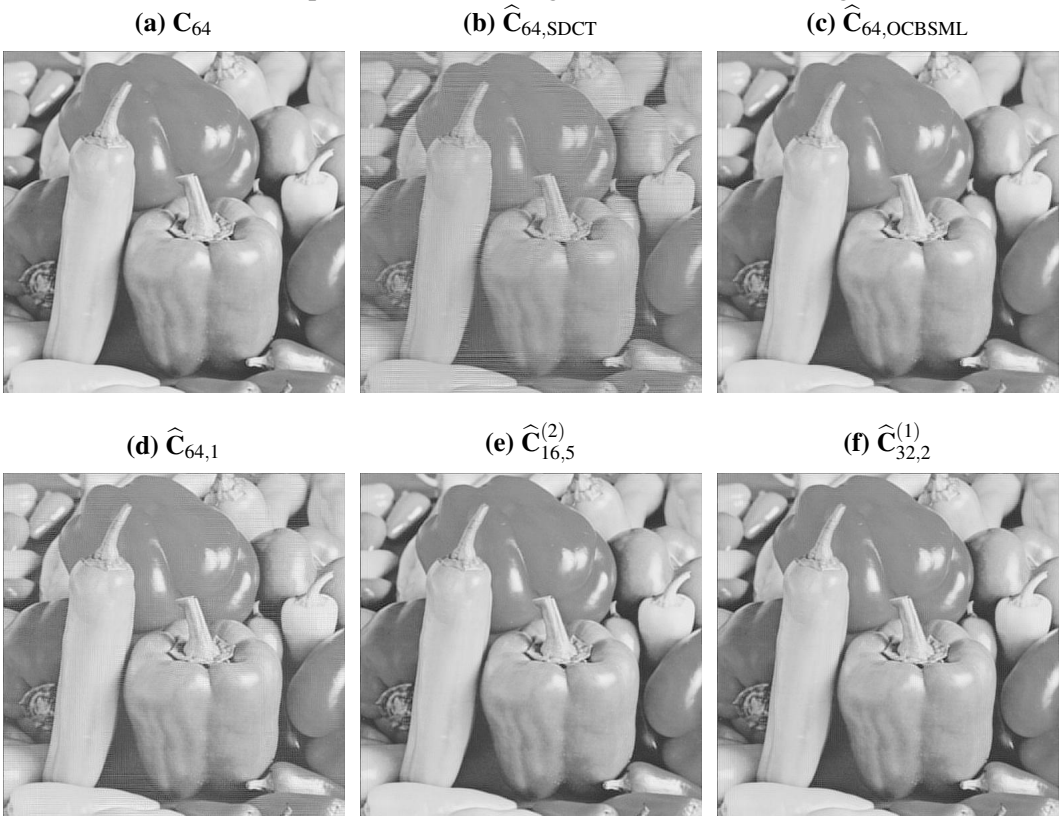
Source: Author (2023).

relative to the DCT:

$$\text{APE}(\mu) = \left| \frac{\mu(\mathbf{C}_N) - \mu(\widehat{\mathbf{C}}_N)}{\mu(\mathbf{C}_N)} \right|, \quad \mu \in \{\text{MSE}, \text{PSNR}, \text{MSSIM}\}, \tag{5.36}$$

where $\mu(\mathbf{C}_N)$ and $\mu(\widehat{\mathbf{C}}_N)$ indicate the measurements according to the exact and approximate N-point DCT, respectively. For this evaluation, we considered the optimal proposed transforms $\widehat{\mathbf{C}}_{16,5}$, $\widehat{\mathbf{C}}_{32,2}$, $\widehat{\mathbf{C}}^{(1)}_{16,5}$, $\widehat{\mathbf{C}}_{64,1}$, $\widehat{\mathbf{C}}^{(2)}_{16,5}$, and $\widehat{\mathbf{C}}^{(1)}_{32,2}$ and compared with the exact DCT ($\mathbf{C}_N$) and the three best approximations according to Table 18: $\widehat{\mathbf{C}}_{N,\text{BAS}}$, $\widehat{\mathbf{C}}_{N,\text{JAM}}$, and $\widehat{\mathbf{C}}_{N,\text{OCBSML}}$. **The proposed approximations are the ones with the best results when contrasted with competing DCT approximations for** $N = 16$**,** 32**, and** 64**.**

Figure 36 – Compressed *Peppers* image for $N = 64$ considering $r = 820$.



Source: Author (2023).

5.7 CONCLUSIONS

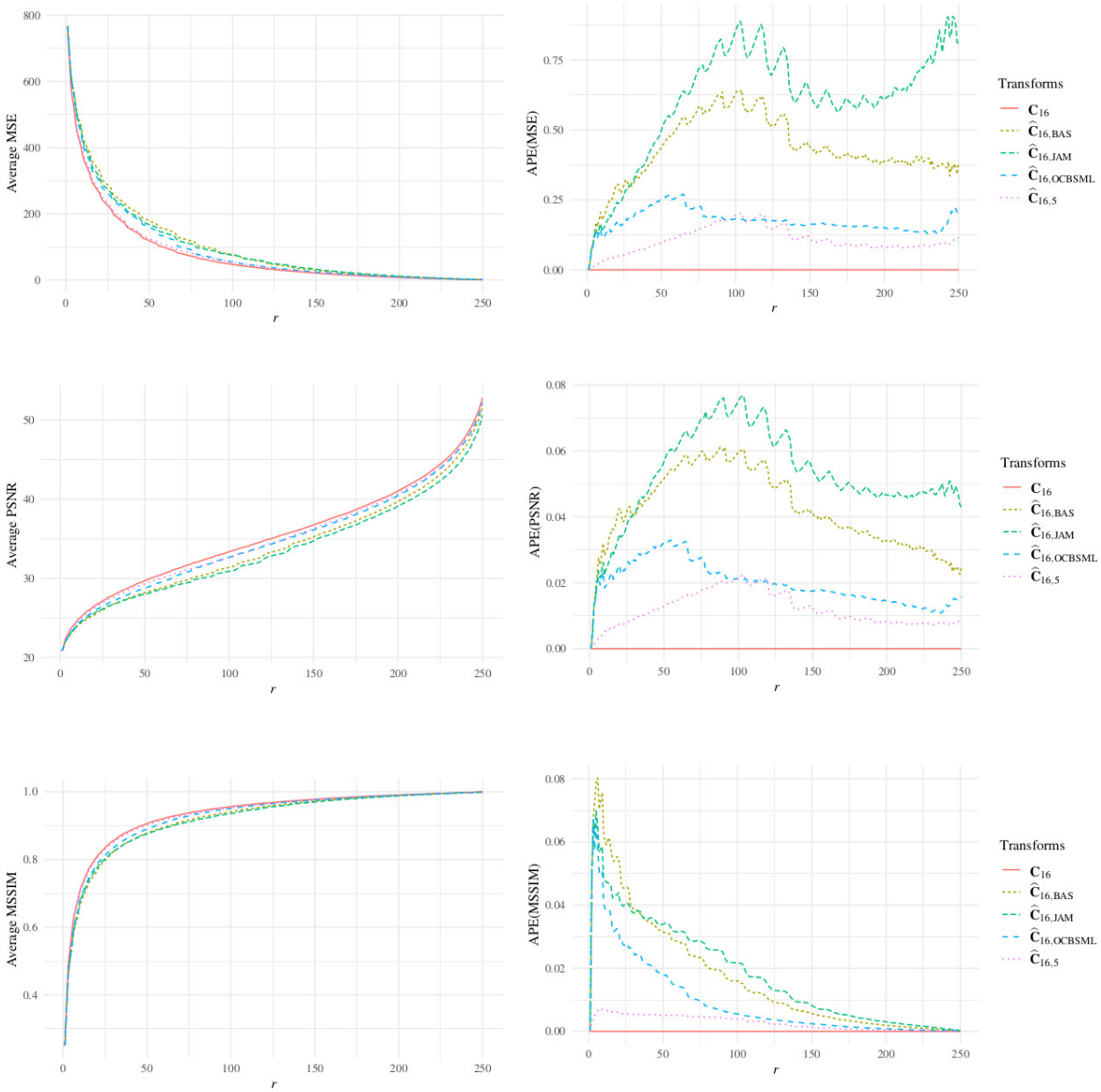
This chapter introduces low-complexity approximations for the DCT of lengths 16, 32, and 64, based on the method proposed in [80], which approximates the rows of the exact and the approximate matrix transform according to the angle between them. The proposed transforms were assessed according to classical figures of merit and showed a better performance when compared to the DCT approximations already archived in the literature. Fast algorithms were also derived for the best transforms of each blocklength, which further reduced their arithmetic costs. Only additions and bit-shifting operations are necessary for their computation. Furthermore, the transforms were evaluated in the context of image compression. *The experiments demonstrated that the proposed transforms outperformed all the considered DCT approximations already known in the literature for* $N = 16$*,* 32*, and* 64*.*

Table 20 – Image quality measures of compressed *Peppers* image for $N = 16(r = 50)$, $32(r = 205)$, and $64(r = 820)$.

N	Approximation	MSE	PSNR	MSSIM
16	$\mathbf{C}_{16}$	**32.3449**	**33.0328**	**0.9416**
	$\widehat{\mathbf{C}}_{16,1}$	75.0342	29.3782	0.9154
	$\widehat{\mathbf{C}}_{16,2}$	70.1802	29.6687	0.9184
	$\widehat{\mathbf{C}}_{16,3}$	43.2034	31.7756	0.9355
	$\widehat{\mathbf{C}}_{16,4}$	39.0721	32.2121	**0.9378**
	$\widehat{\mathbf{C}}_{16,5}$	**38.8975**	**32.2316**	**0.9378**
	$\widehat{\mathbf{C}}_{16,\mathrm{SDCT}}$	135.8288	26.8009	0.8905
	$\widehat{\mathbf{C}}_{16,\mathrm{JAM}}$	59.6891	30.3718	0.9242
	$\widehat{\mathbf{C}}_{16,\mathrm{BAS}}$	58.1761	30.4834	0.9248
	$\widehat{\mathbf{C}}_{16,\mathrm{BCEM}}$	93.1786	28.4376	0.9064
	$\widehat{\mathbf{C}}_{16,\mathrm{SOBCM}}$	115.2820	27.5132	0.8897
	$\widehat{\mathbf{C}}_{16,\mathrm{OCBSML}}$	49.3669	31.1964	0.9303
32	$\mathbf{C}_{32}$	**29.2562**	**33.4686**	**0.9690**
	$\widehat{\mathbf{C}}_{32,1}$	87.8409	28.6938	0.9443
	$\widehat{\mathbf{C}}_{32,2}$	43.5848	31.7375	0.9634
	$\widehat{\mathbf{C}}^{(1)}_{16,5}$	**37.8366**	**32.3517**	**0.9652**
	$\widehat{\mathbf{C}}_{32,\mathrm{SDCT}}$	189.2796	25.3598	0.9122
	$\widehat{\mathbf{C}}_{32,\mathrm{JAM}}$	58.4496	30.4630	0.9554
	$\widehat{\mathbf{C}}_{32,\mathrm{BAS}}$	61.6647	30.2304	0.9524
	$\widehat{\mathbf{C}}_{32,\mathrm{OCBSML}}$	47.9872	31.3196	0.9591
64	$\mathbf{C}_{64}$	**28.1330**	**33.6386**	**0.9880**
	$\widehat{\mathbf{C}}_{64,1}$	118.8871	27.3795	0.9641
	$\widehat{\mathbf{C}}^{(2)}_{16,5}$	**37.3748**	**32.4050**	**0.9851**
	$\widehat{\mathbf{C}}^{(1)}_{32,2}$	42.8390	31.8124	0.9836
	$\widehat{\mathbf{C}}_{64,\mathrm{SDCT}}$	273.3575	23.7635	0.9294
	$\widehat{\mathbf{C}}_{64,\mathrm{OCBSML}}$	54.2638	30.7857	0.9794

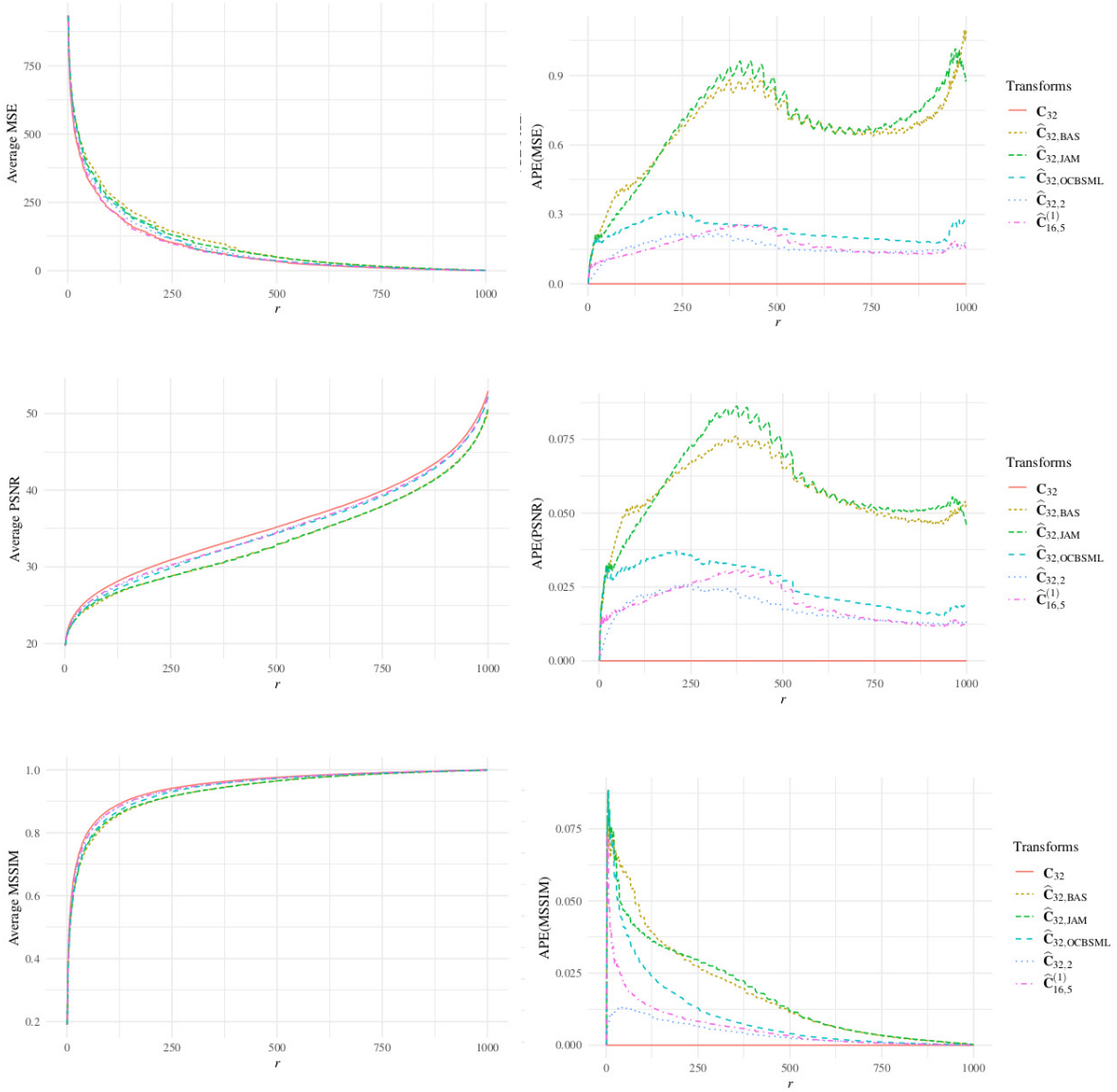
Source: Author (2023).

Figure 37 – Average curves for the MSE, PSNR, and MSSIM for the 16-point DCT and approximations.



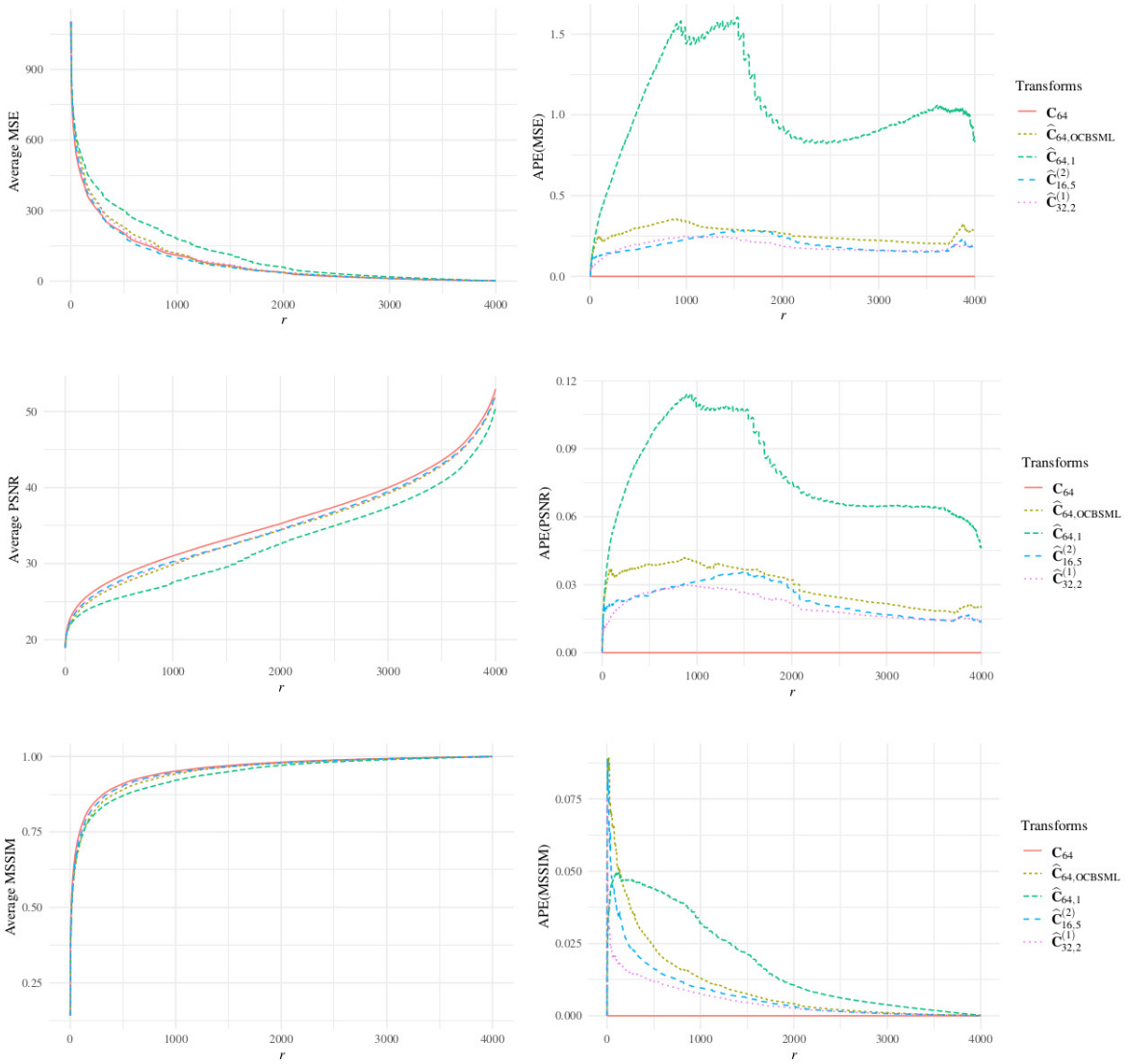
Source: Author (2023).

Figure 38 – Average curves for the MSE, PSNR, and MSSIM for the 32-point DCT and approximations.



Source: Author (2023).

Figure 39 – Average curves for the MSE, PSNR, and MSSIM for the 64-point DCT and approximations.



Source: Author (2023).

6 LOW-COMPLEXITY METHODS FOR SIGNAL DETECTION

6.1 INTRODUCTION

The use of the discrete Fourier transform (DFT) finds applications in the most different fields of the digital signal processing context [17, 232], because its relevance as a design and analysis tool [233], and also because of the existence of several efficient algorithms for its computation [23, 231]. A useful application of the DFT is the signal detection theory, which describes the capacity of a receptor to recognize a signal [232]. This process of detection can be defined as a statistical hypothesis test, where the null hypothesis (H_0) is that only noise is present while the alternative hypothesis (H_1) is the presence of signal+noise [232]. Signal detection is a fundamental tool to the design of efficient signal processing systems for decision making [234] and its widely used in the most different scenarios, such as radar [235, 236], communications [237, 238], speech [239, 240], sonar [241], image processing [242, 243], biomedicine [244, 245], seismology [246, 247], among others [232].

The detection problem can be linked to the Discrete Fourier Transform because the calculation of a statistical test, which is used to determine whether to accept or reject the null hypothesis, can be based on the computation of the DFT [40, 248]. Even though the use of a fast Fourier transform (FFT) can be an efficient alternative, the computation of the statistical test can still be an issue in scenarios where agility in the detection is needed.

In the past decades, many works were developed in order to make the computational realization of the DFT even more efficient [64, 101, 102, 249, 250]. Although the search for efficient fast algorithms for computing the DFT is a quite mature area, there is still room for improvement. For instance, considering signals of length 2^N, the well-known algorithms for the FFT [23] reduce the computational cost of the DFT from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \cdot \log_2 N)$ [231], but they still require multiplications for their implementation. To the best of our knowledge, the literature is scarce in methods devoted to the computation of a multiplierless DFT [102, 103, 106, 107, 251]. Thus, we aim to propose multiplierless approximations for the DFT based on integer functions, on a similar way to what has been proposed for other discrete transforms, such as approximations for the discrete cosine transform (DCT) [71, 72, 81] and for the Karhunen-Loève transform (KLT) [252, 253]. In this sense, **the contributions of this thesis chapter are: (i) to propose low-complexity multiplierless approximations for the DFT and (ii) to propose a low-complexity detector based on the DFT approximations.**

The chapter is structured as follows. In Section 6.2, we present the detection problem as a statistical hypothesis test, as well as the types of errors the statistical test are subject to. In Section 6.3, we present the mathematical formulation of the DFT, its computational complexity and the approximation theory. In Section 6.4, we describe the proposed approximate detector. Section 6.5 presents the simulated experiments for the proposed approximate test. Section 6.6 concludes the chapter.

6.2 STATISTICAL BACKGROUND

In many contexts of signal processing, there is an interest in designing electronic systems for decision-making and information extraction [232]. The main goal of the detection theory is to be able to decide whether an event of interest occurs or not. In statistical jargon, the detection theory is also called hypothesis testing.

6.2.1 Detection Problem

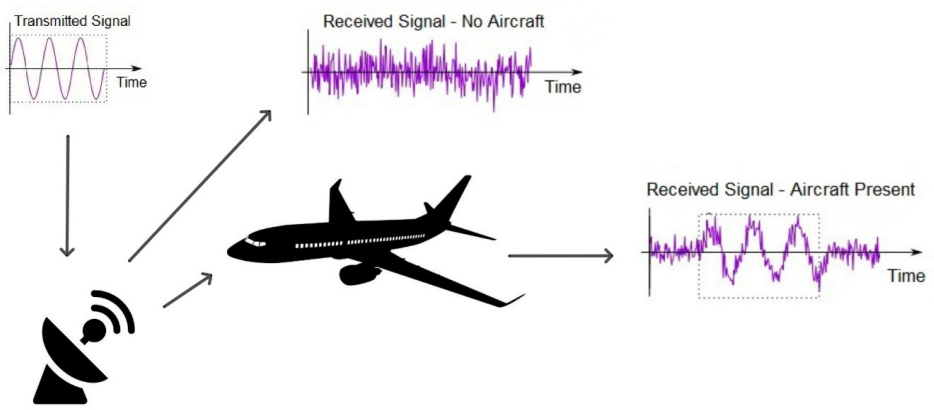
In detection theory, one usual problem is to decide whether a signal is present or not [232]. Generally, the signal is embedded in noise. Thus, the system must detect whether there is signal and noise, or if only noise is present. One example of this problem is detecting an aircraft based on a radar return. The radar transmits a signal that bounces off the target and receives information from the emitting object. If the signal is reflected by a moving object, it will indicate the presence of an aircraft [232, 254]. Figure 40, which is an adaptation from [232, pg. 3], illustrates the decision-making process.

This type of problem is called binary hypothesis testing, once we aim to decide between two possible hypotheses. Let $x[n]$ be a sequence in time that contains the information collected by the radar. Under the null hypothesis H_0, the received sequence corresponds to noise only, i.e., there is no information about the presence of an object (aircraft). Under the alternative hypothesis H_1, the sequence is a signal, i.e., there is information about the presence of an object together with noise. Thus, this decision can be statistical defined as a hypothesis test, where

$$\begin{cases} H_0 : x[n] \sim \text{noise} \\ H_1 : x[n] \sim \text{signal + noise} \end{cases} . \tag{6.1}$$

There are different approaches considered in the context of statistical signal processing for the detection problem [255–258]. The periodogram of a sequence provides the spectral

Figure 40 – Decision-making process of detecting an aircraft by radar pulse emission.



Source: Author (2023).

analysis of the signal and is a useful tool as detector [255]. Hereafter, the focus of this chapter will be on scenarios where the input signal is a simple sinusoidal tone because this is a basic problem in signal detection theory [39, 40]. We aim to determine whether a radar system is detecting a signal or not.

Let the signal x be

$$x[n] = A\sin(2\pi nf + \phi) + q[n], \tag{6.2}$$

where $A \in \mathbb{R}$ is the amplitude, $f \in (0,1)$ is the signal frequency, $\phi \in (0, 2\pi)$ is the phase, and

$$q[n] \sim \mathrm{N}(0, \sigma^2), \quad n = 0, 1, \ldots N-1, \tag{6.3}$$

is the Gaussian white noise. Thus, the hypothesis test is defined by

$$\begin{cases} H_0 : x[n] = q[n], \\ H_1 : x[n] = A\sin(2\pi nf + \phi) + q[n] \end{cases}. \tag{6.4}$$

Note that:

(i) under H_0, x is an independent random sample normally distributed;

(ii) under H_1, x is a simple sinusoidal tone with additive Gaussian noise.

The usual statistical test considered to reject or not the null hypothesis is given by [40]

$$T = 2 \cdot \frac{\max(S(f_k))}{\sigma^2}, \tag{6.5}$$

where $S(f_k)$ is the periodogram which is defined by:

$$S(f_k) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x[n]\exp(-j2\pi nf_k)\right|^2, \quad k = 0, 1, \ldots N/2-1, \tag{6.6}$$

$f_k = k/N$ is the frequency and σ^2 is the power of $q[n]$ [259]. Note that computing the periodogram is the same as computing the DFT of the sequence $x[n]$. Thus, efficient algorithms for the DFT can be used for obtaining efficient detectors. Under H_0, the statistical test T has chi-square distribution with two degrees of freedom [40], $T \sim \chi^2_{(2)}$. If $T > \chi^2_{(\tau,2)}$, where

$$\tau = (1-\alpha)^{\frac{N-2}{2}}, \tag{6.7}$$

then we reject H_0 and assume that a sinusoidal signal is presented. In the case where the power of the signal σ^2 is unknown we can substitute it by the Maximum Likelihood Estimator (MLE). In this case, the distribution for T is asymptotically a chi-square distribution with two degrees of freedom, since the MLE is a consistent estimator.

6.2.1.1 Type I and Type II error

When we perform a hypothesis test, we are subject to making mistakes. The type I error occurs when we reject the null hypothesis H_0 when it is true. That is,

$$P(\text{Type I error}) = P_{H_0}(\text{Reject} \quad H_0) = \alpha, \tag{6.8}$$

where α is called the significance level of the test. The values of $\alpha = 0.05$ or $\alpha = 0.1$ are commonly used in practice.

The type II error occurs when we do not reject H_0 when it is false,

$$P(\text{Type II error}) = P_{H_1}(\text{Do not reject} \quad H_0) = \beta. \tag{6.9}$$

Thus, the **power of the detector** can be defined as

$$\text{Power} = P_{H_1}(\text{Reject} \quad H_0) = 1-\beta. \tag{6.10}$$

Table 21 summarizes this information.

Table 21 – Types of error in hypothesis testing

Decision	H_0 is true	H_0 is false
Do not reject H_0	Correct decision	Type II error
Reject H_0	Type I error	Correct decision

Source: Author (2023).

6.3 DISCRETE FOURIER TRANSFORM COMPUTATION

The DFT is an alternative Fourier representation for finite duration sequences [231]. The DFT of an input signal $x[n]$ can be defined by [17]

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp(-j2\pi nk/N), \quad k = 0, 1, \ldots, N-1, \tag{6.11}$$

and the inverse discrete Fourier transform is given by

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \exp(j2\pi nk/N), \quad n = 0, 1, \ldots, N-1. \tag{6.12}$$

The DFT can also be represented by the following $N \times N$ matrix:

$$\mathbf{F}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)^2} \end{bmatrix}, \tag{6.13}$$

where

$$\omega_{n,k} \triangleq \omega_N^{nk} = \exp(-j2\pi nk/N), \tag{6.14}$$

with $n, k = 0, 1, \ldots, N-1$. Thus, the computation of the DFT of a sequence $x[n]$ is given by

$$\mathbf{X} = \mathbf{F}_N \mathbf{x}, \tag{6.15}$$

where

$$\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \ldots & x_{N-1} \end{bmatrix}^\top \tag{6.16}$$

is the corresponding vector from $x[n]$ and

$$\mathbf{X} = \begin{bmatrix} X_0 & X_1 & \ldots & X_{N-1} \end{bmatrix}^\top. \tag{6.17}$$

is the DFT from $x[n]$. Since the DFT matrix is orthogonal, we have that

$$\mathbf{F}_N^{-1} = \mathbf{F}_N^*, \tag{6.18}$$

where $\mathbf{F}_N^*$ is the $\mathbf{F}_N$ conjugate [227]. If we multiply $\mathbf{F}_N$ by $1/\sqrt{N}$ we have that $|\det(\mathbf{F}_N)| = 1$, where $\det(\cdot)$ is the determinant of a matrix [227].

6.3.1 Computational Complexity

The computational complexity of discrete transforms depends on many different factors, from purely mathematical aspects to details of their physical realization in hardware or software implementation. Combining many of these aspects to reduce the computational cost can be considered a very hard task because there are many factors that are difficult to quantify or control [17, 130]. However, there is one factor that we can control, which is the arithmetic complexity of the transform. The arithmetic complexity can be defined as a function that relates the additive, multiplicative, and bit shift operations. Among these operations, the multiplicative complexity is the one with the most costly operations [23, 260].

Multiplicative operations can be defined in two different ways: trivial and non-trivial multiplications. Trivial multiplications can be performed without using a numerical multiplication algorithm, such as, for example, multiplications by the elements over the set $\{0, \pm 1/2, \pm 1, \pm 2\}$. Multiplications by power-of-two can easily be performed by simple bit shifts in a binary computational architecture. Therefore, only non-trivial multiplications increase the multiplicative complexity.

The direct implementation of the DFT can be defined as

$$X[k] = \sum_{n=0}^{N-1} (\Re\{x[n]\}\Re\{\omega_N^{nk}\} - \Im\{x[n]\}\Im\{\omega_N^{nk}\}) \pm j(\Re\{x[n]\}\Im\{\omega_N^{nk}\} + \Im\{x[k]\Re\{\omega_N^{nk}\}\}), \tag{6.19}$$

$$k = 0, 1, \ldots, N-1, \tag{6.20}$$

where $\Re\{\cdot\}$ is the real part and $\Im\{\cdot\}$ is the imaginary part of a complex number. This operation requires N^2 complex multiplications and $N(N-1)$ complex additions. Each complex multiplication requires four real multiplications and two real additions and each complex addition requires two real additions. Thus, the direct implementation of the DFT requires $4N^2$ real multiplications and $N(4N-2)$ real additions operations. The computational complexity of the DFT can be reduced using fast algorithms, such as the Cooley-Tukey [261], the Rader-Brenner [66], and the Winograd algorithm [65], to name but a few.

6.3.2 Approximation Theory

Besides using fast algorithms to compute the DFT, there are other alternatives that can reduce the computational complexity for the computation of the transform, such as proposing low-complexity approximations. Several discrete transforms approximations have been proposed and

shown to be efficient alternatives for computing transforms such as the DCT [71, 72, 74, 80–93], the KLT [94–99, 252, 253], the discrete Tchebichef transform [166], and the DFT [102, 103, 106, 107, 251].

In a similar way to what was proposed for the rounded approximations RDCT [72] and RKLT [253], we aim to propose an efficient method for estimating the periodogram $S(f_k)$ based on a low-complexity rounded DFT approximation. Following this approach, the obtention of multiplierless DFT approximations can be done by using round-off functions in the elements of the exact matrix. By applying the round-off functions to elements of the matrix we eliminate the multiplications by rational numbers, once the approximate matrix contains only integers elements. The round-off functions considered are defined as

$$\text{round}(x) = \lfloor \Re(x) + 0.5 \rfloor + j \cdot \lfloor \Im(x) + 0.5 \rfloor, \tag{6.21}$$

$$\text{round}_{\text{HD}}(x) = \lceil \Re(x) - 0.5 \rceil + j \cdot \lceil \Im(x) - 0.5 \rceil, \tag{6.22}$$

$$\text{round}_{\text{HAFZ}}(x) = \text{sign}(\Re(x)) \cdot \lfloor \Re(x) + 0.5 \rfloor + j \cdot \text{sign}(\Im(x)) \cdot \lfloor \Im(x) + 0.5 \rfloor, \tag{6.23}$$

$$\text{round}_{\text{HTZ}}(x) = \text{sign}(\Re(x)) \cdot \lceil \Re(x) - 0.5 \rceil + j \cdot \text{sign}(\Im(x)) \cdot \lceil \Im(x) - 0.5 \rceil, \tag{6.24}$$

with $\lfloor x \rfloor = \max\{m \in \mathbb{Z} | m \leq x\}$, $\lceil x \rceil = \min\{n \in \mathbb{Z} | n \geq x\}$, and the function sign is defined as

$$\text{sign}(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{if } x = 0, \\ -1, & \text{if } x < 0. \end{cases} \tag{6.25}$$

6.4 LOW-COMPLEXITY DFT-BASED INFERENCE

In light of the problem presented in Section 6.2.1 of detecting a simple sinusoidal tone and considering the transforms approximation theory discussed in Section 6.3.2, we aim at proposing a low-complexity detector based on approximations for the DFT.

6.4.1 Proposed detector

Thus, consider the following hypothesis:

$$\begin{cases} H_0 : x[n] = q[n] \\ H_1 : x[n] = A \sin(2\pi n f + \phi) + q[n], \end{cases} \tag{6.26}$$

where $A \in \mathbb{R}$, $f \in (0,1)$, $\phi \in (0,2\pi)$, and $q[n] \sim \mathrm{N}(0,\sigma^2)$; we can propose the following statistical test:

$$\widehat{T} = 2 \cdot \frac{\max(\widehat{S}(f_k))}{\widehat{\sigma}^2}. \tag{6.27}$$

The approximate periodogram is defined as

$$\widehat{S^*}(f_k) = \frac{1}{N \cdot (\mathrm{int}(\gamma))^2} \left(\frac{A^2}{\sigma_A^2} + \frac{B^2}{\sigma_B^2} \right), \tag{6.28}$$

where

$$A = \sum_{n=0}^{N-1} q[n] \, \mathrm{int}(\gamma\{\cos(2\pi n f_k)\}), \quad B = \sum_{n=0}^{N-1} q[n] \, \mathrm{int}(\gamma\{\sin(2\pi n f_k)\}), \tag{6.29}$$

$$\sigma_A^2 = \sigma^2 \sum_{n=0}^{N-1} (\mathrm{int}(\gamma\{\cos(2\pi n f_k)\}))^2, \quad \sigma_B^2 = \sigma^2 \sum_{n=0}^{N-1} (\mathrm{int}(\gamma\{\sin(2\pi n f_k)\}))^2, \tag{6.30}$$

with $\mathrm{int}(\cdot)$ one of the round-off functions, and γ is a defined expansion factor [13].

Corollary 6.4.0.1. *The approximate periodogram $\widehat{S^*}(f_k)$ is asymptotically distributed by a chi-square with two degrees of freedom, $\widehat{S^*}(f_k) \overset{a}{\sim} \chi^2_{(2)}$.*

Proof. Under H_0 we have that $x[n] = q[n]$, where $q[n] \sim N(0,\sigma^2)$. Let

$$\widehat{S}(f_k) = \frac{1}{N \cdot (\mathrm{int}(\gamma))^2} \left| \sum_{n=0}^{N-1} q[n] \, \mathrm{int}(\gamma \cdot \exp(-j2\pi n f_k)) \right|^2, \tag{6.31}$$

$$k = 0, 1, \ldots N/2 - 1. \tag{6.32}$$

We can rewrite $\widehat{S}(f_k)$ by

$$\widehat{S}(f_k) = \frac{1}{N \cdot (\mathrm{int}(\gamma))^2} \left| \sum_{n=0}^{N-1} q[n] \, \mathrm{int}(\gamma\{\cos(2\pi n f_k) - j\sin(2\pi n f_k)\}) \right|^2 \tag{6.33}$$

$$= \frac{1}{N \cdot (\mathrm{int}(\gamma))^2} \left[\left(\sum_{n=0}^{N-1} q[n] \, \mathrm{int}(\gamma\{\cos(2\pi n f_k)\}) \right)^2 + \left(\sum_{n=0}^{N-1} q[n] \, \mathrm{int}(\gamma\{\sin(2\pi n f_k)\}) \right)^2 \right] \tag{6.34}$$

$$= \frac{1}{N \cdot (\mathrm{int}(\gamma))^2} \left(A^2 + B^2 \right) \tag{6.35}$$

Note that A and B are normally distributed with

$$\mathrm{E}(A) = \mathrm{E} \left(\sum_{n=0}^{N-1} q[n] \, \mathrm{int}(\gamma\{\cos(2\pi n f_k)\}) \right) \tag{6.36}$$

$$= \sum_{n=0}^{N-1} \mathrm{E}(q[n]) \, \mathrm{int}(\gamma\{\cos(2\pi n f_k)\}) = 0, \tag{6.37}$$

and

$$\mathrm{E}(B) = \mathrm{E}\left(\sum_{n=0}^{N-1} q[n]\,\mathrm{int}(\gamma\{\sin(2\pi n f_k)\})\right) \tag{6.38}$$

$$= \sum_{n=0}^{N-1} \mathrm{E}(q[n])\,\mathrm{int}(\gamma\{\sin(2\pi n f_k)\}) = 0, \tag{6.39}$$

since $\mathrm{E}(q[n]) = 0$. The variances of A and B are computed respectively by

$$\mathrm{VAR}(A) = \mathrm{VAR}\left(\sum_{n=0}^{N-1} q[n]\,\mathrm{int}(\gamma\{\cos(2\pi n f_k)\})\right) \tag{6.40}$$

$$= \sum_{n=0}^{N-1} \mathrm{VAR}(q[n])(\mathrm{int}(\gamma\{\cos(2\pi n f_k)\}))^2 \tag{6.41}$$

$$= \sigma^2 \sum_{n=0}^{N-1} (\mathrm{int}(\gamma\{\cos(2\pi n f_k)\}))^2 = \sigma_A^2 \tag{6.42}$$

and

$$\mathrm{VAR}(B) = \mathrm{VAR}\left(\sum_{n=0}^{N-1} q[n]\,\mathrm{int}(\gamma\{\sin(2\pi n f_k)\})\right) \tag{6.43}$$

$$= \sum_{n=0}^{N-1} \mathrm{VAR}(q[n])(\mathrm{int}(\gamma\{\sin(2\pi n f_k)\}))^2 \tag{6.44}$$

$$= \sigma^2 \sum_{n=0}^{N-1} (\mathrm{int}(\gamma\{\sin(2\pi n f_k)\}))^2 = \sigma_B^2. \tag{6.45}$$

Thus, since the sum of two squared standard normally distributed independent random variables is a chi-square distributed random variable [259], we have that

$$\frac{1}{N\cdot(\mathrm{int}(\gamma))^2}\left(\frac{A^2}{\sigma_A^2} + \frac{B^2}{\sigma_B^2}\right) \sim \chi_2^2. \tag{6.46}$$

The computation of (6.46) requires the unknown parameter σ^2. If we consider $\widehat{\sigma}^2$ the maximum likelihood estimator (MLE) of σ^2, which is a consistent estimator, we have that $\widehat{S}(f_k)$ is asymptotically distributed by a chi-square distribution when $N \to \infty$,

$$\widehat{S}^*(f_k) = \frac{1}{N\cdot(\mathrm{int}(\gamma))^2}\left(\frac{A^2}{\sigma_A^2} + \frac{B^2}{\sigma_B^2}\right) \overset{a}{\sim} \chi_2^2. \tag{6.47}$$

□

Corollary 6.4.0.2. *The approximate statistical test $\widehat{T}$ is asymptotically distributed by a chi-square with two degrees of freedom, $\widehat{T} \overset{a}{\sim} \chi^2_{(2)}$, and has the asymptotic probability distribution function given by*

$$P(\widehat{T} < t) = \left[1 - \exp\left(-t\widehat{\sigma}^2\right)\right]^{\frac{N}{2}-1}. \tag{6.48}$$

Proof. The approximate statistical test $\widehat{T}$ is given by

$$\widehat{T} = 2 \cdot \frac{\max(\widehat{S}^*(f_k))}{\widehat{\sigma}^2}. \tag{6.49}$$

Under the null hypothesis H_0, we have that $\widehat{S}^*(f_k) \overset{a}{\sim} \chi^2_2$. Thus, the asymptotic probability density function is given by [259]

$$p_{\widehat{S}^*}(s) = \frac{1}{2}\exp\left(-\frac{s}{2}\right), \tag{6.50}$$

and the distribution function is

$$P(\widehat{S}^* < t) = \int_0^t p_{\widehat{S}^*}(s)ds \tag{6.51}$$

$$= \int_0^t \frac{1}{2}\exp\left(-\frac{s}{2}\right)ds = 1 - \exp\left(-\frac{t}{2}\right). \tag{6.52}$$

Therefore, the probability distribution function of $\widehat{T}$ is given by

$$P(\widehat{T} < t) = P\left(2 \cdot \frac{\max(\widehat{S}^*(f_k))}{\widehat{\sigma}^2} < t\right) \tag{6.53}$$

$$= P\left(\max(\widehat{S}^*(f_k)) < \frac{t\widehat{\sigma}^2}{2}\right), \tag{6.54}$$

for $\widehat{\sigma}^2$ the MLE of σ^2 and $k = 0, 1, \ldots, N/2-1$,

$$P(\widehat{T} < t) = P\left(\max(\widehat{S}^*(f_k)) < \frac{t\widehat{\sigma}^2}{2}\right) \tag{6.55}$$

$$= P\left((\widehat{S}^*(f_0), \widehat{S}^*(f_1), \ldots, \widehat{S}^*(f_{N/2-1})) < \frac{t\widehat{\sigma}^2}{2}\right) \tag{6.56}$$

$$= \prod_{k=0}^{N/2-1} P\left(\widehat{S}^*(f_k) < \frac{t\widehat{\sigma}^2}{2}\right) \tag{6.57}$$

$$= P\left(\widehat{S}^*(f_k) < \frac{t\widehat{\sigma}^2}{2}\right)^{\frac{N}{2}-1} \tag{6.58}$$

$$= \left[1 - \exp\left(-t\widehat{\sigma}^2\right)\right]^{\frac{N}{2}-1}. \tag{6.59}$$

□

6.5 NUMERICAL EXPERIMENTS

In order to assess the hypothesis test, we carry out the following experiment. Let $x[n]$ be

$$\begin{cases} H_0 : x[n] = q[n] \\ H_1 : x[n] = A\sin(2\pi nf + \phi) + q[n], \end{cases} \tag{6.60}$$

where $A \in \mathbb{R}$, $f \in (0,1)$, $\phi \in (0,2\pi)$, and $q[n] \sim \mathrm{N}(0,\sigma^2)$. For instance, we generated $N = 256$ random observations of $\mathbf{x}$ with $A = 2$, $f = 0.01$, $\phi = \pi$, $q[n] \sim \mathrm{N}(0,\sigma^2)$, and $\sigma^2 = 1$. In Figure 41, we present one signal realization under the null hypothesis, i.e., only the white noise, and under H_1.

In order to perform the hypothesis test, we need to estimate the periodogram, once the statistical test is based on it. Figure 42 presents the exact periodogram $S(f_k)$ of the signals generated under H_0 and H_1. For this case, considering the signal under the null hypothesis, i.e, the signal is only the white noise, the statistical test is

$$T = 2 \cdot \frac{\max(S(f_k))}{\widehat{\sigma}^2} = 7.6325, \tag{6.61}$$

and the threshold based on the quantile of the chi-square distribution is $\chi^2_{(\tau,2)} = 15.6291$, with τ defined in Equation (6.7) with $\alpha = 0.05$. Here, $T < \chi^2_{(\tau,2)}$, what gives us no evidence to reject H_0, that is the signal is only the white noise. Still, considering $x[n]$ under the alternative hypothesis, we have $T = 86.9917$. Since $T > \chi^2_{(\tau,2)}$, we can now reject H_0 and assume the presence of a sinusoidal signal. By this simple experiment realization, we can note that the test has a correct performance.

6.5.1 Approximate test evaluation

Simulation studies can be used to evaluate the performance of the test in terms of the probabilities of the type I and type II errors. Considering this particular case where we want to detect a simple sinusoidal tone, we proceeded with the following approach to compute the statistical test, $\widehat{T}$.

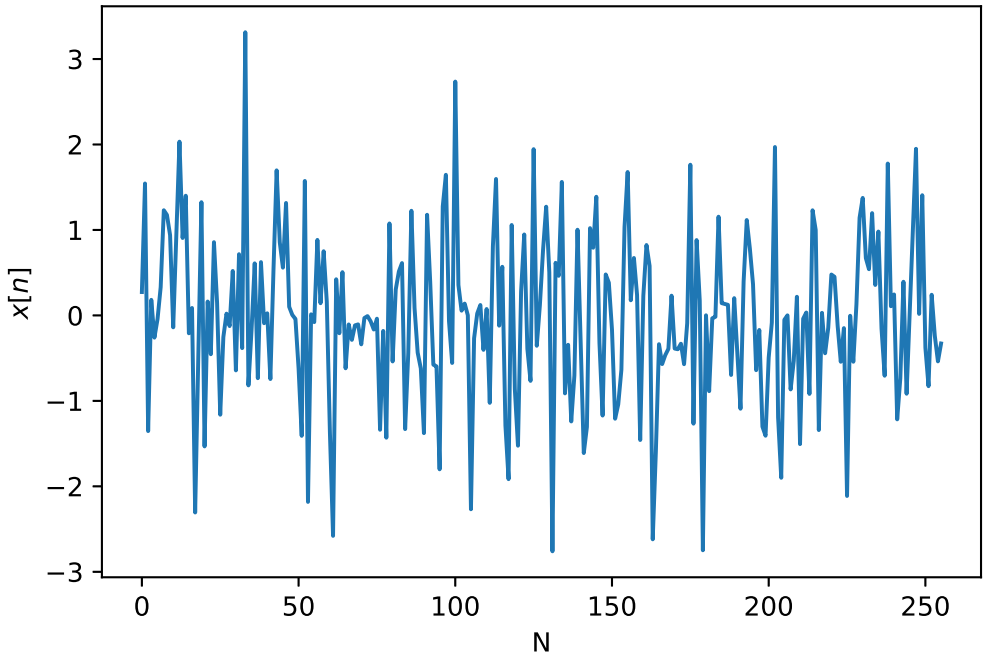
First, we evaluate the test in terms of the type I error. The pseudocode for computing the type I error is presented in Algorithm 4.

This evaluation was implemented considering the following values: $N = 8$, 16, 32, 64, 128, and 256; $\alpha = 0.05$; $R = 1000$; $\sigma^2 = 1$; the round-off functions ($\mathrm{int}(\cdot)$) presented in (6.21); and the expansion factor $0 < \gamma < 3$ with steps of 10^{-2}. For each length N we evaluated $\widehat{\alpha}$ in order to be closest to $\alpha = 0.05$, that is,

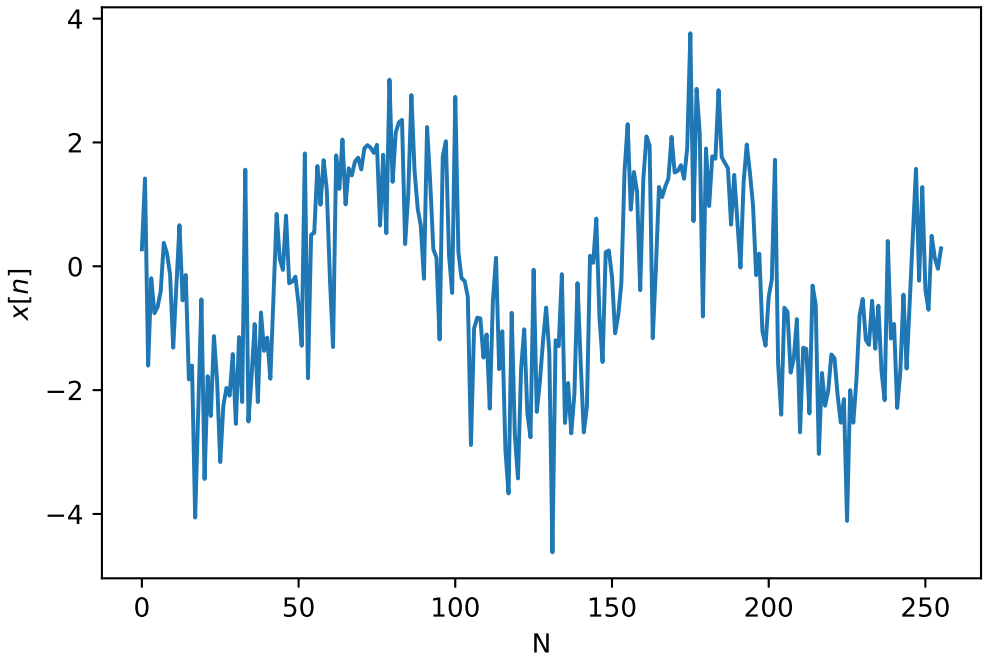
$$\widehat{T}^* = \arg\min_{\gamma,\mathrm{int}} \mathrm{error}(\widehat{\alpha} - \alpha). \tag{6.62}$$

Table 22 presents the best results for each length N. The round-off function that had the best performance for all values of N was the $\mathrm{round}(\cdot)$ function. We can note that as far as N increases, the probability of type I error of the approximate test becomes closer to the expected value,

Figure 41 – Generated signals.
(a) Generated signal under H_0



(b) Generated signal under H_1

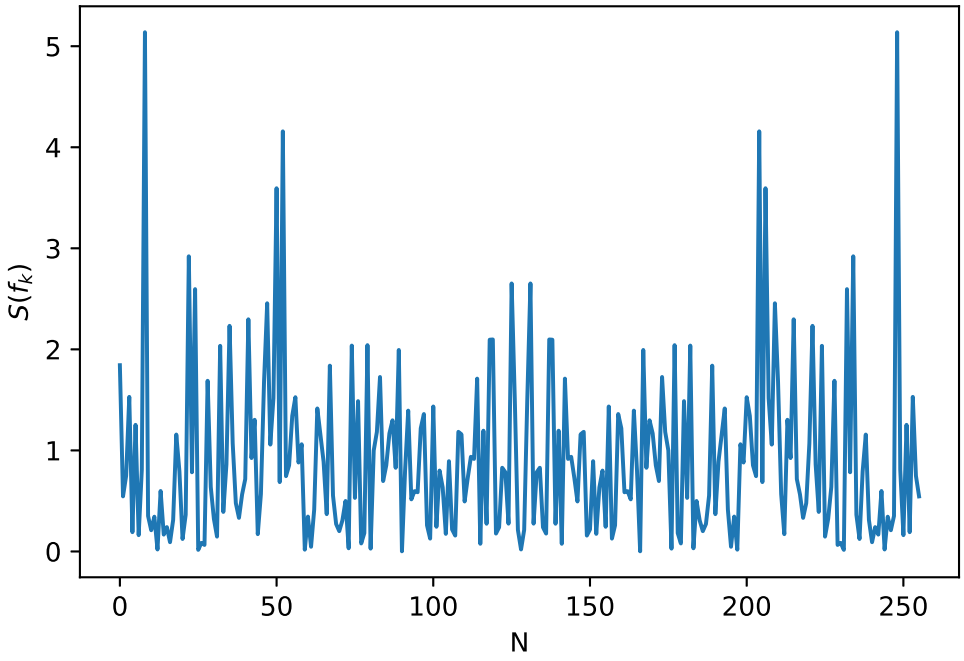


Source: Author (2023).

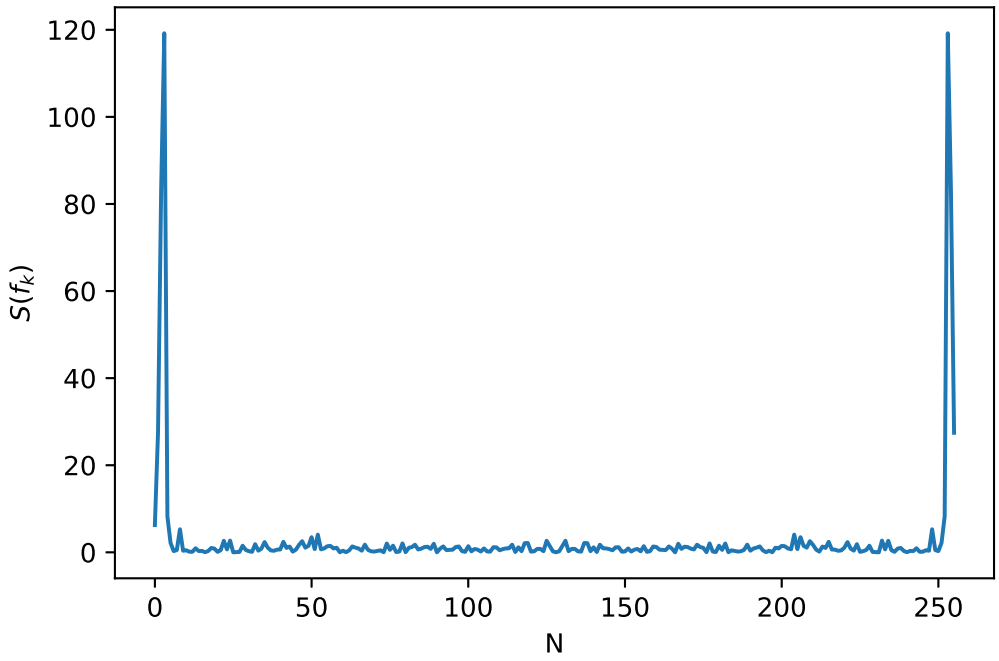
$\alpha = 0.05$. Thus, considering these results, the proposed approximate test is computed as

$$\widehat{T} = 2 \cdot \frac{\max(\widehat{S}^*(f_k))}{\widehat{\sigma}^2}, \tag{6.63}$$

Figure 42 – Periodogram of the generated signals.
(a) Periodogram of the signal under H_0



(b) Periodogram of the signal under H_1



Source: Author (2023).

where

$$\widehat{S}^{*}(f_k) = \frac{1}{4N}\left|\sum_{n=0}^{N-1} x[n]\,\text{round}(\gamma \cdot \exp(-j2\pi n f_k))\right|^2, \tag{6.64}$$

Algorithm 4: Pseudocode for computing the type I error.

Input: N, α, R, $\chi^2_{(\tau,2)}$, σ^2,int$(\cdot)$, γ
Output: Type I error ($\widehat{\alpha}$)
  for j in $(0:R)$ **do**
    Generate $x[n]$ under H_0;
    Compute $S(f_k)$ and then T;
    if $T > \chi^2_{(\tau,2)}$ **then**
      $i \leftarrow i+1$;
    end if
  end for
  return $\widehat{\alpha} = i/R$

Table 22 – Values of γ and $\widehat{\alpha}$ from the best approximate test for each lenght N

N	γ	$\widehat{\alpha}$
8	1.68	0.120
16	1.65	0.064
32	1.79	0.050
64	1.74	0.049
128	1.85	0.051
256	1.82	0.051

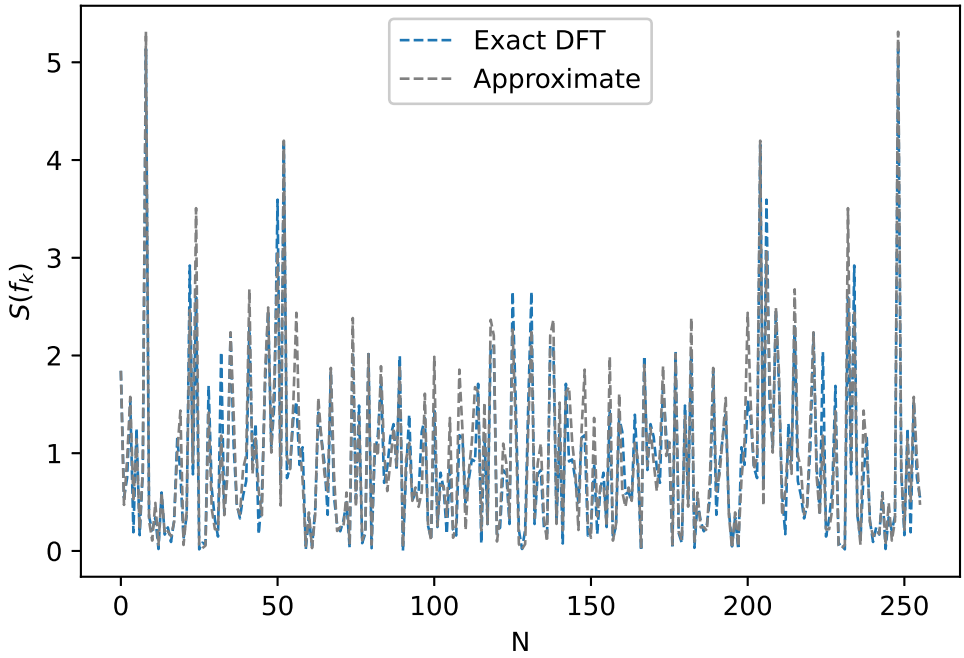
Source: Author (2023).

with the values of γ presented in Table 22 for each length N.

For instance, considering one realization of a signal with $N = 256$, $f = 0.01$, $A = 2$, $\phi = \pi$, and $\sigma^2 = 1$, we have the following results. Figures 43 and 44 present the approximate and exact periodograms for the signal considering both scenarios, under the null hypothesis H_0 and under the alternative hypothesis H_1. It is worth to emphasize here that the periodogram estimated with the approximate DFT is very similar to the exact one, at a lower cost.

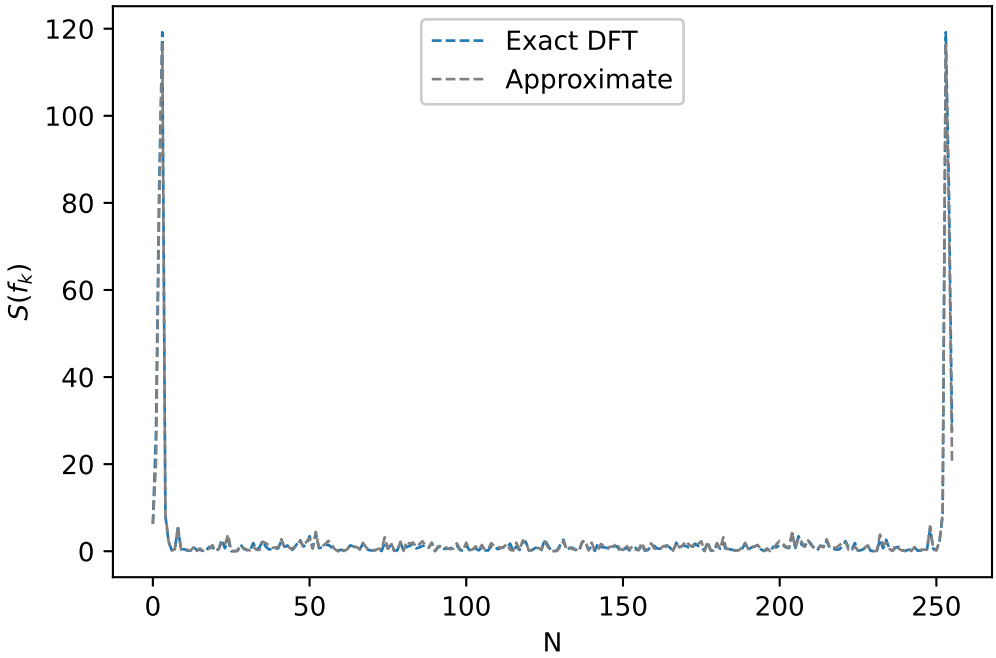
Besides computing the probability of the type I error, we can also compute the probability of the type II error and the power of the test. The procedure for computing numerically the probability is presented in Algorithm 5. Figures 45, 46, 47, and 48 present the power of the test for the statistics computed using the exact and approximate DFT, considering $N = 8$, 16, 32, 64, 128, and 256, and different values of the frequency $f = 0.01$,0.10, 0.25, and 0.375. We can note that for larger values of N ($N > 32$) when the signal amplitude (A) is close to zero, the power of the tests tend to $\alpha = 0.05$ as expected, since under H_0 we have that $A = 0$. We can also emphasize that the proposed low-complexity test has a performance very similar to the exact one, requiring a lower implementation cost. The arithmetic cost of the proposed test approximation is detailed next.

Figure 43 – Exact and approximate periodogram ($S(f_k)$) under H_0.



Source: Author (2023).

Figure 44 – Exact and approximate periodogram ($S(f_k)$) under H_1.

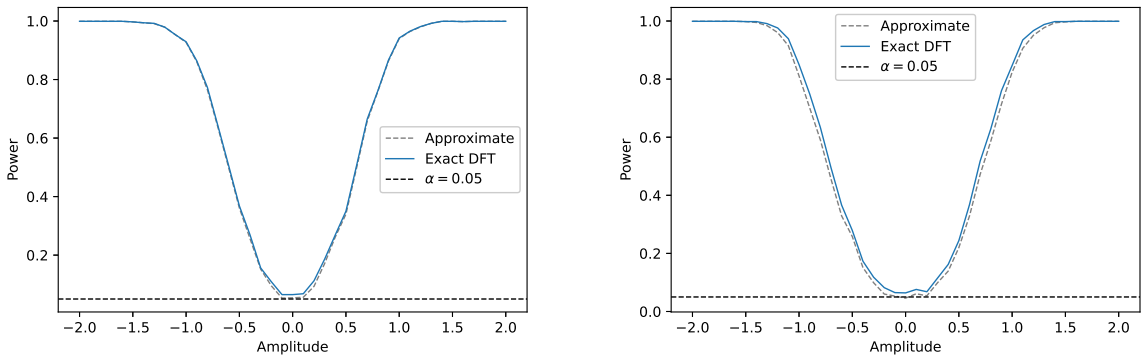


Source: Author (2023).

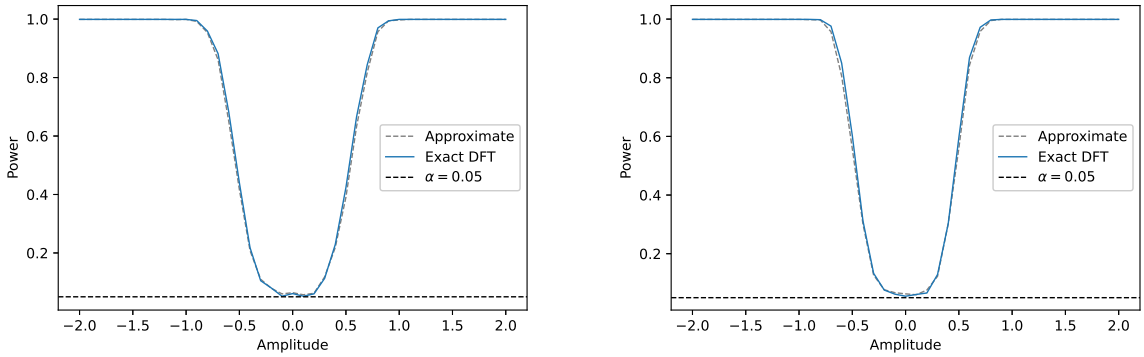
Algorithm 5: Pseudocode for computing the power of the test.

Input: N, α, R, $\chi^2_{(\tau,2)}$
Output: Power of the test $(1-\beta)$
  for j in $(0:R)$ **do**
    Generate $\mathbf{x}[n]$ under H_1;
    Compute $S(f_k)$ and then T;
    if $T > \chi^2_{(\tau,2)}$ **then**
      $i \leftarrow i+1$;
    end if
  end for
  return i/R

Figure 45 – Power of the test for different lengths of the signal for $f = 0.01$.
(a) $N = 32$ **(b)** $N = 64$



(c) $N = 128$ **(d)** $N = 256$

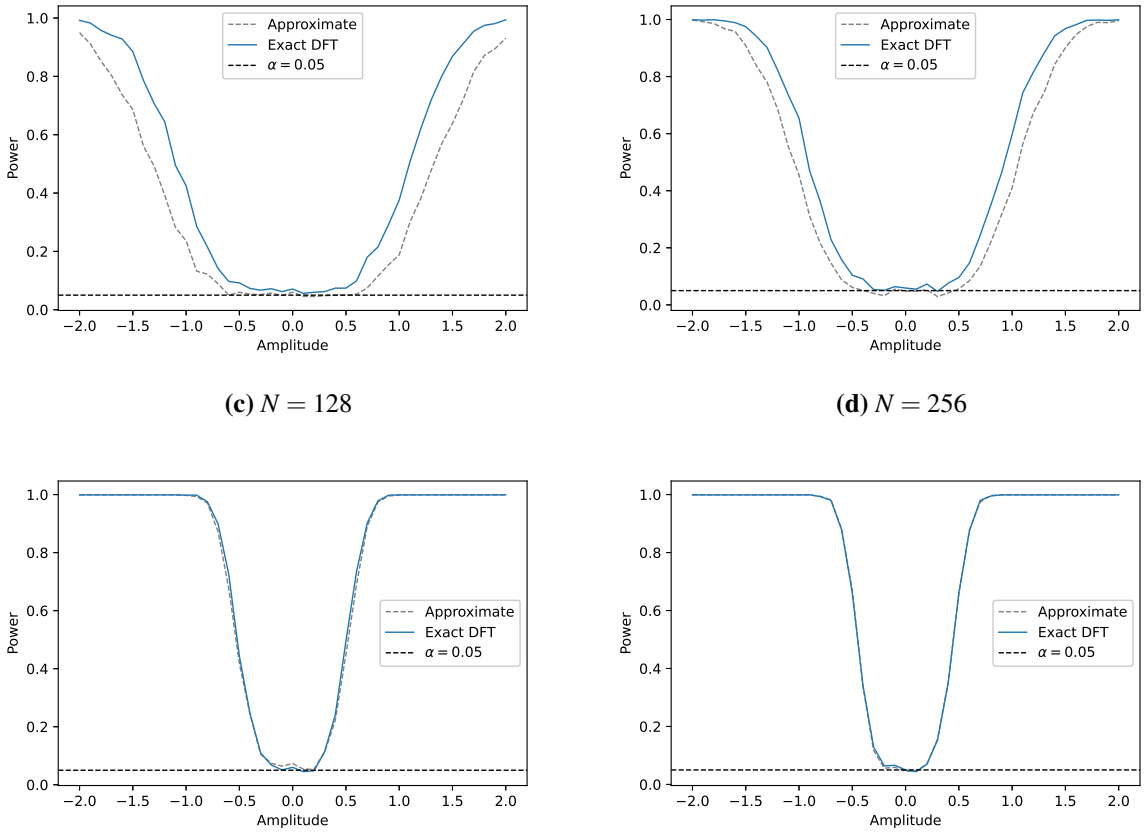


Source: Author (2023).

6.5.2 Fast algorithms

The proposed DFT approximations for computing the statistical test were derived in order to be low-complexity transforms. Given the fact that the approximate transforms are multiplierless, they have already a gain when compared to the exact DFT or FFTs. Table 23 presents the arithmetic complexity of the DFT computed by the Radix-2 algorithm [262] and the

Figure 46 – Power of the test for different lengths of the signal for $f = 0.10$.
(a) $N = 32$ **(b)** $N = 64$



(c) $N = 128$ **(d)** $N = 256$

Source: Author (2023).

proposed approximations. Notice that the proposed approximations only require real additions. Still, we can reduce the arithmetic cost of the proposed transforms by factorizing them into

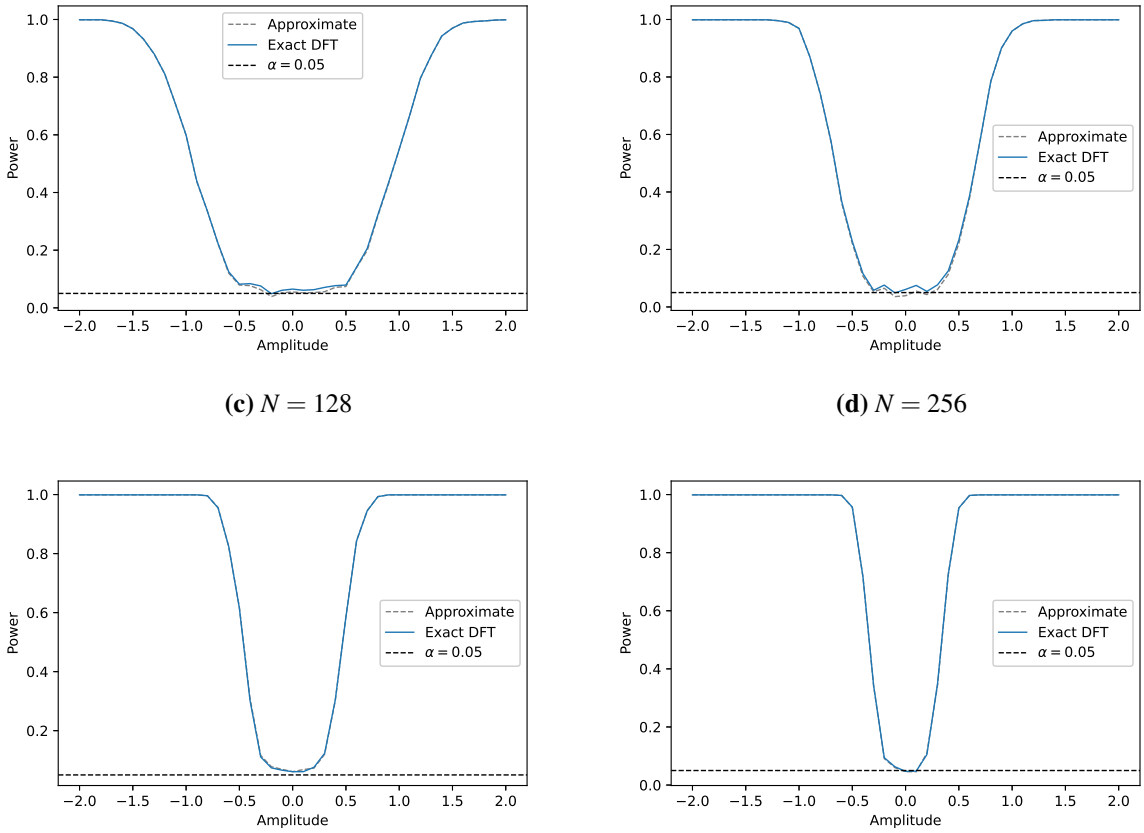
Table 23 – Arithmetic complexity of the DFT computed by the Radix-2 algorithm the and proposed transforms

	FFT (Radix-2 Algorithm) [262]		Approximate DFT	
N	Complex Multiplications	Complex Adds	Multiplications	Adds
8	12	24	0	66
16	32	64	0	354
32	80	160	0	1538
64	192	384	0	6530
128	448	896	0	26626
256	1024	2048	0	107010

Source: Author (2023).

sparce matrices [23]. We derived the fast algorithms for the approximate DFTs for $N = 8$, 16, 32, and 64. The fast algorithms are presented in Appendix D. The arithmetic complexity of the proposed approximations with and without the factorization are presented in Table 24.

Figure 47 – Power of the test for different lengths of the signal for $f = 0.25$.
(a) $N = 32$ **(b)** $N = 64$



(c) $N = 128$ **(d)** $N = 256$

Source: Author (2023).

Table 24 – Arithmetic complexity of the proposed transforms before and after the factorization

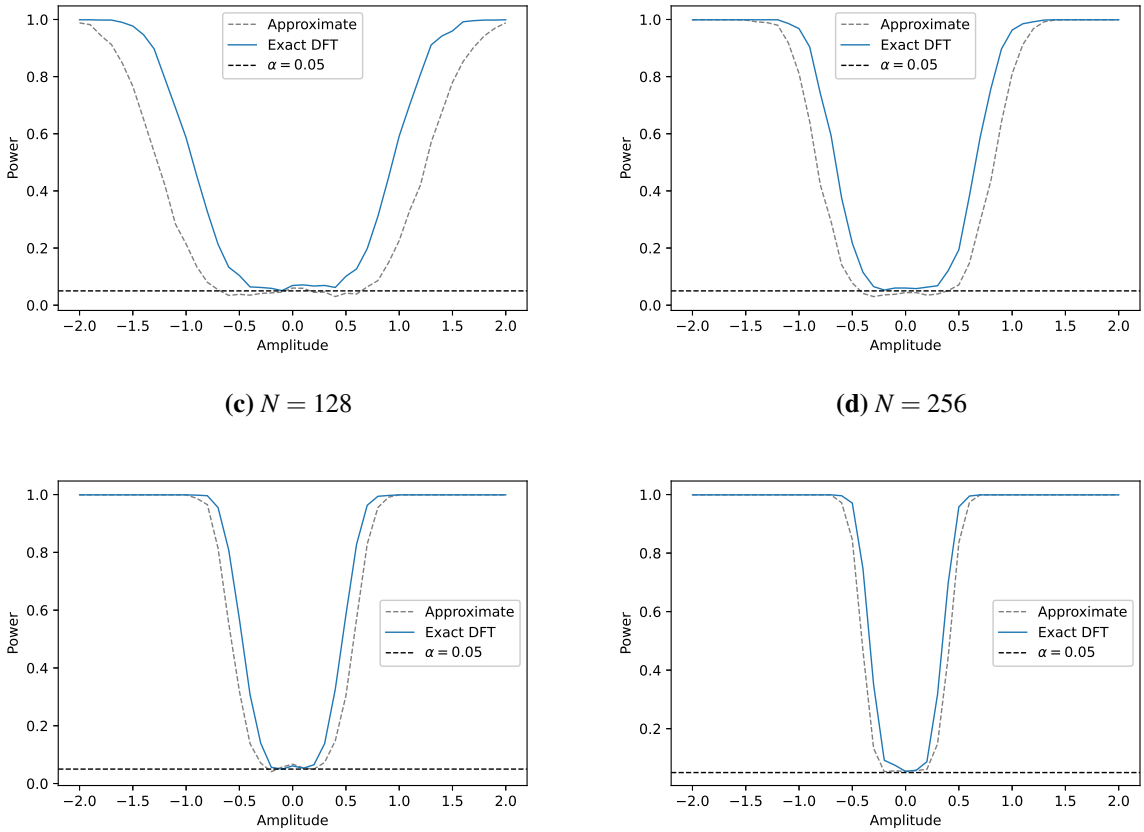
	Before factorization		After factorization		Reduction (%)	
N	Adds	Bit-shifts	Adds	Bit-shifts	Adds	Bit-shifts
8	66	48	20	6	69.7	87.5
16	354	192	58	12	83.6	93.7
32	1538	704	152	26	90.1	96.3
64	6530	2816	422	88	93.5	96.8

Source: Author (2023).

6.6 CONCLUSIONS

In this chapter, we present a review of the problem of detection theory in the context of statistical inference. The signal detection can be related to a hypothesis test and the statistical test can be defined through the DFT of the sequence of the signal. We proposed an approximate test based on low-complexity approximations for the DFT. The proposed approximations are multiplierless and require fewer additions and bit-shift operations than the direct implementation of the DFT. Considering the problem of detecting a simple sinusoidal tone, we have shown with

Figure 48 – Power of the test for different lengths of the signal for $f = 0.375$.
(a) $N = 32$ **(b)** $N = 64$



(c) $N = 128$ **(d)** $N = 256$

Source: Author (2023).

computational experiments that the proposed low-complexity test performs similarly to the exact test, at a lower cost.

7 CONCLUSIONS

In this final chapter, we present the concluding remarks, contributions, and future works of the thesis.

7.1 CONCLUDING REMARKS

In this thesis, we presented the relevance of discrete transforms in the context of signal processing systems. Emphasis was given to the Karhunen-Loève transform, the discrete cosine transform, and the discrete Fourier transform, as well as how approximations of these transforms can be useful in contexts where there is a severe shortage of energy resources. In each chapter, we introduced new classes of approximations for the transforms, along with fast algorithms and applications where these approximations can be considered.

In Chapter 2, we presented a new class of KLT approximations based on the signum function, the signed KLT (SKLT). The proposed approximate transforms are data-independent which differs from the fast approximations for the KLT that were already known in the literature. Total figures of merit were proposed for the selection of optimal transforms. The 4-, 8-, 16-, and 32-point transforms were submitted to a comprehensive assessment in the context of image and video coding. We also derived fast algorithms for the proposed transforms which were capable of reducing the arithmetic cost of their implementation. The experiments on image and video coding demonstrated the suitability of the proposed approximations, generating high-quality images according to coding and similarity metrics.

In Chapter 3, analogously to the approach used for deriving the SKLT, we presented a new class of low-complexity transforms that are obtained through the application of the rounding-off function to the elements of the KLT matrix, the rounded KLT (RKLT). The proposed transforms were evaluated considering figures of merit that measure the coding power and distance of the proposed approximations to the exact KLT and were also explored in image compression experiments. Fast algorithms were introduced for the proposed approximate transforms. Due to its relevance in practical image coding systems, the special case $N = 8$ was comprehensively examined, but the methodology presented in that chapter is capable of finding an approximation for any blocklength N. It was shown that the proposed transforms perform well in image compression and require a low implementation cost.

In Chapter 4, a design methodology was introduced to obtain optimal KLT approximations. The collection of the proposed new approximations was generated based on integer

functions and an optimization problem was solved aiming at the proposition of optimal transforms according to defined figures of merit. Fast algorithms were also derived for the optimal approximations proposed by factorizing the transforms into sparse matrices. The applicability of the proposed approximation in the context of image compression was demonstrated, and an FPGA hardware implementation was provided, showing the trade-off between performance and resource usage.

In Chapter 5, we presented new approximations for larger blocklengths for the DCT based on minimal angle similarity. We considered an approach already known in the literature used for proposing 8-point DCT approximations to propose 16-, 32-, and 64-point approximations. We showed that the proposed approximations outperformed approximations for the DCT already known in the literature according to some figures of merit. We also derived fast algorithms for the low-complexity transforms. The proposed approximations were assessed for image compression experiments, where their applicability and performance were highly satisfactory.

In Chapter 6, we reviewed the signal detection problem as in the context of a statistical hypothesis test. We also presented a low-complexity alternative for computing the statistical test based on an approximation of the DFT. We have shown that the proposed approximation test performs very similarly to the statistical test considering the exact DFT when assessed under the power and size of the test (type I and type II errors).

7.2 CONTRIBUTIONS

Some of the contributions of the thesis are already published in the literature. The work from Chapters 2, 3, and 5 are published in international journals. The paper from Chapter 3, "Low-complexity rounded KLT approximation for image compression" has 3 citations. The paper from Chapter 4 is submitted and under review, and the manuscript from Chapter 6 is in preparation for publication.

The contributions of the thesis are listed below:

1. Chapter 2: RADÜNZ, A. et al. Data-independent low-complexity KLT approximations for image and video coding. Signal Processing: Image Communication, 2021, https://doi.org/10.1016/j.image.2021.116585
2. Chapter 3: RADÜNZ, A. P.; BAYER, F. M.; CINTRA, R. J. Low-complexity rounded KLT approximation for image compression. Journal of Real-Time Image Processing, p. 1–11, 2021, https://doi.org/10.1007/s11554-021-01173-0.

3. Chapter 4: RADÜNZ, A. P.; COELHO, D. F. G.; BAYER, F. M.; CINTRA, R. J. Fast Data-independent KLT Approximations Based on Integer Functions submitted to Multimedia Tools and Applications, under review.
4. Chapter 5: RADÜNZ, A. P.; PORTELLA, L.; OLIVEIRA, R. S.; BAYER, F. M.; CINTRA, R. J. Extensions on low-complexity DCT approximations for larger blocklengths based on minimal angle similarity. Journal of Signal Processing Systems, 2023, https://doi.org/10.1007/s11265-023-01848-w
5. Chapter 6: Manuscript in development: RADÜNZ, A. P.; BAYER, F. M.; CINTRA, R. J. Low-complexity methods for statistical signal processing.

7.3 FUTURE WORK

As a suggestion for future work based on this work, the following lines of research stand out: to find new applications for the approximate discrete transforms, to expand the approximation approach to the multidimensional case, and to derive approximations of larger blocklengths. The suggestions for future work research are detailed next:

1. Studying other applications for the KLT approximations: we have shown the applicability of the KLT when assessed to image and video compression, but there is still room for improvement in scenarios where we have signals with low correlation.
2. Proposing new approximations for larger blocklengths: with the advancement of image and video encoding technology, there is a demand for larger blocklengths that could be employed in modern codecs. For instance, there is a new video standard coding, Versatile Video Coding (VVC), that requires 64-point transforms.
3. Expanding the approximation approach to the multidimensional case: the multidimensional transform can be applied for filtering multispectral images, for example. This type of image can be seen as a three-dimensional cuboid, in which the third dimension is the wavelength. It is widely used and with great current importance in geoscience and remote sensing contexts.
4. Studying other contexts where the DFT is used as a tool: the bispectrum estimation. The bispectrum is the discrete Fourier transform of the third-order cumulant sequence and is a very useful tool in the context of signal processing. Since the bispectrum estimation depends on the DFT computation, low-complexity approximations can be a good alternative in contexts of lower processing power or severe restrictions of energy autonomy.

5. Applying the developed methods in the most different real applications in signal and image processing.

REFERENCES

1 WU, X. *et al.* Data mining with big data. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 1, p. 97–107, 2014.

2 OUSSOUS, A. *et al.* Big data technologies: A survey. **Journal of King Saud University - Computer and Information Sciences**, v. 30, n. 4, p. 431–448, 2018. ISSN 1319-1578.

3 GOLDSTON, D. Big data: Data wrangling. **Nature News**, Nature Publishing Group, v. 455, n. 7209, p. 15, 2008.

4 MERVIS, J. Agencies rally to tackle big data. **Science**, American Association for the Advancement of Science, v. 336, n. 6077, p. 22, 2012. ISSN 0036-8075.

5 MUKHOPADHYAY, S. C.; SURYADEVARA, N. K. **Internet of things: Challenges and opportunities**. Switzerland: Springer, 2014.

6 WEBER, R. H.; WEBER, R. **Internet of things**. Heidelberg: Springer, 2010. v. 12.

7 HUNT, E. B. **Artificial intelligence**. New York, NY: Academic Press, 2014.

8 ZHANG, C.; LU, Y. Study on artificial intelligence: The state of the art and future prospects. **Journal of Industrial Information Integration**, Elsevier, v. 23, p. 100224, 2021.

9 ALLAM, Z.; DHUNNY, Z. A. On big data, artificial intelligence and smart cities. **Cities**, v. 89, p. 80–91, 2019. ISSN 0264-2751.

10 MOSTELLER, F. On some useful "inefficient" statistics. In: **Breakthroughs in Statistics**. New York, NY: Springer, 1992. p. 209–228.

11 STEIN, J. Y. **Digital signal processing: a computer science perspective**. New York, NY: Wiley Online Library, 2000.

12 GONZALEZ, R. C.; WOODS, R. E. *et al.* **Digital image processing**. Upper Saddle River, NJ: Prentice hall, 2002.

13 BRITANAK, V.; YIP, P. C.; RAO, K. R. **Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations**. Oxford, UK: Academic Press, 2010.

14 OCHOA-DOMINGUEZ, H.; RAO, K. R. **Discrete Cosine Transform**. Boca Raton, FL: CRC Press, 2019.

15 HAHN, S.; POULARIKAS, A. Transforms and applications handbook. In: **Hilbert Transforms**. Boca Raton: CRC Press, 2010.

16 SALOMON, D. **Data compression: the complete reference**. New York, NY: Springer Science & Business Media, 2004.

17 BRIGGS, W. L.; HENSON, V. E. **The DFT: an owner's manual for the discrete Fourier transform**. Philadelphia, PA: SIAM, 1995. v. 45.

18 BRACEWELL, R. N. Discrete Hartley transform. **Journal of the Optical Society of America**, Optica Publishing Group, v. 73, n. 12, p. 1832–1835, 1983.

19 AHMED, N. *et al.* Walsh-Hadamard transform. **Orthogonal transforms for digital signal processing**, Springer, p. 99–152, 1975.

20 MUKUNDAN, R.; ONG, S.; LEE, P. A. Image analysis by Tchebichef moments. **IEEE Transactions on Image Processing**, v. 10, n. 9, p. 1357–1364, 2001.

21 KARHUNEN, K. Under lineare methoden in der wahr scheinlichkeitsrechnung. **Annales Academiae Scientiarun Fennicae Series A1: Mathematia Physica**, v. 47, 1947.

22 LOÈVE, M. Fonctions aléatoires de second ordre. **Processus Stochastique et Mouvement Brownien**, Gauthier-Villars, p. 366–420, 1948.

23 BLAHUT, R. E. **Fast algorithms for signal processing**. Cambridge, UK: Cambridge University Press, 2010.

24 RAO, K. R.; YIP, P. C. **The transform and data compression handbook**. Boca Raton, FL: CRC Press, 2000. v. 1.

25 JAIN, A. K. A fast Karhunen-Loève transform for a class of random processes. **IEEE Transactions on Communications**, v. 24, n. 9, p. 1023–1029, 1976.

26 WALLACE, G. K. The JPEG still picture compression standard. **IEEE Transactions on Consumer Electronics**, v. 38, n. 1, p. xviii–xxxiv, 1992.

27 PURI, A. Video coding using the H.264/MPEG-4 AVC compression standard. **Signal Processing: Image Communication**, v. 19, 2004. ISSN 09235965.

28 POURAZAD, M. T. *et al.* HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC? **IEEE Consumer Electronics Magazine**, v. 1, n. 3, p. 36–46, 2012.

29 STRANG, G. Wavelets. **American Scientist**, v. 82, n. 3, p. 250–255, 1994.

30 JERRI, A. J. **Linear difference equations with discrete transform methods**. New York, NY: Springer Science & Business Media, 2013. v. 363.

31 BEAUDOIN, N.; BEAUCHEMIN, S. S. An accurate discrete Fourier transform for image processing. In: IEEE. **2002 International Conference on Pattern Recognition**. Quebec City, QC, Canada, 2002. v. 3, p. 935–939.

32 HAN, Y. *et al.* DFT-based hybrid beamforming multiuser systems: Rate analysis and beam selection. **IEEE Journal of Selected Topics in Signal Processing**, v. 12, n. 3, p. 514–528, 2018.

33 MADANAYAKE, A. *et al.* Fast radix-32 approximate DFTs for 1024-beam digital RF beamforming. **IEEE Access**, v. 8, p. 96613–96627, 2020.

34 BUJAKOVIĆ, D. *et al.* Analysis of human radar echo signal using warped discrete Fourier transform. In: IEEE. **2016 New Trends in Signal Processing (NTSP)**. Slovakia, 2016. p. 1–6.

35 SOHN, J.; KIM, N. S.; SUNG, W. A statistical model-based voice activity detection. **IEEE Signal Processing Letters**, v. 6, n. 1, p. 1–3, 1999.

36 QUINN, B. G. Estimation of frequency, amplitude, and phase from the DFT of a time series. **IEEE Transactions on Signal Processing**, v. 45, n. 3, p. 814–817, 1997.

37 ZAKHAROV, Y.; BARONKIN, V.; TOZER, T. DFT-based frequency estimators with narrow acquisition range. **IEE Proceedings-Communications**, IET, v. 148, n. 1, p. 1–7, 2001.

38 AHMED, N.; RAO, K. **Orthogonal Transforms for Digital Signal Processing**. Berlin Heidelberg: Springer-Verlag, 1975.

39 KAY, S. Detection of a sinusoid in white noise by autoregressive spectrum analysis. In: IEEE. **ICASSP'80. IEEE International Conference on Acoustics, Speech, and Signal Processing**. Denver, CO, USA, 1980. v. 5, p. 658–661.

40 SO, H. *et al.* Comparison of various periodograms for sinusoid detection and frequency estimation. **IEEE Transactions on Aerospace and Electronic Systems**, v. 35, n. 3, p. 945–952, 1999.

41 YANYUN, Q. *et al.* Updating algorithm for extracting the basis of Karhunen-Loève transform in nonzero mean data. In: IEEE. **Proceedings 7th International Conference on Signal Processing, 2004. Proceedings. ICSP'04. 2004.** Beijing, China, 2004. v. 2, p. 1403–1406.

42 SOLE, J. *et al.* Joint sparsity-based optimization of a set of orthonormal 2-D separable block transforms. In: IEEE. **2009 16th IEEE International Conference on Image Processing (ICIP)**. Cairo, Egypt, 2009. p. 9–12.

43 BLANES, I. *et al.* Divide-and-conquer strategies for hyperspectral image processing: A review of their benefits and advantages. **IEEE Signal Processing Magazine**, v. 29, n. 3, p. 71–81, 2012.

44 HAO, P.; SHI, Q. Reversible integer KLT for progressive-to-lossless compression of multiple component images. In: IEEE. **Proceedings 2003 International Conference on Image Processing**. Barcelona, Spain, 2003. v. 1, p. I–633.

45 LOEFFLER, C.; LIGTENBERG, A.; MOSCHYTZ, G. S. Practical fast 1-D DCT algorithms with 11 multiplications. In: IEEE. **Acoustics, Speech, and Signal Processing, 1989. ICASSP-89, 1989 International Conference on**. Glasgow, UK, 1989. p. 988–991.

46 LIANG, J.; TRAN, T. D. Fast multiplierless approximation of the DCT with the lifting scheme. **IEEE Transactions on Signal Processing**, v. 49, p. 3032–3044, December 2001.

47 MASERA, M.; MARTINA, M.; MASERA, G. Odd type DCT/DST for video coding: Relationships and low-complexity implementations. In: IEEE. **2017 IEEE International Workshop on Signal Processing Systems (SiPS)**. Lorient, France, 2017. p. 1–6.

48 SUEHIRO, N.; HATORI, M. Fast algorithms for the DFT and other sinusoidal transforms. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 34, n. 3, p. 642–644, 1986.

49 YUAN, W.; HAO, P.; XU, C. Matrix factorization for fast DCT algorithms. In: IEEE. **ICASSP 2006 Proceedings. IEEE International Conference on Acoustics, Speech and Signal Processing, 2006.** Toulouse, France, 2006. v. 3, p. III.

50 WANG, Z. Combined DCT and companding for PAPR reduction in OFDM signals. **Journal of Signal and Information Processing**, Scientific Research Publishing, v. 2, n. 2, p. 100–104, 2011.

51 LEE, B. A new algorithm to compute the discrete cosine transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 32, n. 6, p. 1243–1245, 1984.

52 SNIGDHA, F. S. *et al.* Optimal design of JPEG hardware under the approximate computing paradigm. In: ACM. **Proceedings of the 53rd Annual Design Automation Conference**. Austin, Texas, 2016. p. 106.

53 SUZUKI, T.; IKEHARA, M. Integer DCT based on direct-lifting of DCT-IDCT for lossless-to-lossy image coding. **IEEE Transactions on Image Processing**, v. 19, n. 11, p. 2958–2965, 2010.

54 VETTERLI, M.; NUSSBAUMER, H. J. *et al.* Simple FFT and DCT algorithms with reduced number of operations. **Signal Processing**, Elsevier, v. 6, n. 4, p. 267–278, 1984.

55 HOU, H. A fast recursive algorithm for computing the discrete cosine transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 35, n. 10, p. 1455–1461, 1987.

56 FONG, C.-K.; CHAM, W.-K. LLM integer cosine transform and its fast algorithm. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 22, n. 6, p. 844–854, 2012.

57 CHOI, K.; LEE, S.; JANG, E. S. Zero coefficient-aware IDCT algorithm for fast video decoding. **IEEE Transactions on Consumer Electronics**, v. 56, n. 3, 2010.

58 WANG, Z. Fast algorithms for the discrete w transform and for the discrete Fourier transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 32, n. 4, p. 803–816, 1984.

59 ARAI, Y.; AGUI, T.; NAKAJIMA, M. A fast DCT-SQ scheme for images. **IEICE TRANSACTIONS (1976-1990)**, The Institute of Electronics, Information and Communication Engineers, v. 71, n. 11, p. 1095–1097, 1988.

60 CHEN, W.-H.; SMITH, C.; FRALICK, S. A fast computational algorithm for the discrete cosine transform. **IEEE Transactions on Communications**, v. 25, n. 9, p. 1004–1009, 1977.

61 PARK, J.-S. *et al.* 2-D large inverse transform (16× 16, 32× 32) for HEVC (High Efficiency Video Coding). **JSTS: Journal of Semiconductor Technology and Science**, v. 12, n. 2, p. 203–211, 2012.

62 FEIG, E.; WINOGRAD, S. Fast algorithms for the discrete cosine transform. **IEEE Transactions on Signal processing**, v. 40, n. 9, p. 2174–2193, 1992.

63 XU, X. *et al.* Multiple morphological component analysis based decomposition for remote sensing image classification. **IEEE Transactions on Geoscience and Remote Sensing**, v. 54, n. 5, p. 3083–3102, 2016.

64 HEIDEMAN, M. T.; JOHNSON, D. H.; BURRUS, C. S. Gauss and the history of the fast Fourier transform. **Archive for History of Exact Sciences**, p. 265–277, 1985.

65 WINOGRAD, S. On computing the discrete Fourier transform. **Mathematics of Computation**, v. 32, n. 141, p. 175–199, 1978.

66 RADER, C.; BRENNER, N. A new principle for fast Fourier transformation. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 24, n. 3, p. 264–266, 1976.

67 RADER, C. M. Discrete Fourier transforms when the number of data samples is prime. **Proceedings of the IEEE**, v. 56, n. 6, p. 1107–1108, 1968.

68 FRIGO, M.; JOHNSON, S. G. FFTW: An adaptive software architecture for the FFT. In: IEEE. **Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)**. Seattle, WA, 1998. v. 3, p. 1381–1384.

69 GOOD, I. J. The interaction algorithm and practical Fourier analysis. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 20, n. 2, p. 361–372, 1958.

70 KOLBA, D.; PARKS, T. A prime factor FFT algorithm using high-speed convolution. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 25, n. 4, p. 281–294, 1977.

71 HAWEEL, T. I. A new square wave transform based on the DCT. **Signal Processing**, Elsevier, v. 81, n. 11, p. 2309–2319, 2001.

72 CINTRA, R. J.; BAYER, F. M. A DCT approximation for image compression. **IEEE Signal Processing Letters**, v. 18, n. 10, p. 579–582, 2011.

73 CINTRA, R. J. An integer approximation method for discrete sinusoidal transforms. **Circuits, Systems, and Signal Processing**, Springer, v. 30, n. 6, p. 1481, 2011.

74 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. Low-complexity 8×8 transform for image compression. **Electronics Letters**, IET, v. 44, n. 21, p. 1249–1250, 2008.

75 BOUGUEZEL, S.; AHMAD, M.; SWAMY, M. A multiplication-free transform for image compression. In: IEEE. **2nd International Conference on Signals, Circuits and Systems (SCS)**. Nabeul, Tunisia, 2008. p. 1–4.

76 Bouguezel, S.; Ahmad, M. O.; Swamy, M. N. S. A low-complexity parametric transform for image compression. In: IEEE. **IEEE International Symposium of Circuits and Systems (ISCAS)**. Rio de Janeiro, Brazil, 2011. p. 2145–2148.

77 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. A fast 8×8 transform for image compression. In: IEEE. **Microelectronics (ICM), 2009 International Conference on**. Marrakech, Morocco, 2009. p. 74–77.

78 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. A novel transform for image compression. In: IEEE. **2010 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)**. Washington, USA, 2010. p. 509–512.

79 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. Binary discrete cosine and Hartley transforms. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 60, n. 4, p. 989–1002, 2013.

80 OLIVEIRA, R. S. *et al.* Low-complexity 8-point DCT approximation based on angle similarity for image and video coding. **Multidimensional Systems and Signal Processing**, Springer, v. 30, n. 3, p. 1363–1394, 2019.

81 CINTRA, R. J.; BAYER, F. M.; TABLADA, C. Low-complexity 8-point DCT approximations based on integer functions. **Signal Processing**, Elsevier, v. 99, p. 201–214, 2014.

82 POTLURI, U. S. *et al.* Improved 8-point approximate DCT for image and video compression requiring only 14 additions. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 61, n. 6, p. 1727–1740, 2014.

83 BAYER, F. M.; CINTRA, R. J. DCT-like transform for image compression requires 14 additions only. **Electronics Letters**, IET, v. 48, n. 15, p. 919–921, 2012.

84 JRIDI, M.; ALFALOU, A.; MEHER, P. K. A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal approximation of DCT. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 62, n. 2, p. 449–457, 2015.

85 SILVEIRA, T. L. da *et al.* Multiplierless 16-point DCT approximation for low-complexity image and video coding. **Signal, Image and Video Processing**, Springer, v. 11, n. 2, p. 227–233, 2017.

86 CANTERLE, D. R. *et al.* A multiparametric class of low-complexity transforms for image and video coding. **Signal Processing**, Elsevier, v. 176, p. 107685, 2020.

87 PUCHALA, D. Approximate calculation of 8-point DCT for various scenarios of practical applications. **EURASIP Journal on Image and Video Processing**, Springer, v. 2021, n. 1, p. 1–34, 2021.

88 ZIDANI, N. *et al.* Low complexity pruned DCT approximation for image compression in wireless multimedia sensor networks. In: IEEE. **2019 5th International Conference on Frontiers of Signal Processing (ICFSP)**. Marseille, France, 2019. p. 26–30.

89 CHEN, J. *et al.* Hardware efficient integer discrete cosine transform for efficient image/video compression. **IEEE Access**, v. 7, p. 152635–152645, 2019.

90 HUANG, J. *et al.* A deterministic low-complexity approximate (multiplier-less) technique for DCT computation. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 66, n. 8, p. 3001–3014, 2019.

91 COELHO, D. F. *et al.* Low-complexity scaling methods for DCT-II approximations. **IEEE Transactions on Signal Processing**, p. 4557 – 4566, 2021.

92 BRAHIMI, N. *et al.* A novel and efficient 8-point DCT approximation for image compression. **Multimedia Tools and Applications**, Springer, v. 79, n. 11, p. 7615–7631, 2020.

93 SINGHADIA, A.; BANTE, P.; CHAKRABARTI, I. A novel algorithmic approach for efficient realization of 2-D-DCT architecture for HEVC. **IEEE Transactions on Consumer Electronics**, v. 65, n. 3, p. 264–273, 2019.

94 LAN, L.-S.; REED, I. S. Fast approximate Karhunen-Loève transform with applications to digital image coding. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Visual Communications and Image Processing'93**. Cambridge, Massachusetts, 1993. v. 2094, p. 444–455.

95 LAN, L.-S.; REED, I. S. An improved JPEG image coder using the adaptive fast approximate Karhunen-Loève transform (AKLT). In: IEEE. **Proceedings of ICSIPNN'94. International Conference on Speech, Image Processing and Neural Networks**. Hong Kong, 1994. p. 160–163.

96 PIROOZ, A.; REED, I. A new approximate Karhunen-Loève transform for data compression. In: IEEE. **Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers**. Pacific Grove, CA, 1998. v. 2, p. 1471–1475.

97 CAGNAZZO, M. *et al.* Low-complexity compression of multispectral images based on classified transform coding. **Signal Processing: Image Communication**, Elsevier, v. 21, n. 10, p. 850–861, 2006.

98 WONGSAWAT, Y.; ORAINTARA, S.; RAO, K. R. Integer sub-optimal Karhunen-Loève transform for multi-channel lossless EEG compression. In: IEEE. **2006 14th European Signal Processing Conference**. Florence, Italy, 2006. p. 1–5.

99 BHAIRANNAWAR, S. S.; SARKAR, S.; RAJA, K. FPGA implementation of optimized Karhunen–Loève transform for image processing applications. **Journal of Real-Time Image Processing**, Springer, v. 17, n. 2, p. 357–370, 2020.

100 ARIYARATHNA, V. *et al.* Multibeam digital array receiver using a 16-point multiplierless DFT approximation. **IEEE Transactions on Antennas and Propagation**, v. 67, n. 2, p. 925–933, 2018.

101 KULASEKERA, S. *et al.* Multi-beam receiver apertures using multiplierless 8-point approximate DFT. In: IEEE. **2015 IEEE Radar Conference (RadarCon)**. Arlington, VA, 2015. p. 1244–1249.

102 SUAREZ, D. *et al.* Multi-beam RF aperture using multiplierless FFT approximation. **Electronics Letters**, Wiley Online Library, v. 50, n. 24, p. 1788–1790, 2014.

103 COUTINHO, V. A. *et al.* An 8-beam 2.4 GHz digital array receiver based on a fast multiplierless spatial DFT approximation. In: IEEE. **2018 IEEE/MTT-S International Microwave Symposium - IMS**. Philadelphia, PA, 2018. p. 1538–1541.

104 MARTINS, W. A.; SHANKAR, M. R. B.; OTTERSTEN, B. Oversampled DFT-modulated biorthogonal filter banks: perfect reconstruction designs and multiplierless approximations. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 67, n. 11, p. 2777–2781, 2019.

105 CHAN, S.-C.; YIU, P. An efficient multiplierless approximation of the fast Fourier transform using sum-of-powers-of-two (SOPOT) coefficients. **IEEE signal processing letters**, v. 9, n. 10, p. 322–325, 2002.

106 PERERA, W. Architectures for multiplierless fast Fourier transform hardware implementation in VLSI. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 35, n. 12, p. 1750–1760, 1987.

107 MIRFARSHBAFAN, S. H.; TANER, S.; STUDER, C. SMUL-FFT: A streaming multiplierless fast Fourier transform. **IEEE Transactions on Circuits and Systems II: Express Briefs**, v. 68, n. 5, p. 1715–1719, 2021.

108 R Core Team. **R: A Language and Environment for Statistical Computing**. Vienna, Austria, 2018. Disponível em: <https://www.R-project.org/>.

109 ROSSUM, G. V.; DRAKE, F. L. **Python 3 Reference Manual**. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

110 JOLLIFFE, I. T. **Principal Component Analysis**. New York, NY: Springer-Verlag, 1986.

111 CHIPMAN, H. A.; GU, H. Interpretable dimension reduction. **Journal of Applied Statistics**, Taylor & Francis, v. 32, n. 9, p. 969–987, 2005.

112 JOHNSON, R. A.; WICHERN, D. W. *et al.* **Applied multivariate statistical analysis**. Upper Saddle River, NJ: Prentice Hall, 2002. v. 5.

113 DU, Q.; FOWLER, J. E. Hyperspectral image compression using JPEG2000 and principal component analysis. **IEEE Geoscience and Remote Sensing Letters**, v. 4, n. 2, p. 201–205, 2007.

114 RAY, W.; DRIVER, R. Further decomposition of the Karhunen-Loève series representation of a stationary random process. **IEEE Transactions on Information Theory**, v. 16, n. 6, p. 663–668, 1970.

115 AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transfom. **IEEE Transactions on Computers**, C-23, n. 1, p. 90–93, 1974.

116 CLARKE, R. Relation between the Karhunen-Loève and sine transforms. **Electronics Letters**, IET, v. 20, n. 1, p. 12–13, 1984.

117 CINTRA, R. J. *et al.* Low-complexity DCT approximations for biomedical signal processing in big data. In: **Signal Processing and Machine Learning for Biomedical Big Data**. Boca Raton: CRC Press, 2018. p. 151–176.

118 SHELTAMI, T.; MUSADDIQ, M.; SHAKSHUKI, E. Data compression techniques in wireless sensor networks. **Future Generation Computer Systems**, Elsevier, v. 64, p. 151–162, 2016.

119 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. Binary discrete cosine and Hartley transforms. **IEEE Transactions on Circuits and Systems I: Regular Papers**, IEEE, v. 60, n. 4, p. 989–1002, 2012.

120 LENGWEHASATIT, K.; ORTEGA, A. Scalable variable complexity approximate forward DCT. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 14, n. 11, p. 1236–1248, 2004.

121 HAWEEL, R. T.; EL-KILANI, W. S.; RAMADAN, H. H. Fast approximate DCT with GPU implementation for image compression. **Journal of Visual Communication and Image Representation**, Elsevier, v. 40, p. 357–365, 2016.

122 ALMURIB, H. A.; KUMAR, T. N.; LOMBARDI, F. Approximate DCT image compression using inexact computing. **IEEE Transactions on Computers**, v. 67, n. 2, p. 149–159, 2017.

123 TABLADA, C. *et al.* DCT approximations based on Chen's factorization. **Signal Processing: Image Communication**, Elsevier, v. 58, p. 14–23, 2017.

124 JAYANT, N. S.; NOLL, P. Digital coding of waveforms: principles and applications to speech and video. **Englewood Cliffs, NJ**, p. 115–251, 1984.

125 TAKALA, J.; NIKARA, J. Unified pipeline architecture for discrete sine and cosine transforms of type IV. In: **Proceedings of the 3rd Internacional Conference on Information Communication and Signal Processing**. Singapore: [s.n.], 2001.

126 KATTO, J.; KOMATSU, K.; YASUDA, Y. Short-tap and linear-phase PR filter banks for subband coding of images. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Visual Communications and Image Processing'92**. Boston, MA, 1992. v. 1818, p. 735–747.

127 SEBER, G. A. **A matrix handbook for statisticians**. Hoboken, New Jersey: John Wiley & Sons, 2008. v. 15.

128 HARVILLE, D. A. Trace of a (square) matrix. In: **Matrix Algebra From a Statistician's Perspective**. New York, NY: Springer, 1997. p. 49–53.

129 OPPENHEIM, A. V.; SCHAFER, R. W. **Digital Signal Processing:(by) Alan V. Oppenheim (and) Ronald W. Schafer**. Michigan, USA: Prentice-Hall, 1975.

130 LEVITIN, A. **Introduction To Design And Analysis Of Algorithms, 2/E**. India: Pearson Education, 2008.

131 BAYER, F. M. *et al.* A digital hardware fast algorithm and FPGA-based prototype for a novel 16-point approximate DCT for image compression applications. **Measurement Science and Technology**, IOP Publishing, v. 23, n. 11, p. 114010, 2012.

132 SILVEIRA, T. L. da *et al.* An orthogonal 16-point approximate DCT for image and video compression. **Multidimensional Systems and Signal Processing**, Springer, v. 27, n. 1, p. 87–104, 2016.

133 GUPTA, A.; RAO, K. R. A fast recursive algorithm for the discrete sine transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 38, n. 3, p. 553–557, 1990.

134 Joint Collaborative Team on Video Coding (JCT-VC). **HEVC Reference Software documentation**. 2013. Fraunhofer Heinrich Hertz Institute. Disponível em: <https://hevc.hhi.fraunhofer.de/>.

135 HUYNH-THU, Q.; GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. **Electronics Letters**, IET, v. 44, n. 13, p. 800–801, 2008.

136 WANG, Z. *et al.* Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600–612, 2004.

137 SIPI, U. **The USC-SIPI image database**. 1977. Disponível em: <http://sipi.usc.edu/database/>.

138 OHM, J.-R. *et al.* Comparison of the coding efficiency of video coding standards - including High Efficiency Video Coding (HEVC). **IEEE Transactions on Circuits and Systems for Video Technology**, v. 22, n. 12, p. 1669–1684, dez. 2012.

139 GALL, D. J. L. The MPEG video compression algorithm. **Signal Processing: Image Communication**, Elsevier, v. 4, n. 2, p. 129–140, 1992.

140 International Telecommunication Union. **ITU-T Recommendation H.261 Version 1: Video Codec for Audiovisual Services at** $p \times 64$ **kbits**. [S.l.], 1990.

141 International Telecommunication Union. **ITU-T Recommendation H.263 version 1: Video Coding for Low Bit Rate Communication**. [S.l.], 1995.

142 LUTHRA, A.; SULLIVAN, G. J.; WIEGAND, T. Introduction to the special issue on the H.264/AVC video coding standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 13, n. 7, p. 557–559, jul. 2003.

143 HARIZE, S. *et al.* Hardware implementation and performance evaluation of the HEVC intra-predicted 4×4 blocks transforms. **Circuits, Systems, and Signal Processing**, Springer, v. 36, n. 5, p. 2050–2074, 2017.

144 BOSSEN, F. **Common Test Conditions and Software Reference Configurations**. San Jose, CA, USA: [s.n.], 2013. Document JCT-VC L1100.

145 MEDIA, X. video test. **Xiph.Org Foundation**. 2014. Disponível em: <https://media.xiph.org/video/derf/>.

146 BJØNTEGAARD, G. Calculation of average PSNR differences between RD-curves. In: **13th VCEG Meeting**. Austin, TX, USA: [s.n.], 2001. Document VCEG-M33.

147 HANHART, P.; EBRAHIMI, T. Calculation of average coding efficiency based on subjective quality scores. **Journal of Visual Communication and Image Representation**, v. 25, n. 3, p. 555 – 564, 2014. ISSN 1047-3203. QoE in 2D/3D Video Systems.

148 ZENG, Y. *et al.* Approximated reconfigurable transform architecture for VVC. In: IEEE. **2021 IEEE International Symposium on Circuits and Systems (ISCAS)**. Daegu, Korea, 2021. p. 1–5.

149 IMEN, W. *et al.* DCT-II transform hardware-based acceleration for VVC standard. In: IEEE. **2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS)**. [S.l.], 2021. p. 1–5.

150 DING, K. *et al.* A fast transform algorithm based on VVC. In: **2020 4th International Conference on Computer Science and Artificial Intelligence**. Zhuhai, China: [s.n.], 2020. p. 80–85.

151 BETZEL, F. *et al.* Approximate communication: Techniques for reducing communication bottlenecks in large-scale parallel systems. **ACM Computing Surveys (CSUR)**, ACM New York, NY, USA, v. 51, n. 1, p. 1–32, 2018.

152 PENNEBAKER, W. B.; MITCHELL, J. L. **JPEG: Still image data compression standard**. Norwell, Massachusetts: Springer Science & Business Media, 1992.

153 SAYOOD, K. **Introduction to data compression**. San Francisco, CA: Morgan Kaufmann, 2017.

154 REED, I. S.; LAN, L.-S. A fast approximate Karhunen-Loève transform (AKLT) for data compression. **Journal of Visual Communication and Image Representation**, Elsevier, v. 5, n. 4, p. 304–316, 1994.

155 FAN, K. *et al.* Signal-independent separable KLT by offline training for video coding. **IEEE Access**, v. 7, p. 33087–33093, 2019.

156 HIGHAM, N. J. **Functions of matrices: theory and computation**. Philadelphia, PA: SIAM, 2008. v. 104.

157 WANG, Z.; BOVIK, A. C. Mean squared error: Love it or leave it? a new look at signal fidelity measures. **IEEE Signal Processing Magazine**, IEEE, v. 26, n. 1, p. 98–117, 2009.

158 JAYAKUMAR, R.; DHANDAPANI, S. Karhunen Loève transform with adaptive dictionary learning for coherent and random noise attenuation in seismic data. **Sādhanā**, Springer, v. 45, n. 1, p. 1–13, 2020.

159 GEETHA, V. *et al.* Hybrid optimal algorithm-based 2D discrete wavelet transform for image compression using fractional KCA. **Multimedia Systems**, Springer, v. 26, n. 6, p. 687–702, 2020.

160 ZHANG, X.; KWONG, S.; KUO, C.-C. J. Data-driven transform based compressed image quality assessment. **IEEE Transactions on Circuits and Systems for Video Technology**, 2020.

161 YANG, C. *et al.* Blind image quality assessment based on multi-scale KLT. **IEEE Transactions on Multimedia**, 2020.

162 CHEN, H.; ZENG, B. New transforms tightly bounded by DCT and KLT. **IEEE Signal Processing Letters**, v. 19, n. 6, p. 344–347, 2012.

163 THOMAKOS, D. Smoothing non-stationary time series using the discrete cosine transform. **Journal of Systems Science and Complexity**, Springer, v. 29, n. 2, p. 382–404, 2016.

164 RAO, K. R.; YIP, P. C. **Discrete Cosine Transform, Algorithm, Advantage and Applications**. San Diego, CA: [s.n.], 1990.

165 BISWAS, M.; PICKERING, M. R.; FRATER, M. R. Improved H.264-based video coding using an adaptive transform. In: IEEE. **2010 IEEE International Conference on Image Processing**. Hong Kong, 2010. p. 165–168.

166 OLIVEIRA, P. A. *et al.* A discrete Tchebichef transform approximation for image and video coding. **IEEE Signal Processing Letters**, v. 22, n. 8, p. 1137–1141, 2015.

167 OLDHAM, K. B.; MYLAND, J.; SPANIER, J. **An atlas of functions: with equator, the atlas function calculator**. New York, NY: Springer Science & Business Media, 2010.

168 PLONKA, G. A global method for invertible integer DCT and integer wavelet algorithms. **Applied and Computational Harmonic Analysis**, Academic Press, v. 16, n. 2, p. 90–110, 2004.

169 HARTIGAN, J. A.; WONG, M. A. Algorithm AS 136: A k-means clustering algorithm. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, Wiley Online Library, v. 28, n. 1, p. 100–108, 1979.

170 FLURY, B. N.; GAUTSCHI, W. An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. **SIAM Journal on Scientific and Statistical Computing**, v. 7, n. 1, p. 169–184, 1986.

171 BAGHAIE, R.; DIMITROV, V. Computing Haar transform using algebraic integers. **Conference Record of Thirty-Fourth Asilomar Conference on Signal, Systems and Computers**, v. 1, p. 438–442, 2000.

172 SAFIRI, H. *et al.* Design and FPGA implementation of systolic FIR filters using the Fermat number ALU. In: **Asilomar Conference on Signals, Systems and Computers**. [S.l.: s.n.], 1996. v. 2, p. 1052–1056.

173 PERRE, L. Van der; LIU, L.; LARSSON, E. G. Efficient DSP and circuit architectures for massive MIMO: State of the art and future directions. **IEEE Transactions on Signal Processing**, v. 66, n. 18, p. 4717–4736, 2018.

174 DOMOUCHTSIDIS, S. *et al.* Symbol-level precoding for low complexity transmitter architectures in large-scale antenna array systems. **IEEE Transactions on Wireless Communications**, v. 18, n. 2, p. 852–863, 2019.

175 IBHAZE, A. E.; ORUKPE, P. E.; EDEKO, F. O. High capacity data rate system: Review of visible light communications technology. **Journal of Electronic Science and Technology**, v. 18, n. 3, p. 100055, 2020.

176 ZHANG, L. *et al.* Using layered division multiplexing for wireless in-band distribution links in next generation broadcast systems. **IEEE Transactions on Broadcasting**, v. 67, n. 1, p. 68–82, 2021.

177 POULARIKAS, A. D. **Transforms and applications handbook**. Boca Raton: CRC press, 2010.

178 JAIN, A. K. Image data compression: A review. **Proceedings of the IEEE**, v. 69, n. 3, p. 349–389, 1981.

179 WELCH, T. A. A technique for high-performance data compression. **Computer**, n. 6, p. 8–19, 1984.

180 CLARKE, R. Application of sine transform in image processing. **Electronics Letters**, IET, v. 19, n. 13, p. 490–491, 1983.

181 TABLADA, C.; BAYER, F. M.; CINTRA, R. J. A class of DCT approximations based on the Feig–Winograd algorithm. **Signal Processing**, Elsevier, v. 113, p. 38–51, 2015.

182 COUTINHO, V. A. *et al.* A multiplierless pruned DCT-like transformation for image and video compression that requires ten additions only. **Journal of Real-Time Image Processing**, Springer, p. 1–9, 2015.

183 SHI, Y. Q.; SUN, H. **Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards**. Boca Raton: CRC press, 1999.

184 ZHAO, X. *et al.* Transform coding in the VVC standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 31, n. 10, p. 3878–3890, 2021.

185 COELHO, D. F.; CINTRA, R. J.; DIMITROV, V. S. Efficient computation of the 8-point DCT via summation by parts. **Journal of Signal Processing Systems**, Springer, v. 90, n. 4, p. 505–514, 2018.

186 SUN, H. *et al.* Approximate DCT design for video encoding based on novel truncation scheme. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 66, n. 4, p. 1517–1530, 2019.

187 LIANG, W.-D.; LIU, X.-D. Comparison of approximate DCT and approximate DTT for image compression. In: **2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering**. Nanchang, China: [s.n.], 2021. p. 337–341.

188 PAIM, G. *et al.* Power efficient 2-D rounded cosine transform with adder compressors for image compression. In: **2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS)**. Cairo, Egypt: IEEE, 2015. p. 348–351.

189 BELYAEV, E.; BIE, L.; KORHONEN, J. Motion JPEG decoding via iterative thresholding and motion-compensated deflickering. In: **2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)**. Fully Virtual: IEEE, 2020. p. 1–6.

190 BUSSON, A. J. *et al.* Video quality enhancement using deep learning-based prediction models for quantized DCT coefficients in MPEG I-frames. In: IEEE. **2020 IEEE International Symposium on Multimedia (ISM)**. Naples, Italy, 2020. p. 29–32.

191 SINGHADIA, A.; MAMILLAPALLI, M.; CHAKRABARTI, I. Hardware-efficient 2D-DCT/IDCT architecture for portable HEVC-compliant devices. **IEEE Transactions on Consumer Electronics**, v. 66, n. 3, p. 203–212, 2020.

192 MASERA, M.; MASERA, G.; MARTINA, M. An area-efficient variable-size fixed-point DCT architecture for HEVC encoding. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 30, n. 1, p. 232–242, 2020.

193 MAHER, J.; MEHER, P. K. Scalable approximate DCT architectures for efficient HEVC-compliant video coding. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 27, n. 8, p. 1815–1825, 2017.

194 SURESH, H.; HEGDE, S.; SARTORI, J. Approximate compression: enhancing compressibility through data approximation. In: IEEE. **Proceedings of the 15th IEEE/ACM Symposium on Embedded Systems for Real-Time Multimedia**. Seoul, Republic of Korea, 2017. p. 41–50.

195 SIDATY, N. *et al.* Compression performance of the versatile video coding: HD and UHD visual quality monitoring. In: IEEE. **2019 Picture Coding Symposium (PCS)**. Ningbo, China, 2019. p. 1–5.

196 DONG, J. *et al.* 2-D order-16 integer transforms for HD video coding. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 19, n. 10, p. 1462–1474, 2009.

197 THIRIPURASUNDARI, C.; SUMATHY, V.; THIRUVENGADAM, C. An FPGA implementation of novel smart antenna algorithm in tracking systems for smart cities. **Computers & Electrical Engineering**, Elsevier, v. 65, p. 59–66, 2018.

198 MADANAYAKE, A. *et al.* Low-power VLSI architectures for DCT\DWT: Precision vs approximation for HD video, biomedical, and smart antenna applications. **IEEE Circuits and Systems Magazine**, v. 15, n. 1, p. 25–47, 2015.

199 RAJAPAKSHA, N. *et al.* Asynchronous realization of algebraic integer-based 2D DCT using Achronix Speedster SPD60 FPGA. **Journal of Electrical and Computer Engineering**, v. 2013, p. 1–9, 2013.

200 MADISHETTY, S. K. *et al.* VLSI architectures for the 4-tap and 6-tap 2-D Daubechies wavelet filters using algebraic integers. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 60, n. 6, p. 1455–1468, 2012.

201 HAGHIGHAT, M. B. A.; AGHAGOLZADEH, A.; SEYEDARABI, H. Multi-focus image fusion for visual sensor networks in DCT domain. **Computers & Electrical Engineering**, Elsevier, v. 37, n. 5, p. 789–797, 2011.

202 LIANG, Y. *et al.* Image encryption combining multiple generating sequences controlled fractional DCT with dependent scrambling and diffusion. **Journal of Modern Optics**, Taylor & Francis, v. 62, n. 4, p. 251–264, 2015.

203 WAHID, K.; KO, S.-B.; TENG, D. Efficient hardware implementation of an image compressor for wireless capsule endoscopy applications. In: IEEE. **2008 IEEE International Joint Conference on Neural Networks**. Hong Kong, China, 2008. p. 2761–2765.

204 WAHID, K. A.; ISLAM, M. A.; KO, S.-B. Lossless implementation of Daubechies 8-tap wavelet transform. In: IEEE. **2011 IEEE International Symposium of Circuits and Systems**. Rio de Janeiro, Brazil, 2011. p. 2157–2160.

205 CHIPER, D. F.; COTOROBAI, L. T. A novel VLSI algorithm for a low complexity VLSI implementation of DCT based on pseudo circular correlation structures. In: IEEE. **2020 International Symposium on Electronics and Telecommunications**. Timisoara, 2020. p. 1–4.

206 CHUNG, R.-L. *et al.* VLSI implementation of a cost-efficient Loeffler DCT algorithm with recursive CORDIC for DCT-based encoder. **Electronics**, Multidisciplinary Digital Publishing Institute, v. 10, n. 7, p. 862, 2021.

207 HSIAO, S.-F. *et al.* Efficient VLSI implementations of fast multiplierless approximated DCT using parameterized hardware modules for silicon intellectual property design. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 52, n. 8, p. 1568–1579, 2005.

208 SENTHILPARI, C. *et al.* Design a low voltage amp; low power multiplier-free pipelined DCT architecture using hybrid full adder. In: IEEE. **2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS)**. Bangkok, Thailand, 2018. p. 1–6.

209 SAPONARA, S. Real-time and low-power processing of 3D direct/inverse discrete cosine transform for low-complexity video codec. **Journal of Real-Time Image Processing**, Springer, v. 7, n. 1, p. 43–53, 2012.

210 BAHAR, A. N.; WAHID, K. A. Design and implementation of approximate DCT architecture in quantum-dot cellular automata. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, v. 28, n. 12, p. 2530–2539, 2020.

211 LEE, D.-U. *et al.* Energy-efficient image compression for resource-constrained platforms. **IEEE Transactions on Image Processing**, v. 18, n. 9, p. 2100–2113, 2009.

212 MECHOUEK, K. *et al.* Low complexity DCT approximation for image compression in wireless image sensor networks. **Journal of Circuits, Systems and Computers**, World Scientific, v. 25, n. 08, p. 1650088, 2016.

213 MA, S.; AMPADU, P. Approximate memory with approximate DCT. In: **Proceedings of the 2019 on Great Lakes Symposium on VLSI**. Tysons Corner, VA: ACM, 2019. p. 355–358.

214 CAMPOBELLO, G. *et al.* RAKE: A simple and efficient lossless compression algorithm for the internet of things. In: **2017 25th European Signal Processing Conference (EUSIPCO)**. Kos island, Greece: [s.n.], 2017. p. 2581–2585.

215 HAMZA, R.; HASSAN, A.; PATIL, A. S. A lightweight secure IoT surveillance framework based on DCT-DFRT algorithms. In: SPRINGER. **International Conference on Machine Learning for Cyber Security**. Xian, China, 2019. p. 271–278.

216 MA, Z. *et al.* A detection and relative direction estimation method for UAV in sense-and-avoid. In: IEEE. **2015 IEEE International Conference on Information and Automation**. Yunnan, China, 2015. p. 2677–2682.

217 MARGELIS, G. *et al.* Efficient DCT-based secret key generation for the internet of things. **Ad Hoc Networks**, Elsevier, v. 92, p. 101744, 2019.

218 KANSAL, L. *et al.* Efficient and robust image communication techniques for 5G applications in smart cities. **Energies**, Multidisciplinary Digital Publishing Institute, v. 14, n. 13, p. 3986, 2021.

219 POTLURI, U. *et al.* Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing. **Measurement Science and Technology**, IOP Publishing, v. 23, n. 11, p. 114003, 2012.

220 ZHANG, J. *et al.* A low-power and high-PSNR unified DCT/IDCT architecture based on EARC and enhanced scale factor approximation. **IEEE Access**, v. 7, p. 165684–165691, 2019.

221 JAIN, R.; JAIN, P. FPGA implementation of recursive algorithm of DCT. In: SPRINGER. **Proceedings of International Conference on Artificial Intelligence and Applications**. Settat, Morocco, 2021. p. 203–212.

222 ESCOBAR, R. V. *et al.* Evaluation and comparison of DCT approximations on FPGA for hardware reduction. In: **2020 IEEE International Autumn Meeting on Power, Electronics and Computing**. Guerrero, Mexico: [s.n.], 2020. v. 4, p. 1–5.

223 TSOUNIS, I.; PAPADIMITRIOU, A.; PSARAKIS, M. Analyzing the impact of approximate adders on the reliability of FPGA accelerators. In: IEEE. **2021 IEEE European Test Symposium**. Fully Virtual, 2021. p. 1–2.

224 SENAPATI, R. K.; PATI, U. C.; MAHAPATRA, K. K. A low complexity orthogonal 8×8 transform matrix for fast image compression. **Proceeding of the Annual IEEE India Conference (INDICON), Kolkata, India**, p. 1–4, 2010.

225 MARDIA, K.; JUPP, P. **Directional Statistics**. England: Wiley, 2009. (Wiley Series in Probability and Statistics). ISBN 9780470317815.

226 JAMMALAMADAKA, S.; SENGUPTA, A. **Topics in Circular Statistics**. Singapore: World Scientific, 2001. v. 5. (Series on multivariate analysis, v. 5). ISBN 9789812779267.

227 STRANG, G. **Linear algebra and its applications.** Belmont, CA: Thomson, Brooks/Cole, 2006.

228 BARTLE, R. G.; SHERBERT, D. R. **Introduction to real analysis**. New York: Wiley, 2000. v. 2.

229 KATTO, J.; YASUDA, Y. Performance evaluation of subband coding and optimization of its filter coefficients. **Journal of Visual Communication and Image Representation**, v. 2, n. 4, p. 303–313, dez. 1991. ISSN 1047-3203.

230 YIP, P.; RAO, K. The decimation-in-frequency algorithms for a family of discrete sine and cosine transforms. **Circuits, Systems and Signal Processing**, Springer, v. 7, n. 1, p. 3–19, 1988.

231 OPPENHEIM, A. V.; SCHAFER, R. W.; BUCK, J. R. **Discrete-time signal processing**. 2nd. ed. Upper Saddle River, New Jersey: Prentice Hall, 1999.

232 KAY, S. M. **Fundamentals of statistical signal processing - Volume II - Detection theory**. Upper Saddle Tiver, NJ: Pearson Education, 1998.

233 HAYES, M. H. **Statistical digital signal processing and modeling**. United States: John Wiley & Sons, 1996.

234 HELSTROM, C. W. **Statistical theory of signal detection: international series of monographs in electronics and instrumentation**. [S.l.]: Elsevier, 2013. v. 9.

235 ZHANG, X.; CUI, C. Signal detection for cognitive radar. **Electronics Letters**, Wiley Online Library, v. 49, n. 8, p. 559–560, 2013.

236 MA, J.; LI, H.; GAN, L. Order-statistic based target detection with compressive measurements in single-frequency multistatic passive radar. **Signal Processing**, Elsevier, v. 203, p. 108785, 2023.

237 RAHMAN, M. H. *et al.* Deep learning based improved cascaded channel estimation and signal detection for reconfigurable intelligent surfaces-assisted MU-MISO systems. **IEEE Transactions on Green Communications and Networking**, IEEE, 2023.

238 GONG, Y. *et al.* Viterbinet-based signal detection for OTFS system. **IEEE Communications Letters**, 2023.

239 KĄKOL, K. *et al.* Detecting lombard speech using deep learning approach. **Sensors**, v. 23, n. 1, p. 315, 2023.

240 KORKMAZ, Y.; BOYACI, A. Hybrid voice activity detection system based on LSTM and auditory speech features. **Biomedical Signal Processing and Control**, Elsevier, v. 80, p. 104408, 2023.

241 SUN, T. *et al.* A multi-order fractional Fourier domain feature union method for active sonar target classification. **Applied Acoustics**, Elsevier, v. 205, p. 109257, 2023.

242 ANSARI, M. D.; GHRERA, S. P.; TYAGI, V. Pixel-based image forgery detection: A review. **IETE Journal of Education**, v. 55, n. 1, p. 40–46, 2014.

243 AVCI, H.; KARAKAYA, J. A novel medical image enhancement algorithm for breast cancer detection on mammography images using machine learning. **Diagnostics**, v. 13, n. 3, p. 348, 2023.

244 RAFIEE, J. *et al.* Wavelet basis functions in biomedical signal processing. **Expert Systems with Applications**, v. 38, n. 5, p. 6190–6201, 2011.

245 ARDETI, V. A. *et al.* An overview on state-of-the-art electrocardiogram signal processing methods: Traditional to AI-based approaches. **Expert Systems with Applications**, Elsevier, p. 119561, 2023.

246 GALLUZZO, D. *et al.* Detection of low frequency seismicity at mt. vesuvius based on coherence and statistical moments of seismic signals. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 13, n. 1, p. 194, 2023.

247 CHIEN, C.-C. *et al.* Automatic classification with an autoencoder of seismic signals on a distributed acoustic sensing cable. **Computers and Geotechnics**, Elsevier, v. 155, p. 105223, 2023.

248 WELCH, P. The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. **IEEE Transactions on audio and electroacoustics**, v. 15, n. 2, p. 70–73, 1967.

249 COOLEY, J. W.; LEWIS, P. A.; WELCH, P. D. The fast Fourier transform and its applications. **IEEE Transactions on Education**, IEEE, v. 12, n. 1, p. 27–34, 1969.

250 SINGLETON, R. An algorithm for computing the mixed radix fast Fourier transform. **IEEE Transactions on audio and electroacoustics**, IEEE, v. 17, n. 2, p. 93–103, 1969.

251 ARIYARATHNA, V. *et al.* Multibeam digital array receiver using a 16-point multiplierless DFT approximation. **IEEE Transactions on Antennas and Propagation**, v. 67, n. 2, p. 925–933, 2019.

252 RADÜNZ, A. P. *et al.* Data-independent low-complexity KLT approximations for image and video coding. **Signal Processing: Image Communication**, Elsevier, v. 101, p. 116585, 2022.

253 RADÜNZ, A. P.; BAYER, F. M.; CINTRA, R. J. Low-complexity rounded KLT approximation for image compression. **Journal of Real-Time Image Processing**, Springer, p. 1–11, 2021.

254 AXELSSON, S. R. Noise radar using random phase and frequency modulation. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 42, n. 11, p. 2370–2384, 2004.

255 HUNG, E.; HERRING, R. Simulation experiments to compare the signal detection properties of DFT and MEM spectra. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 29, n. 5, p. 1084–1089, 1981.

256 MANN, M. E.; LEES, J. M. Robust estimation of background noise and signal detection in climatic time series. **Climatic Change**, Springer, v. 33, n. 3, p. 409–445, 1996.

257 SCHULTE, S. *et al.* A fuzzy impulse noise detection and reduction method. **IEEE Transactions on Image Processing**, IEEE, v. 15, n. 5, p. 1153–1162, 2006.

258 WAX, M.; KAILATH, T. Detection of signals by information theoretic criteria. **IEEE Transactions on acoustics, speech, and signal processing**, IEEE, v. 33, n. 2, p. 387–392, 1985.

259 CASELLA, G.; BERGER, R. L. **Statistical inference**. Pacific Grove, CA: Cengage Learning, 2021.

260 HEIDEMAN, M. T. Multiplicative complexity of linear and bilinear systems. In: **Multiplicative Complexity, Convolution, and the DFT**. [S.l.]: Springer, 1988. p. 5–26.

261 COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. **Mathematics of computation**, v. 19, n. 90, p. 297–301, 1965.

262 PROAKIS, J. G. **Digital signal processing: principles, algorithms, and applications, 4/E**. Upper Saddle River, New Jersey: Pearson Education, 2007.

263 DUMMIT, D. S.; FOOTE, R. M. **Abstract algebra**. Hoboken: Wiley, 2004. v. 3.

264 ZASSENHAUS, H. J. **The theory of groups**. New York: Courier Corporation, 2013.

265 ROTMAN, J. J. **An introduction to the theory of groups**. New York: Springer Science & Business Media, 2012. v. 148.

APPENDIX A – MATRIX FACTORIZATION SKLT

In the following appendix, we present the matrices factorization of length $N = 16$ and 32 from Chapter 2.

- For $N = 16$, we can rewrite $\widehat{\mathbf{T}}_{16,1}$, $\widehat{\mathbf{T}}_{16,2}$, and $\widehat{\mathbf{T}}_{16,3}$ [71] as follows

$$\widehat{\mathbf{T}}_{16,1} = \frac{1}{4} \cdot \mathbf{P}_{16} \cdot \mathbf{A}'_{16,3} \cdot \mathbf{A}_{16,2} \cdot \mathbf{A}_{16,1},$$
$$\widehat{\mathbf{T}}_{16,2} = \frac{1}{4} \cdot \mathbf{P}_{16} \cdot \mathbf{A}'_{16,5} \cdot \mathbf{A}'_{16,4} \cdot \mathbf{A}''_{16,3} \cdot \mathbf{A}_{16,2} \cdot \mathbf{A}_{16,1},$$
$$\widehat{\mathbf{T}}_{16,3} = \frac{1}{4} \cdot \mathbf{P}_{16} \cdot \mathbf{A}''_{16,5} \cdot \mathbf{A}''_{16,4} \cdot \mathbf{A}'''_{16,3} \cdot \mathbf{A}_{16,2} \cdot \mathbf{A}_{16,1},$$

where

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

$$\mathbf{A}_{16,1} = \begin{bmatrix} \mathbf{I}_8 & \bar{\mathbf{I}}_8 \\ \bar{\mathbf{I}}_8 & -\mathbf{I}_8 \end{bmatrix}, \quad \mathbf{A}_{16,2} = \begin{bmatrix} \mathbf{A}_{8,1} & \\ & \mathbf{A}_{8,1} \end{bmatrix},$$

$$\mathbf{A}'_{16,3} = \begin{bmatrix} 1 & 1 & 1 & 1 & & & & & & & & & & & & \\ 1 & 1 & - & - & 1 & & & & & & & & & & & \\ 1 & - & - & 1 & & & & & & & & & & & & \\ 1 & - & 1 & - & & & & & & & & & & & & \\ & & & 1 & & 1 & 1 & 1 & & & & & & & & \\ & 1 & & & - & - & & 1 & & & & & & & & \\ & & & & 1 & 1 & - & 1 & & & & & & & & \\ & & & & - & 1 & - & 1 & & & & & & & & \\ & & & & & & & & 1 & 1 & 1 & 1 & & & & \\ & & & & & & & & 1 & 1 & - & - & 1 & & & \\ & & & & & & & & 1 & - & 1 & & & 1 & & \\ & & & & & & & & 1 & - & 1 & & 1 & & & \\ & & & & & & & & & & & - & - & - & - & \\ & & & & & & & & - & & & 1 & 1 & - & \\ & & & & & & & & & 1 & & - & & 1 & - & \\ & & & & & & & & & & & 1 & - & 1 & - & \end{bmatrix},$$

$$\mathbf{A}''_{16,3} = \begin{bmatrix} 1 & 1 & 1 & & & & & \\ 1 & 1 & - & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & 1 & \\ & & & & & 1 & - & \\ & & & & & & & \mathbf{I}_8 \end{bmatrix}, \quad \mathbf{A}'_{16,4} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & 1 & & \\ & & & & 1 & - & & \\ & & & & & & \mathbf{I}_8 & \end{bmatrix},$$

$$\mathbf{A}'_{16,5} = \begin{bmatrix} 1 & 1 & & 1 & & & & & & & & & & & & \\ & 1 & 1 & - & & & & & & & & & & & & \\ & - & 1 & 1 & & & & & & & & & & & & \\ 1 & - & & - & & & & & & & & & & & & \\ & & & 1 & & 1 & & & & & & & & & & \\ & 1 & & & - & & & - & & & & & & & & \\ & & & & 1 & & 1 & & & & & & & & & \\ & & & & - & & 1 & & & & & & & & & \\ & & & & & & & & 1 & 1 & 1 & 1 & & & & \\ & & & & & & & & 1 & 1 & & - & & 1 & & \\ & & & & & & & & 1 & & - & 1 & & & 1 & \\ & & & & & & & & 1 & - & 1 & & 1 & & & \\ & & & & & & & & & & & - & & - & - & - \\ & & & & & & & & - & & & & 1 & 1 & & - \\ & & & & & & & & & 1 & & - & & & 1 & - \\ & & & & & & & & & & & & 1 & - & 1 & - \end{bmatrix},$$

$$\mathbf{A}'''_{16,3} = \begin{bmatrix} \mathbf{A}_{8,2} & \\ & \mathbf{I}_8 \end{bmatrix}, \quad \mathbf{A}''_{16,4} = \begin{bmatrix} 1 & 1 & & & & & & & & & & & & & & \\ 1 & - & & & & & & & & & & & & & & \\ & & 1 & 1 & & & & & & & & & & & & \\ & & 1 & - & & & & & & & & & & & & \\ & & & & 1 & & & & & & & & & & & \\ & & & & & 1 & & & & & & & & & & \\ & & & & & & 1 & & & & & & & & & \\ & & & & & & & 1 & & & & & & & & \\ & & & & & & & & \mathbf{I}_8 & & & & & & & \end{bmatrix},$$

$$\mathbf{A}''_{16,5} = \begin{bmatrix} 1 & & & & & & & & & & & & & & & \\ & 1 & & & & & & & & & & & & & & \\ & & - & & & & & & & & & & & & & \\ & & & 1 & 1 & & & & & & & & & & & \\ & & & & - & & - & & & & & & & & & \\ & & & 1 & & 1 & & & & & & & & & & \\ & & & & 1 & - & & & & & & & & & & \\ & & & & & & 1 & 1 & 1 & 1 & & & & & & \\ & & & & & & 1 & 1 & & - & & 1 & & & & \\ & & & & & & 1 & & - & & - & & 1 & & & \\ & & & & & & 1 & - & 1 & & 1 & & & & & \\ & & & & & & & & & - & & - & - & - & \\ & & & & & & & & - & & 1 & & 1 & & - & \\ & & & & & & & & & 1 & & - & & 1 & - & \\ & & & & & & & & & & & & 1 & - & 1 & - \end{bmatrix}.$$

- Considering $N = 32$, we have

$$\widehat{\mathbf{T}}_{32,1} = \frac{1}{\sqrt{32}} \cdot \mathbf{P}_{32} \cdot \mathbf{A}'_{32,3} \cdot \mathbf{A}_{32,2} \cdot \mathbf{A}_{32,1},$$
$$\widehat{\mathbf{T}}_{32,2} = \frac{1}{\sqrt{32}} \cdot \mathbf{P}_{32} \cdot \mathbf{A}''_{32,3} \cdot \mathbf{A}_{32,2} \cdot \mathbf{A}_{32,1},$$
$$\widehat{\mathbf{T}}_{32,3} = \frac{1}{\sqrt{32}} \cdot \mathbf{P}_{32} \cdot \mathbf{A}'''_{32,3} \cdot \mathbf{A}_{32,2} \cdot \mathbf{A}_{32,1},$$
$$\widehat{\mathbf{T}}_{32,4} = \frac{1}{\sqrt{32}} \cdot \mathbf{P}_{32} \cdot \mathbf{A}'_{32,6} \cdot \mathbf{A}'_{32,5} \cdot \mathbf{A}'_{32,4} \cdot \mathbf{A}''''_{32,3} \cdot \mathbf{A}_{32,2} \cdot \mathbf{A}_{32,1},$$

where

$$\mathbf{P}_{32} = \begin{bmatrix} 1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & 1 & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & 1 & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & 1 & & \\ & 1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & 1 & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\ & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \end{bmatrix},$$

$$\mathbf{A}_{32,1} = \begin{bmatrix} \mathbf{I}_{16} & \bar{\mathbf{I}}_{16} \\ \bar{\mathbf{I}}_{16} & -\mathbf{I}_{16} \end{bmatrix}, \quad \mathbf{A}_{32,2} = \begin{bmatrix} \mathbf{A}_{16,1} & \\ & \mathbf{A}_{16,1} \end{bmatrix},$$

$$\mathbf{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 & 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 & & - & & & & & & \\ & & & & & & & & & & & & & & & & 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},$$

$$\mathbf{A}''_{32,3} = \left[\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & & & - & - & \\
1 & 1 & & - & - & - & & 1 \\
1 & & - & - & 1 & 1 & & - \\
1 & - & - & 1 & 1 & - & - & 1 \\
1 & - & & 1 & - & - & 1 & - \\
1 & - & 1 & - & - & 1 & - & 1 \\
1 & - & 1 & - & 1 & - & 1 & - \\
 & & & & 1 & 1 & & \\
 & & 1 & 1 & & & - & \\
 & 1 & & - & & & & \\
 & & & 1 & & & & \\
 & & & & 1 & & & \\
 & & & & - & 1 & - & 1 \\
 & & & & & - & 1 & - \\
 & & & & & & - & 1
\end{array}\right.$$

$$\left.\begin{array}{cccccccc}
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & & - & - \\
1 & 1 & 1 & - & - & - & & 1 \\
1 & 1 & - & - & & 1 & 1 & - \\
1 & & - & & 1 & & - & 1 \\
1 & & & 1 & & 1 & & - \\
1 & - & & & - & 1 & - & \\
1 & - & 1 & - & 1 & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & - & & & & \\
 & & & & - & & & \\
 & & & & & - & & \\
 & & & & & & - & \\
 & & & & & & & -
\end{array}\right.$$

$$\left.\begin{array}{cccccccc}
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & & - & - & \\
1 & 1 & & - & - & - & & 1 \\
1 & 1 & - & - & & 1 & & - \\
1 & & - & & 1 & & - & 1 \\
1 & & & 1 & & 1 & & - \\
1 & - & & & - & 1 & - & \\
1 & - & 1 & - & 1 & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & - & & & & \\
 & & & & - & & & \\
 & & & & & - & & \\
 & & & & & & - & \\
 & & & & & & & -
\end{array}\right] ,$$

$$\mathbf{A}'''_{32,3} = \left[\begin{array}{cccccccc}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & & & - & - & \\
1 & 1 & & - & - & - & & 1 \\
1 & & - & - & 1 & 1 & 1 & - \\
1 & - & - & 1 & 1 & - & - & 1 \\
1 & - & 1 & 1 & - & - & 1 & - \\
1 & - & 1 & - & - & 1 & - & 1 \\
1 & - & 1 & - & 1 & - & 1 & - \\
 & & & & 1 & 1 & & \\
 & & 1 & 1 & & & - & \\
 & & & & & & & \\
 & & & & 1 & & & \\
 & & & & & 1 & & \\
 & & & & - & 1 & 1 & - \\
 & & & & 1 & - & 1 & - \\
 & & & & - & 1 & - & 1
\end{array}\right.$$

$$\left.\begin{array}{cccccccc}
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & & - & - & \\
1 & 1 & & - & - & - & & 1 \\
1 & 1 & - & - & & 1 & & - \\
1 & & - & & 1 & & - & 1 \\
1 & & & 1 & & 1 & & - \\
1 & - & & & - & 1 & - & \\
1 & - & 1 & - & 1 & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & - & & & & \\
 & & & & - & & & \\
 & & & & & - & & \\
 & & & & & & - & \\
 & & & & & & & -
\end{array}\right.$$

$$\left.\begin{array}{cccccccc}
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & & & & & \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & & - & - & \\
1 & 1 & & - & - & - & & 1 \\
1 & 1 & - & - & & 1 & & - \\
1 & & - & & 1 & & - & 1 \\
1 & & & 1 & & 1 & & - \\
1 & - & & & - & 1 & - & \\
1 & - & 1 & - & 1 & & & \\
 & & & & & & & \\
 & & & & & & & \\
 & & & - & & & & \\
 & & & & - & & & \\
 & & & & & - & & \\
 & & & & & & - & \\
 & & & & & & & -
\end{array}\right] ,$$

$$\mathbf{A}''''_{32,3} = \begin{bmatrix} \mathbf{A}_{16,2} & \\ & \mathbf{I}_{16} \end{bmatrix}, \quad \mathbf{A}'_{32,4} = \begin{bmatrix} \mathbf{A}'''_{16,3} & \\ & \mathbf{I}_{16} \end{bmatrix}, \quad \mathbf{A}'_{32,5} = \begin{bmatrix} \mathbf{A}''_{16,4} & \\ & \mathbf{I}_{16} \end{bmatrix},$$

and

$$\mathbf{A}'_{32,6} = \left[\begin{array}{cccccccccccccccccccccccccccccccc}
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 & & - & & - & & & 1 & & - \\
 & & & & & & & & & & & & & & & - & & 1 & & - & & 1 & & 1 & & - & & 1 & & - \\
 & & & & & & & & & & & & & & & - & & & - & & 1 & & - & & 1 & & - & 1 & & - \\
 & & & & & & & & & & & & & & & & 1 & - & & & 1 & - & & 1 & & & 1 & & 1 & - \\
 & & & & & & & & & & & & & & & & & - & 1 & - & & & - & 1 & & & & - & 1 & - \\
 & & & & & & & & & & & & & & & & & & & & & & 1 & - & 1 & - & 1 & - & 1 & -
\end{array}\right].$$

APPENDIX B – 2D TRANSFORMATION AND QUANTIZATION STEP

In the present appendix, the 2D transformation and quantization step from the image compression scheme cited in Chapter 3 is detailed.

Let $\mathbf{A}$ be an 8×8 sub-block from an image. The 2D transformation from $\mathbf{A}$ induced by an approximation $\widehat{\mathbf{T}}$ is given by

$$\begin{aligned}\mathbf{B} &= \begin{cases} \widehat{\mathbf{T}} \cdot \mathbf{A} \cdot \widehat{\mathbf{T}}^{\top}, & \text{if } \mathbf{T} \text{ is orthogonal,} \\ \widehat{\mathbf{T}} \cdot \mathbf{A} \cdot \widehat{\mathbf{T}}^{-1}, & \text{if } \mathbf{T} \text{ is non-orthogonal,} \end{cases} \\ &= \begin{cases} (\mathbf{u} \cdot \mathbf{u}^{\top}) \odot (\mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^{\top}), & \text{if } \mathbf{T} \text{ is orthogonal,} \\ (\mathbf{u} \cdot \mathbf{v}^{\top}) \odot (\mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^{-1}), & \text{if } \mathbf{T} \text{ is non-orthogonal,} \end{cases} \\ &= \mathbf{R} \odot \widehat{\mathbf{B}}, \end{aligned} \tag{B.1}$$

where $\mathbf{u} = \text{diag}(\mathbf{S})$ and $\mathbf{v}$ is given by the inverse elements from $\mathbf{u}$. In the context of JPEG-like compression [26], the quantized coefficient matrix $\widehat{\mathbf{B}}$ is given by

$$\bar{\mathbf{B}} = \text{round}(\mathbf{B} \div \mathbf{Q}), \tag{B.2}$$

where $\mathbf{Q}$ is a quantization matrix and $\div$ denotes the element-wise matrix division.

By applying Equation (B.1) into (B.2), we obtain

$$\bar{\mathbf{B}} = \text{round}(\mathbf{R} \odot \widehat{\mathbf{B}} \div \mathbf{Q}) = \text{round}(\widehat{\mathbf{B}} \div \tilde{\mathbf{Q}}),$$

where $\tilde{\mathbf{Q}} = \mathbf{Q} \div \mathbf{R}$. Note that $\mathbf{R}$ can be absorbed in the quantization step, thus, the complexity of matrix $\mathbf{S}$ can be dismissed in the image compression applications [12, 16, 153].

APPENDIX C – MATRIX FACTORIZATION DCT BASED ON ANGLE SIMILARITY

The matrices used in the Chapter 5 are detailed here. First, consider the following butterfly-structure:

$$\mathbf{B}_N = \begin{bmatrix} \mathbf{I}_{\frac{N}{2}} & \bar{\mathbf{I}}_{\frac{N}{2}} \\ -\bar{\mathbf{I}}_{\frac{N}{2}} & \mathbf{I}_{\frac{N}{2}} \end{bmatrix}.$$

For the sake of brevity, cycle notation [263–265] is used to present the permutation matrices. The resulting permutation matrix is obtained by permuting the columns of the identity matrix following the zero indexed cycle mapping.

- The low-complexity matrix $\mathbf{T}_{16,5}$ can be represented as

$$\mathbf{T}_{16,5} = \mathbf{P}_{16} \cdot \mathbf{M}_{16} \cdot \begin{bmatrix} \mathbf{B}_2 & \\ & \mathbf{I}_{14} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_4 & \\ & \mathbf{I}_{12} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_8 & \\ & \mathbf{I}_8 \end{bmatrix} \cdot \mathbf{B}_{16}$$

where

$$\mathbf{P}_{16} = (1\ 8)(2\ 4)(3\ 12\ 9)(5\ 6\ 10)(7\ 14\ 13\ 11),$$

$$\mathbf{M}_{16} = \begin{bmatrix} 1 & & & & & & & & & & & & & & & \\ & -1 & & & & & & & & & & & & & & \\ & & -1 & -2 & & & & & & & & & & & & \\ & & 2 & -1 & & & & & & & & & & & & \\ & & & & -\frac{1}{2} & -1 & -2 & -2 & & & & & & & & \\ & & & & 1 & 2 & \frac{1}{2} & -2 & & & & & & & & \\ & & & & -2 & -\frac{1}{2} & 2 & -1 & & & & & & & & \\ & & & & 2 & -2 & 1 & -\frac{1}{2} & & & & & & & & \\ & & & & & & & & -\frac{1}{4} & -\frac{1}{2} & -1 & -1 & -2 & -2 & -2 & -2 \\ & & & & & & & & \frac{1}{2} & 2 & 2 & 2 & 1 & -\frac{1}{4} & -1 & -2 \\ & & & & & & & & -1 & -2 & -1 & \frac{1}{2} & 2 & 2 & -\frac{1}{4} & -2 \\ & & & & & & & & 1 & 2 & -\frac{1}{2} & -2 & -\frac{1}{4} & 2 & 1 & -2 \\ & & & & & & & & -2 & -1 & 2 & \frac{1}{4} & -2 & \frac{1}{2} & 2 & -1 \\ & & & & & & & & 2 & -\frac{1}{4} & -2 & 2 & -\frac{1}{2} & -1 & 2 & -1 \\ & & & & & & & & -2 & 1 & -\frac{1}{4} & -1 & 2 & -2 & 2 & -\frac{1}{2} \\ & & & & & & & & 2 & -2 & 2 & -2 & 1 & -1 & \frac{1}{2} & -\frac{1}{4} \end{bmatrix}.$$

- The low-complexity matrix $\mathbf{T}_{32,2}$ can be represented as

$$\mathbf{T}_{32,2} = \mathbf{P}_{32} \cdot \begin{bmatrix} \mathbf{L}_1 & \\ & \mathbf{L}_2 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_2 & \\ & \mathbf{I}_{30} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_4 & \\ & \mathbf{I}_{28} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_8 & \\ & \mathbf{I}_{24} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_{16} & \\ & \mathbf{I}_{16} \end{bmatrix} \cdot \mathbf{B}_{32}$$

where

$$\mathbf{P}_{32} = (1\ 16\ 5\ 20\ 13\ 26\ 25\ 23\ 19\ 11\ 18\ 9\ 10\ 14\ 30)(2\ 8\ 6\ 28\ 29\ 31\ 3\ 24\ 21\ 15)(4\ 12\ 22\ 17\ 7),$$

$$\mathbf{L}_1 = \begin{bmatrix}
1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & -1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & -\frac{1}{2} & -1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & 1 & -\frac{1}{2} & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & \frac{1}{2} & 1 & & -1 & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & -1 & & 1 & -\frac{1}{2} & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & 1 & -1 & \frac{1}{2} & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & -\frac{1}{2} & -1 & -1 & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & \frac{1}{2} & 1 & 1 & 1 & \frac{1}{2} & & -\frac{1}{2} & -1 & & & & & & & & & & & & & & & \\
 & & & & & & & & -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & & -1 & & & & & & & & & & & & & & & \\
 & & & & & & & & \frac{1}{2} & 1 & -\frac{1}{2} & -1 & & 1 & \frac{1}{2} & -1 & & & & & & & & & & & & & & & \\
 & & & & & & & & -1 & -\frac{1}{2} & 1 & & -1 & \frac{1}{2} & 1 & -\frac{1}{2} & & & & & & & & & & & & & & \\
 & & & & & & & & 1 & & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -\frac{1}{2} & & & & & & & & & & & & & & \\
 & & & & & & & & -1 & \frac{1}{2} & & -\frac{1}{2} & 1 & -1 & 1 & -\frac{1}{2} & & & & & & & & & & & & & & \\
 & & & & & & & & 1 & -1 & 1 & -1 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & & & & & & & & & & & & & & & \\
 & & & & & & & & & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -1 & -1 & -1 & -1 & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & -\frac{1}{2} & -1 & -1 & -1 & -1 & -\frac{1}{2} & & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -1 & -1 \\
 & & & & & & & & & & & & & & & & \frac{1}{2} & 1 & 1 & 1 & 1 & \frac{1}{2} & & -\frac{1}{2} & -\frac{1}{2} & -1 & -1 & -1 & -1 \\
 & & & & & & & & & & & & & & & & -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & & -1 & \frac{1}{2} & 1 & -\frac{1}{2} & 1 & -\frac{1}{2} \\
 & & & & & & & & & & & & & & & & \frac{1}{2} & 1 & -\frac{1}{2} & -1 & & 1 & \frac{1}{2} & -1 & \frac{1}{2} & 1 & -\frac{1}{2} & 1 & -\frac{1}{2} \\
 & & & & & & & & & & & & & & & & -1 & -\frac{1}{2} & 1 & & -1 & \frac{1}{2} & 1 & -\frac{1}{2} & & 1 & -\frac{1}{2} & 1 & -\frac{1}{2} \\
 & & & & & & & & & & & & & & & & 1 & & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -\frac{1}{2} & & 1 & -\frac{1}{2} & 1 & -\frac{1}{2} \\
 & & & & & & & & & & & & & & & & -1 & \frac{1}{2} & & -\frac{1}{2} & 1 & -1 & 1 & -\frac{1}{2} & & 1 & -\frac{1}{2} & 1 & -\frac{1}{2} \\
 & & & & & & & & & & & & & & & & 1 & -1 & 1 & -1 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & & & 1 & -\frac{1}{2} & 1 & -\frac{1}{2} \\
 & & & & & & & & & & & & & & & & & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -1 & -1 & -1 & -1 & & & 1 & -\frac{1}{2} & 1 & -\frac{1}{2} \\
 & & & & & & & & & & & & & & & & & \frac{1}{2} & 1 & 1 & 1 & 1 & 1 & 1 & & & & 1 & -\frac{1}{2} & -1 \\
 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & &
\end{bmatrix},$$

$$\mathbf{L}_2 = \begin{bmatrix}
-\frac{1}{2} & -1 & -1 & -1 & -1 & -\frac{1}{2} & & \frac{1}{2} & 1 & 1 & 1 & \frac{1}{2} & & -\frac{1}{2} & -1 & -1 \\
\frac{1}{2} & 1 & 1 & 1 & & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 & & -\frac{1}{2} & -1 \\
-\frac{1}{2} & -1 & -1 & & 1 & 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & & 1 & 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\
\frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & 1 & 1 & \frac{1}{2} & -1 & -1 & & 1 & 1 & & -1 \\
-\frac{1}{2} & -1 & & 1 & \frac{1}{2} & -1 & -1 & & 1 & \frac{1}{2} & -1 & -1 & \frac{1}{2} & 1 & \frac{1}{2} & -1 \\
1 & 1 & -\frac{1}{2} & -1 & \frac{1}{2} & 1 & & -1 & & 1 & \frac{1}{2} & -1 & -\frac{1}{2} & 1 & \frac{1}{2} & -1 \\
-1 & -\frac{1}{2} & 1 & \frac{1}{2} & -1 & -\frac{1}{2} & 1 & & -1 & & 1 & -\frac{1}{2} & -1 & \frac{1}{2} & 1 & -1 \\
1 & \frac{1}{2} & -1 & \frac{1}{2} & 1 & -1 & -\frac{1}{2} & 1 & & -1 & 1 & \frac{1}{2} & -1 & & 1 & -\frac{1}{2} \\
-1 & & 1 & -1 & & 1 & -1 & -\frac{1}{2} & 1 & -1 & -\frac{1}{2} & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -\frac{1}{2} \\
1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & & 1 & -1 & \frac{1}{2} & \frac{1}{2} & -1 & 1 & & -1 & 1 & -\frac{1}{2} \\
-1 & \frac{1}{2} & & -1 & 1 & -1 & \frac{1}{2} & \frac{1}{2} & -1 & 1 & -\frac{1}{2} & & 1 & -1 & 1 & -\frac{1}{2} \\
1 & -1 & \frac{1}{2} & & -\frac{1}{2} & 1 & -1 & 1 & -\frac{1}{2} & & \frac{1}{2} & -1 & 1 & -1 & 1 & -\frac{1}{2} \\
-1 & 1 & -1 & \frac{1}{2} & -\frac{1}{2} & & \frac{1}{2} & -\frac{1}{2} & 1 & -1 & 1 & -1 & 1 & -1 & \frac{1}{2} & \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & & \\
 & & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
 & \frac{1}{2} & 1 & 1 & 1 & 1 & 1 & 1 & \frac{1}{2} & \frac{1}{2} & & -\frac{1}{2} & -\frac{1}{2} & -1 & -1 & -1
\end{bmatrix}.$$

- The low-complexity matrix $\mathbf{T}_{64,1}$ can be represented as

$$\mathbf{T}_{64,1} = \mathbf{P}_{64} \cdot \begin{bmatrix} \mathbf{Z}_1 & & \\ & \mathbf{Z}_2 & \\ & & \mathbf{Z}_3 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_2 & \\ & \mathbf{I}_{62} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_4 & \\ & \mathbf{I}_{60} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_8 & \\ & \mathbf{I}_{56} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_{16} & \\ & \mathbf{I}_{48} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{B}_{32} & \\ & \mathbf{I}_{32} \end{bmatrix} \cdot \mathbf{B}_{64}$$

where

$\mathbf{P}_{64}$ =(1 32 17 22 42 37 27 62 5 40 33 19 30 14 4 24 50 53 59 9 28 2 16 18 26 58 7 8 20 34 21 38 29 6 56)

(3 48 49 51 55 63 13 60 11 44 41 35 23 46 45 43 39 31 10 36 25 54 61 15 12 52 57),

$$\mathbf{Z}_1 = \begin{bmatrix}
1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & -1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & -1 & -1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & 1 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & 1 & 1 & & -1 & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & -1 & & 1 & -1 & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & 1 & -1 & 1 & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & -1 & -1 & -1 & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & -1 & -1 & -1 & & 1 & 1 & & -1 & & & & & & & & & & & & & & & \\
 & & & & & & & & 1 & 1 & & -1 & & 1 & 1 & -1 & & & & & & & & & & & & & & & \\
 & & & & & & & & -1 & -1 & 1 & & -1 & & 1 & -1 & & & & & & & & & & & & & & & \\
 & & & & & & & & 1 & & -1 & 1 & & -1 & 1 & -1 & & & & & & & & & & & & & & & \\
 & & & & & & & & -1 & 1 & & -1 & 1 & -1 & 1 & & & & & & & & & & & & & & & & \\
 & & & & & & & & 1 & -1 & 1 & -1 & 1 & -1 & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & -1 & -1 & -1 & -1 & -1 & -1 & & & & & & & & & & & & & \\
 & & & & & & & & & & & & 1 & 1 & 1 & 1 & & -1 & -1 & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & & & & & & & & & & & & &
\end{bmatrix},$$

$$\mathbf{Z}_2 = \begin{bmatrix}
-1 & -1 & -1 & & 1 & 1 & 1 & & -1 & -1 & & 1 & 1 & 1 & & -1 \\
1 & 1 & 1 & -1 & -1 & & 1 & 1 & & -1 & -1 & & 1 & 1 & & -1 \\
-1 & -1 & & 1 & 1 & -1 & -1 & & 1 & 1 & -1 & -1 & & 1 & & -1 \\
1 & 1 & -1 & -1 & & 1 & & -1 & & 1 & & -1 & -1 & 1 & 1 & -1 \\
-1 & -1 & 1 & 1 & -1 & & 1 & & -1 & & 1 & & -1 & 1 & 1 & -1 \\
1 & & -1 & & 1 & -1 & -1 & 1 & & -1 & 1 & 1 & -1 & & 1 & -1 \\
-1 & & 1 & -1 & & 1 & -1 & & 1 & -1 & & 1 & -1 & -1 & 1 & -1 \\
1 & & -1 & 1 & -1 & & & 1 & -1 & & 1 & -1 & & -1 & 1 & -1 \\
-1 & 1 & & -1 & 1 & -1 & & 1 & -1 & & 1 & -1 & 1 & & 1 & \\
1 & -1 & & & -1 & 1 & -1 & & 1 & -1 & & 1 & -1 & 1 & & \\
-1 & 1 & -1 & 1 & & & -1 & 1 & -1 & & 1 & -1 & & 1 & & \\
1 & -1 & 1 & -1 & 1 & -1 & & 1 & -1 & & 1 & -1 & & & & \\
& & & & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
& & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & -1 & -1 & -1 & -1 \\
& & 1 & 1 & 1 & & -1 & -1 & -1 & -1 & & 1 & 1 & 1 & & -1 & -1 \\
& -1 & -1 & -1 & -1 & -1 & & & 1 & 1 & 1 & 1 & 1 & & & -1 & -1
\end{bmatrix}, \quad \mathbf{Z}_3 = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{bmatrix},$$

and

$$\mathbf{Q}_1 = \begin{bmatrix}
-1 & -1 & -1 & & 1 & 1 & 1 & & -1 & -1 & -1 & & 1 & 1 & 1 & \\
1 & 1 & 1 & & -1 & -1 & & 1 & 1 & 1 & & -1 & -1 & & 1 & 1 \\
-1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & & 1 & 1 & & -1 & -1 & \\
1 & 1 & & -1 & -1 & & 1 & 1 & & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & & 1 & 1 & -1 & -1 & & 1 & 1 & & -1 & & 1 & 1 & \\
1 & 1 & & -1 & & 1 & 1 & -1 & -1 & & 1 & 1 & -1 & -1 & & 1 \\
-1 & -1 & 1 & 1 & & -1 & & 1 & 1 & -1 & -1 & 1 & 1 & & -1 & \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & & 1 & & -1 & & 1 & & -1 \\
-1 & -1 & 1 & & 1 & -1 & & 1 & 1 & -1 & & 1 & & -1 & & 1 \\
1 & 1 & -1 & & 1 & & -1 & & 1 & -1 & -1 & 1 & 1 & -1 & & 1 \\
-1 & -1 & 1 & & -1 & 1 & 1 & -1 & & 1 & & -1 & 1 & 1 & -1 & \\
1 & & -1 & 1 & 1 & -1 & & 1 & -1 & -1 & 1 & & -1 & 1 & 1 & -1 \\
-1 & & 1 & -1 & & 1 & -1 & -1 & 1 & & -1 & 1 & & -1 & 1 & \\
1 & & -1 & 1 & & -1 & 1 & & -1 & 1 & & -1 & 1 & & -1 & 1 \\
-1 & & 1 & -1 & 1 & 1 & -1 & 1 & & -1 & 1 & & -1 & 1 & -1 & \\
1 & & -1 & 1 & -1 & & 1 & -1 & 1 & & -1 & 1 & 1 & -1 & & 1
\end{bmatrix},$$

$$\mathbf{Q}_2 = \begin{bmatrix}
-1 & -1 & & 1 & 1 & 1 & & -1 & -1 & -1 & & 1 & 1 & 1 & & -1 \\
& -1 & -1 & -1 & & 1 & 1 & & -1 & -1 & -1 & 1 & 1 & 1 & & -1 \\
1 & 1 & & -1 & -1 & & 1 & 1 & & -1 & -1 & & 1 & 1 & & -1 \\
& 1 & 1 & & -1 & -1 & & 1 & 1 & -1 & -1 & & 1 & 1 & & -1 \\
-1 & -1 & 1 & 1 & & -1 & -1 & 1 & 1 & & -1 & -1 & 1 & 1 & & -1 \\
& -1 & -1 & 1 & 1 & & -1 & & 1 & 1 & -1 & -1 & & 1 & 1 & -1 \\
1 & & -1 & -1 & 1 & 1 & -1 & -1 & & 1 & & -1 & & 1 & 1 & -1 \\
& 1 & & -1 & & 1 & & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
-1 & & 1 & & -1 & & 1 & & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
& -1 & & 1 & -1 & -1 & 1 & 1 & -1 & & 1 & & -1 & 1 & 1 & -1 \\
1 & & -1 & 1 & 1 & -1 & & & 1 & -1 & -1 & 1 & & -1 & & 1 & -1 \\
& 1 & -1 & & 1 & & -1 & & -1 & 1 & & -1 & 1 & 1 & -1 & & 1 & -1 \\
-1 & 1 & & 1 & -1 & 1 & 1 & -1 & & 1 & -1 & & 1 & -1 & & 1 & -1 \\
& -1 & & 1 & & -1 & 1 & & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 \\
1 & -1 & & & 1 & -1 & & & 1 & -1 & 1 & & -1 & 1 & & -1 & 1 & -1 \\
& 1 & -1 & 1 & & -1 & 1 & -1 & & & 1 & -1 & 1 & & -1 & 1 & -1
\end{bmatrix},$$

$$\mathbf{Q}_3 = \begin{bmatrix}
-1 & 1 & & -1 & 1 & -1 & & & 1 & -1 & 1 & & -1 & 1 & -1 & & 1 \\
1 & -1 & & & 1 & -1 & 1 & -1 & & & 1 & -1 & 1 & -1 & & 1 & -1 \\
-1 & 1 & & & & 1 & -1 & 1 & -1 & & & 1 & -1 & 1 & -1 & 1 & \\
1 & -1 & 1 & & -1 & 1 & -1 & 1 & -1 & & & & & 1 & -1 & 1 & -1 \\
-1 & 1 & & & & & 1 & -1 & 1 & -1 & 1 & -1 & & & & & 1 \\
1 & -1 & 1 & -1 & 1 & & & & -1 & 1 & -1 & 1 & -1 & 1 & & & \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & & & & & & & -1 & 1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
& & & & & & & & & & & & & & & & \\
& & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
& & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & -1 & -1 & -1 & -1 & -1 & -1 \\
& -1 & -1 & -1 & -1 & -1 & & & & 1 & 1 & 1 & 1 & 1 & & & -1 \\
& & 1 & 1 & 1 & 1 & & & -1 & -1 & -1 & -1 & & 1 & 1 & 1 & 1 \\
& & 1 & 1 & 1 & & -1 & -1 & -1 & & & 1 & 1 & 1 & & -1 & -1 & -1 \\
& & & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & & & 1 & 1 \\
& -1 & -1 & -1 & -1 & & & & 1 & 1 & 1 & & -1 & -1 & -1 & -1 & & 1
\end{bmatrix},$$

$$\mathbf{Q}_4 = \begin{bmatrix}
-1 & 1 & -1 & & 1 & -1 & 1 & & -1 & 1 & -1 & & 1 & -1 & 1 & \\
-1 & & 1 & -1 & 1 & -1 & & 1 & -1 & 1 & & & 1 & -1 & 1 & \\
1 & -1 & 1 & -1 & & 1 & -1 & 1 & -1 & & & -1 & 1 & -1 & 1 & \\
1 & & & 1 & -1 & 1 & -1 & 1 & & & 1 & -1 & 1 & -1 & 1 & \\
-1 & 1 & -1 & 1 & -1 & 1 & & & 1 & -1 & 1 & -1 & 1 & -1 & 1 & \\
-1 & 1 & & & & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & & \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & & & \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & & & & & & & & \\
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & & & & & & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\
-1 & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & -1 & -1 & -1 \\
-1 & -1 & -1 & -1 & & & 1 & 1 & 1 & 1 & 1 & 1 & & -1 & -1 & -1 \\
1 & & -1 & -1 & -1 & -1 & & & 1 & 1 & 1 & 1 & & & -1 & -1 \\
-1 & & 1 & 1 & 1 & & -1 & -1 & -1 & & 1 & 1 & 1 & & -1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & & -1 & -1 & -1 & -1 \\
1 & 1 & 1 & & -1 & -1 & -1 & -1 & & 1 & 1 & 1 & 1 & & -1 & -1
\end{bmatrix}.$$

APPENDIX D – MATRIX FACTORIZATION FROM THE DFT APPROXIMATIONS

In this appendix, we present the matrices factorizations from the DFT approximations from Chapter 6. Similarly to the factorization from Chapter 5, we consider the following butterfly-structure:

$$\mathbf{B}_N = \begin{bmatrix} \mathbf{I}_{\frac{N}{2}} & \bar{\mathbf{I}}_{\frac{N}{2}} \\ -\bar{\mathbf{I}}_{\frac{N}{2}} & \mathbf{I}_{\frac{N}{2}} \end{bmatrix},$$

where $\mathbf{I}$ is the identity matrix and $\bar{\mathbf{I}}$ is the counter-identity matrix.

- For $N = 8$, the approximations $\mathbf{T}_8$ can be represented as

$$\mathbf{T}_8 = \mathbf{A}_5 \cdot \mathbf{M}_1 \cdot \mathbf{A}_4 \cdot \mathbf{A}_3 \cdot \mathbf{A}_2 \cdot \mathbf{A}_1,$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1 & & & \\ & & & \mathbf{B}_7 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{B}_5 & & & \\ & & 1 & \\ & & & \mathbf{B}_3 \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} \mathbf{B}_3 & & & & \\ & 1 & & j & \\ & & 1 & & j \\ & -1 & & 1 & \\ & & -1 & & 1 \\ & & & & & 1 \end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} \mathbf{B}_2 & & & \\ & 1 & & j \\ & & \mathbf{I}_4 & \\ & -1 & & j \end{bmatrix}, \quad \mathbf{A}_5 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & -1 & & \\ & & 1 & & & & & \\ & & & 1 & & & & -1 \\ & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & & & & & 1 & \\ & & & 1 & & & & 1 \end{bmatrix},$$

$$\mathbf{M}_1 = \begin{bmatrix} 2 & & & & & & & \\ & & & & & & 2 & \\ & & & & & & & 2 \\ & & & & -2 & & & \\ & -2 & & & & & & \\ & & & & & -1 & & \\ & & -2 & & & & & \\ & & & -1 & & & & \end{bmatrix}.$$

- For $N = 16$, the approximation $\mathbf{T}_{16}$ can be represented as

$$\mathbf{T}_{16} = \mathbf{A}_6 \cdot \mathbf{A}_7 \cdot \mathbf{M}_1 \cdot \mathbf{A}_5 \cdot \mathbf{A}_4 \cdot \mathbf{A}_3 \cdot \mathbf{A}_2 \cdot \mathbf{A}_1,$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1 & \\ & \mathbf{B}_{15} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{B}_9 & \\ & \mathbf{B}_7 \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} \mathbf{B}_5 & & & \\ & \mathbf{I}_4 & j\cdot\mathbf{I}_4 & \\ & -\mathbf{I}_4 & j\cdot\mathbf{I}_4 & \\ & & & \mathbf{B}_3 \end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} \mathbf{B}_3 & & & \\ & \mathbf{I}_2 & j\cdot\mathbf{I}_2 & \\ & & \mathbf{I}_8 & \\ & -\mathbf{I}_2 & j\cdot\mathbf{I}_2 & \\ & & & 1 \end{bmatrix}, \quad \mathbf{A}_5 = \begin{bmatrix} \mathbf{B}_2 & & & \\ & 1 & & j \\ & & \mathbf{I}_{12} & \\ & -1 & & j \end{bmatrix},$$

$$\mathbf{A}_6 = \begin{bmatrix} 1 & & & & & & & \\ & \mathbf{I}_3 & & & -\mathbf{I}_3 & & & \\ & & 1 & & & & & \\ & & & \mathbf{I}_3 & & & & -\mathbf{I}_3 \\ & & & & 1 & & & \\ & \mathbf{I}_3 & & & & \mathbf{I}_3 & & \\ & & & & & & 1 & \\ & & & \mathbf{I}_3 & & & & \mathbf{I}_3 \end{bmatrix}, \quad \mathbf{A}_7 = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & -1 & & & \\ & & 1 & & & & & \\ & & & 1 & & & -1 & \\ & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & & & & & 1 & \\ & & & 1 & & & & 1 \\ & & & & & & & \mathbf{I}_8 \end{bmatrix},$$

$$\mathbf{M}_1 = \begin{bmatrix}
2 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & 2 & & & \\
 & & & & & & & & & & & & & 2 & & \\
 & & & & & & & & -2 & & & & & & & \\
 & & & & & & & & & & & & & & 2 & \\
 & & & & & & & & & & -1 & & & & & \\
 & & & & -2 & & & & & & & & & & & \\
 & & & & & & -1 & & & & & & & & & \\
 & -2 & & & & & & & & & & & & & & \\
 & & & & & & & & & -1 & & -2 & & & & \\
 & & & & & & & & & & & & & -1 & & \\
 & & & & & -2 & & 1 & & & & & & & & \\
 & & -2 & & & & & & & & & & & & & \\
 & & & & & & & & & -2 & & 1 & & & & \\
 & & & -1 & & & & & & & & & & & & \\
 & & & & & & -1 & & -2 & & & & & & &
\end{bmatrix}.$$

- For $N = 32$, the approximation $\mathbf{T}_{32}$ can be represented as

$$\mathbf{T}_{16} = \mathbf{A}_7 \cdot \mathbf{A}_8 \cdot \mathbf{A}_9 \cdot \mathbf{M}_1 \cdot \mathbf{A}_6 \cdot \mathbf{A}_5 \cdot \mathbf{A}_4 \cdot \mathbf{A}_3 \cdot \mathbf{A}_2 \cdot \mathbf{A}_1,$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1 & \\ & \mathbf{B}_{31} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{B}_{17} & \\ & \mathbf{B}_{15} \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} \mathbf{B}_9 & & & \\ & \mathbf{I}_8 & j \cdot \mathbf{I}_8 & \\ & -\mathbf{I}_8 & j \cdot \mathbf{I}_8 & \\ & & & \mathbf{B}_7 \end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} \mathbf{B}_5 & & & \\ & \mathbf{I}_4 & j \cdot \mathbf{I}_4 & \\ & \mathbf{I}_{16} & & \\ & -\mathbf{I}_4 & j \cdot \mathbf{I}_4 & \\ & & & \mathbf{B}_3 \end{bmatrix}, \quad \mathbf{A}_5 = \begin{bmatrix} \mathbf{B}_3 & & & \\ & \mathbf{I}_2 & j \cdot \mathbf{I}_2 & \\ & \mathbf{I}_{24} & & \\ & -\mathbf{I}_2 & j \cdot \mathbf{I}_2 & \\ & & & 1 \end{bmatrix},$$

$$\mathbf{A}_6 = \begin{bmatrix} \mathbf{B}_2 & & & \\ & 1 & & j \\ & & \mathbf{I}_{28} & \\ & -1 & & j \end{bmatrix}, \quad \mathbf{A}_7 = \begin{bmatrix} 1 & & & & & & & \\ & \mathbf{I}_7 & & & & -\mathbf{I}_7 & & \\ & & 1 & & & & & \\ & & & \mathbf{I}_7 & & & & -\mathbf{I}_7 \\ & & & & 1 & & & \\ & \mathbf{I}_7 & & & & \mathbf{I}_7 & & \\ & & & & & & 1 & \\ & & & \mathbf{I}_7 & & & & \mathbf{I}_7 \end{bmatrix},$$

$$\mathbf{A}_8 = \begin{bmatrix} 1 & & & & & & & & \\ & \mathbf{I}_3 & & & -\mathbf{I}_3 & & & & \\ & & 1 & & & & & & \\ & & & \mathbf{I}_3 & & & -\mathbf{I}_3 & & \\ & & & & 1 & & & & \\ & \mathbf{I}_3 & & & \mathbf{I}_3 & & & & \\ & & & & & 1 & & & \\ & & & \mathbf{I}_3 & & & \mathbf{I}_3 & & \\ & & & & & & & \mathbf{I}_{16} \end{bmatrix}, \quad \mathbf{A}_9 = \begin{bmatrix} 1 & & & & & & & & \\ & 1 & & & & -1 & & & \\ & & 1 & & & & & & \\ & & & 1 & & & & -1 & \\ & & & & 1 & & & & \\ & 1 & & & & 1 & & & \\ & & & & & & 1 & & \\ & & & 1 & & & & 1 & \\ & & & & & & & & \mathbf{I}_{24} \end{bmatrix},$$

- For $N = 64$, the approximation $\mathbf{T}_{64}$ can be represented as

$$\mathbf{T}_{64} = \mathbf{A}_8 \cdot \mathbf{A}_9 \cdot \mathbf{A}_{10} \cdot \mathbf{A}_{11} \cdot \mathbf{M}_1 \cdot \mathbf{A}_7 \cdot \mathbf{A}_6 \cdot \mathbf{A}_5 \cdot \mathbf{A}_4 \cdot \mathbf{A}_3 \cdot \mathbf{A}_2 \cdot \mathbf{A}_1,$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1 & \\ & \mathbf{B}_{63} \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} \mathbf{B}_{33} & \\ & \mathbf{B}_{31} \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} \mathbf{B}_{17} & & & \\ & \mathbf{I}_{16} & j \cdot \mathbf{I}_{16} & \\ & -\mathbf{I}_{16} & j \cdot \mathbf{I}_{16} & \\ & & & \mathbf{B}_{15} \end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} \mathbf{B}_9 & & & \\ & \mathbf{I}_8 & j \cdot \mathbf{I}_8 & \\ & \mathbf{I}_{32} & & \\ & -\mathbf{I}_8 & j \cdot \mathbf{I}_8 & \\ & & & \mathbf{B}_7 \end{bmatrix}, \quad \mathbf{A}_5 = \begin{bmatrix} \mathbf{B}_5 & & & \\ & \mathbf{I}_4 & j \cdot \mathbf{I}_4 & \\ & \mathbf{I}_{48} & & \\ & -\mathbf{I}_4 & j \cdot \mathbf{I}_4 & \\ & & & \mathbf{B}_3 \end{bmatrix},$$

$$\mathbf{M}_1 = \left[\begin{array}{cccccccc|cccccccc}
2 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & 2 & \\
 & & & & & & & & & & & & & 2 & & \\
 & & & & & & & & -2 & & & & & & & \\
 & & & & & & & & & & & & 2 & & & \\
 & & & -2 & & & & & & & & -1 & & & & \\
 & & & & & & & & & & & & & & & 2 \\
 & & & & & & & & & -1 & & -2 & & & & \\
 & & & & & & & & & & & & & -1 & & \\
 & & & & & -2 & & 1 & & & & & & & & \\
 & & -2 & & & & & & & & & & & & & \\
 & & & & & & & & & -2 & & 1 & & & & \\
 & -1 & & & & & & & & & & & & & & \\
 & & & & & -1 & & -2 & & & & & & & & \\
-2 & & & & & & & & & & & & & & & \\
 & & & & & & & & & & -1 & & -1 & -2 & & \\
 & & & & & & & & & & & & & -1 & -2 & \\
 & & & & -1 & & -2 & & 1 & & & & & & & \\
 & & & & & & & & & & & & & & & -1 \\
 & & & & & & & & & -1 & & 2 & -1 & & & \\
 & & -2 & & 1 & & & & & & & & & & & \\
 & & & & & -2 & & 1 & -1 & & & & & & & \\
-2 & & & & & & & & & & & & & & & \\
 & & & & & & & & & -2 & & 1 & -1 & & & \\
 & & & & & & & & & & & & & -2 & 1 & \\
 & & & & -1 & & & & 2 & -1 & & & & & & \\
-1 & & & & & & & & & & & & & & & \\
 & & & & & & & & & -1 & & -2 & & 1 & & \\
 & & -1 & & -2 & & & & & & & & & & & \\
 & & & & & -1 & & -1 & -2 & & & & & & &
\end{array}\right].$$

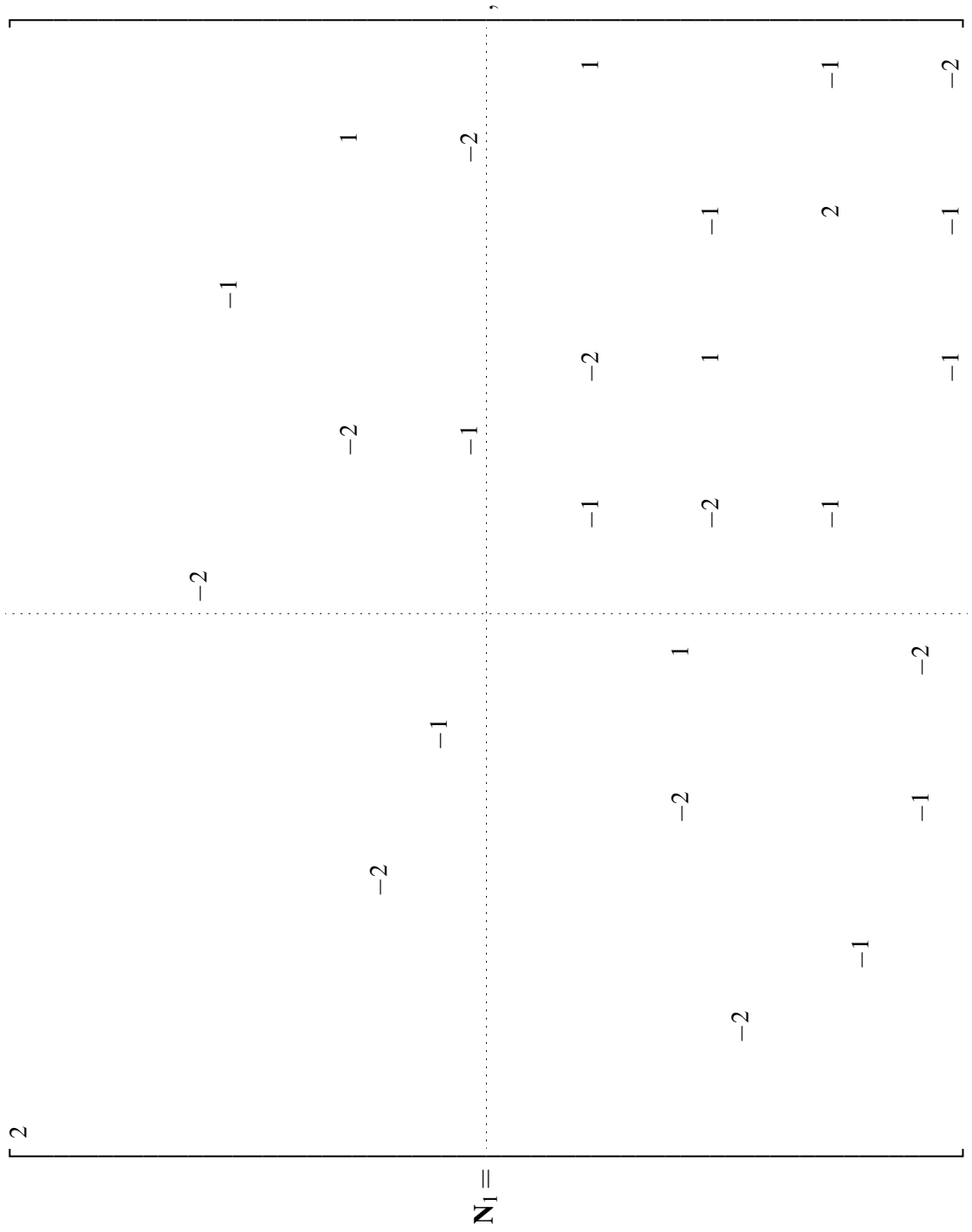
$$\mathbf{A}_6 = \begin{bmatrix} \mathbf{B}_3 & & & \\ & \mathbf{I}_2 & j\cdot\mathbf{I}_2 & \\ & \mathbf{I}_{28} & & \\ & -\mathbf{I}_2 & j\cdot\mathbf{I}_2 & \\ & & & 1 \end{bmatrix}, \quad \mathbf{A}_7 = \begin{bmatrix} \mathbf{B}_2 & & \\ & 1 & j \\ & \mathbf{I}_{60} & \\ & -1 & j \end{bmatrix},$$

$$\mathbf{A}_8 = \begin{bmatrix} 1 & & & & & & & \\ & \mathbf{I}_{15} & & -\mathbf{I}_{15} & & & & \\ & & 1 & & & & & \\ & & & \mathbf{I}_{15} & & -\mathbf{I}_{15} & & \\ & & & & 1 & & & \\ & \mathbf{I}_{15} & & & & \mathbf{I}_{15} & & \\ & & & & & & 1 & \\ & & & \mathbf{I}_{15} & & & & \mathbf{I}_{15} \end{bmatrix}, \quad \mathbf{A}_9 = \begin{bmatrix} 1 & & & & & & & \\ & \mathbf{I}_7 & & -\mathbf{I}_7 & & & & \\ & & 1 & & & & & \\ & & & \mathbf{I}_7 & & -\mathbf{I}_7 & & \\ & & & & 1 & & & \\ & \mathbf{I}_7 & & & & \mathbf{I}_7 & & \\ & & & & & & 1 & \\ & & & \mathbf{I}_7 & & & & \mathbf{I}_7 \end{bmatrix}$$

$$\mathbf{A}_{10} = \begin{bmatrix} 1 & & & & & & & \\ & \mathbf{I}_3 & & -\mathbf{I}_3 & & & & \\ & & 1 & & & & & \\ & & & \mathbf{I}_3 & & -\mathbf{I}_3 & & \\ & & & & 1 & & & \\ & \mathbf{I}_3 & & & & \mathbf{I}_3 & & \\ & & & & & & 1 & \\ & & & \mathbf{I}_3 & & & & \mathbf{I}_3 \end{bmatrix}, \quad \mathbf{A}_{11} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & -1 & & & & \\ & & 1 & & & & & \\ & & & 1 & & -1 & & \\ & & & & 1 & & & \\ & 1 & & & & 1 & & \\ & & & & & & 1 & \\ & & & 1 & & & & 1 \end{bmatrix},$$

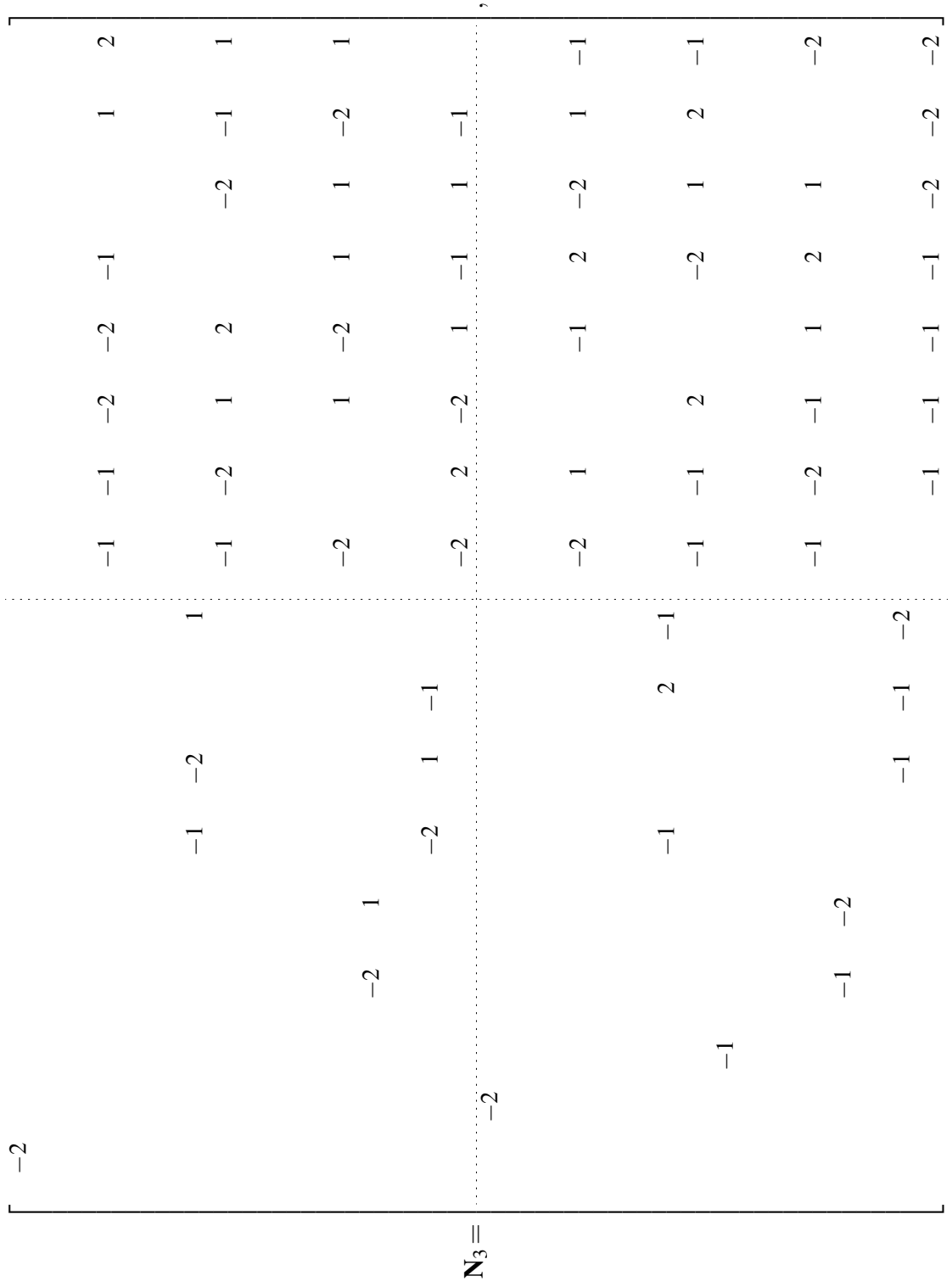
$$\mathbf{M}_1 = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 \\ \mathbf{N}_3 & \mathbf{N}_4 \end{bmatrix},$$

being



$$\mathbf{N}_1 = \left[\begin{array}{cccccccccccccccc|cccccccccccccccc}
2 &&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&& -2 &&&&&&&&&& &&&&& \\
&&&&&&&&&& &&&&&&&&&& &&&& -1 &&&&&&& \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& -2 &&&&&&&& 1 &&& \\
&&&&&&&& -2 &&&&&&&&&& &&&&&&&&&& &&& \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& && -1 &&&&&&&&&& &&&&&&&&& \\
&&&&&&&&&& &&&&&&&&&& -1 &&&&&&&& -2 &&& \\
\hline
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&& -1 &&&& -2 &&&&&&&& 1 & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& -2 &&&& 1 &&&&&&&&&& &&&&&&& \\
&&&&&&&&&& &&&&&&&& -2 &&&& 1 &&&& -1 &&&&& \\
&&&& -2 &&&&&&&&&& &&&&&&&&&& &&&&&&& \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& &&&&&&&& -1 &&&&&&&& 2 &&&& -1 & \\
&&&&&& -1 &&&&&&&&&& &&&&&&&&&& &&&&& \\
&&&&&&&&&& &&&&&&&&&& &&&&&&&&&& & \\
&&&&&&&&&& -1 &&&& -2 &&&&&&&&&& &&&&&&& \\
&&&&&&&&&& &&&&&&&&&& && -1 &&&& -1 &&&& -2 &
\end{array}\right],$$

$$\mathbf{N}_2 = \left[\begin{array}{cccc|cccc}
-2 & & & & 2 & & & \\
 & -1 & & & & 2 & & \\
 & & -1 & & & & 2 & \\
 & & & -2 & & & & 2 \\
\hline
 & -1 & & & -1 & & & \\
 & & -2 & & & -2 & & \\
 & & & 1 & & & -1 & \\
\hline
 & & -1 & -1 & -2 & & & \\
 & -1 & & 2 & -1 & & & \\
 & -2 & 1 & -1 & & & & \\
 & -1 & -2 & & 1 & & & \\
\hline
 & & & & & -1 & -2 & \\
 & & & & & -2 & 1 & \\
 & & & & & & & 2
\end{array}\right],$$


N3 =

$$
\mathbf{N}_4 = \left[\begin{array}{cccccccc:cccccc}
 & -1 & -1 & -1 & -1 & -2 & -2 & -2 & & -1 & -1 & -2 & & \\
 & & & & & & & & & & & & -1 & -2 \\
-1 & -2 & -1 & 1 & 2 & 1 & & -2 & & & & & & \\
 & & & & & & & & & & & & & -1 \\
-1 & -1 & 2 & & -2 & 1 & 2 & -1 & & & & & & \\
 & & & & & & & & -1 & & 2 & -1 & & \\
-2 & 1 & & -1 & 2 & -2 & 1 & -1 & & & & & & \\
\hdashline
-2 & 2 & -2 & 1 & -1 & 1 & -1 & & & & & & & \\
 & & & & & & & & -2 & 1 & -1 & & & \\
 & & & & & & & & & & & & -2 & 1 \\
-2 & & 1 & -2 & 1 & 1 & -2 & 1 & & & & & & \\
 & & & & & & & & & & & & & \\
-1 & -2 & 1 & 2 & & -2 & -1 & 1 & & & & & & \\
 & & & & & & & & -1 & -2 & & 1 & & \\
 & & & & & & & & & & & & & \\
-1 & -1 & -2 & -2 & -1 & & 1 & 2 & & & & & & \\
 & & & & & & & & & & & & &
\end{array}\right].
$$