UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ELETRÔNICA E SISTEMAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

RAÍZA DOS SANTOS OLIVEIRA

PROJETO E IMPLEMENTAÇÃO DE TRANSFORMADAS DISCRETAS DE BAIXA COMPLEXIDADE PARA CODIFICAÇÃO DE IMAGENS E VÍDEOS

Recife
2018

RAÍZA DOS SANTOS OLIVEIRA

PROJETO E IMPLEMENTAÇÃO DE TRANSFORMADAS DISCRETAS DE BAIXA COMPLEXIDADE PARA CODIFICAÇÃO DE IMAGENS E VÍDEOS

Dissertação submetida ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de Mestre em Engenharia Elétrica

Área de concentração: Comunicações.

Orientador: Prof. Dr. Renato José de Sobral Cintra.
Coorientador: Prof. Dr. Fábio M. Bayer.

Recife
2018

Catalogação na fonte
Bibliotecário Josias Machado, CRB-4 / 1690

O048p Oliveira, Raíza dos Santos.
Projeto e implementação de transformadas discretas de baixa complexidade para codificação de imagens e vídeos / Raíza dos Santos Oliveira. – Recife, 2018.
103 f., il., figs., tabs.

Orientador: Prof. Dr. Renato José de Sobral Cintra.
Coorientador: Prof. Dr. Fábio M. Bayer.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2018.
Inclui Referências e Apêndices.

1. Engenharia Elétrica. 2. DCT. 3. Transformadas discretas. 4. Aproximações matriciais. I. Cintra, Renato José de Sobral (Orientador). II. Bayer, Fábio M. (Coorientador). III. Título.

UFPE

621.3 CDD (22. ed.) BCTG/2019-227

RAÍZA DOS SANTOS OLIVEIRA

PROJETO E IMPLEMENTAÇÃO DE TRANSFORMADAS DISCRETAS DE BAIXA COMPLEXIDADE PARA CODIFICAÇÃO DE IMAGENS E VÍDEOS

Dissertação submetida ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do grau de Mestre em Engenharia Elétrica

Aprovada em: 27/07/2018

BANCA EXAMINADORA

_____

Prof. Dr. Ricardo Menezes Campello de Souza (Examinador Interno)
Universidade Federal de Pernambuco

_____

Prof. Dr. Hélio Magalhães de Oliveira (Examinador Externo)
Universidade Federal de Pernambuco

_____

Prof. Dr. André Leite Wanderley (Examinador Externo)
Universidade Federal de Pernambuco

A Abel, com amor.

AGRADECIMENTOS

A Abel, meu esposo, cujo amor, cumplicidade e paciência me salvaram mais vezes do que consigo contar.

À minha família. Em particular, a tia Sheila e Renata, por me acolherem e cuidarem de mim. Sem o suporte de vocês eu não teria chegado até aqui.

A seu Abel, tia Bibi, Debinha e Tiago. Os conceitos de amor e cuidado nunca ficam tão claros para mim quanto quando estou com vocês.

Aos amigos da UFPE, porque sofrer em conjunto torna a carga mais leve. Em especial, a Renan, por estar sempre presente mesmo estando distante. A Bruna Palm e João Eudes, pela amizade tão sincera.

Aos amigos da In Loco. Em especial, agradeço a Rodrigo Paiva, cujos ensinamentos diários levarei comigo para sempre, a Zé Luciano e Yohanna pela parceria constante (melhor time <3), a Luis Galvão por me acolher tão carinhosamente, e a Denys, Thiago Rodrigues e agregados dos cozidos de quinta–feira, pelas discussões instigantes.

Agradeço aos meus professores. Primeiramente ao meu orientador, Prof. Renato Cintra, pela orientação presente e pelo esforço constante para que eu me tornasse a profissional que posso ser. Ao Prof. Fábio M. Bayer, pela co-orientação. Ao Prof. André Leite pela colaboração na implementação dos códigos para este trabalho. Ao Prof. Ricardo Campello, por ministrar um dos melhores cursos que já fiz na vida. E ao Prof. Arjuna Madanayake, pela colaboração nas publicações.

Agradeço também aos colegas Vítor Coutinho, Thiago L. da Silveira, Paulo Oliveira e Diego Coelho pela colaboração em pesquisas e trabalhos publicados.

A FACEPE, pelo auxílio financeiro.

RESUMO

O custo computacional da implementação de transformadas discretas pode ser significativo quando se considera a enorme quantidade de dados que as tecnologias contemporâneas exigem e/ou a demanda por dispositivos de baixa potência. O uso de algoritmos rápidos reduz os custos aritméticos de computação das transformadas e o consumo de energia sem eliminar a necessidade por aritmética em ponto flutuante. Neste sentido, as aproximações matriciais de baixa complexidade são uma alternativa para o cômputo das transformadas. Neste trabalho, é introduzido um método baseado em uma heurística gulosa e na distância angular entre vetores para obtenção de aproximações matriciais. Introduzimos metodologias para a aplicação efetiva do método proposto para aproximar as matrizes das transformadas discretas de Fourier, Hartley e do cosseno (DCT). O método é utilizado para obtenção de novas aproximações para a DCT de comprimento 8. Treze novas aproximações foram obtidas, das quais cinco apresentam resultados melhores que os da DCT em termos do índice de similaridade estrutural em experimentos de compressão de imagens. Uma das aproximações obtidas foi selecionada para análises mais aprofundadas. Aproximações de comprimentos 16 e 32 para as simulações de vídeo foram obtidas escalando, por meio do algoritmo de Jridi-Alfalou-Meher, a aproximação de comprimento 8 selecionada. O codec de vídeo utilizando as aproximações propostas apresentou resultados muito próximos aos do codec original, tendo uma perda máxima de 0.55dB nos testes realizados. Para a aproximação selecionada, foi também realizada a implementação em FPGA. Quando comparada à implementação de outras aproximações da literatura, a implementação da transformada proposta mostrou capacidade de operar numa frequência até 19% maior.

Palavras-chave: DCT, transformadas discretas, aproximações matriciais

ABSTRACT

The computational cost of implementing discrete transforms can be significant when considering the massive amount of data that contemporary technologies require and/or the demand for low–power devices. The use of fast algorithms substantially reduces arithmetic costs without eliminating the need of floating-point arithmetic. In this sense, low-complexity matrix approximations appear as an alternative way to compute these transforms. In this work, a greedy algorithm based on the angular distance between vectors for obtaining low-complexity approximations from a given matrix is proposed. We introduce methodologies for the effective application of the proposed method to approximate the discrete Fourier, Hartley, and cosine (DCT) transforms. The method is employed to derive new approximations for the 8-point DCT. Thirteen new approximations for the 8-point DCT were obtained; five of them outperformed the DCT in terms of the structural similarity index on the image compression experiments. One of the proposed approximations was chosen for further analysis. Approximations for the 16- and 32-point DCT were derived by means of the Jridi–Alfalou–Meher scaling method based on the previously selected 8-point approximation. Such scaled matrices were submitted to video experiments. The encoded video resulted from the approximate transforms performed very closely to the standard video encoding: the maximum loss was 0.55dB in video compression experiments. The selected approximation was also implemented on a FPGA. When compared to implementations of other two approximations in literature, the proposed method was shown to be able to operate at a 19% higher frequency.

Keywords:DCT, discrete transforms, matrix approximations

LIST OF FIGURES

LIST OF TABLES

LIST OF ABREVIATIONS

DFT	Discrete Fourier transform
DHT	Discrete Hertley transform
DCT	Discrete cosine transform
JPEG	Joint photographic experts group
MPEG	Moving picture experts group
HEVC	High efficiency video coding
KLT	Karhunen–Loève transform
HDTV	High-definition TV
FPGA	Field programmable gate array
WHT	Walsh–Hadamard transform
SDCT	Signed DCT
BAS	Bouguezel–Ahmad–Swamy
RDCT	Rounded DCT
MRDCT	Modified rounded DCT
CBT	Cintra–Bayer–Tablada
IDCT	Interger DCT
MSE	Mean squared error
PSNR	Peak signal–to–noise ratio
SSIM	Structural similarity index

RDiff	Relative difference
SFG	Signal flow graph
JAM	Jridi–Alfalou–Meher
CTC	Common test conditions
AI	All Intra
RA	Random acess
LD-B	Low delay B
LD-P	Low delay P
YUV	A color encoding system that defines color space in terms of one luma (Y) and two chrominance (UV) components
RD	Rate distortion
QP	Quantization parameter
BD-PSNR	Bjøntegaard's delta PSNR
BD-Rate	Bjøntegaard's delta rate
JTAG	Joint test action group
CLB	Configurable logic blocks
FF	Flip–flop

LIST OF SYMBOLS

$\mathbf{x}$	An N-dimensional input vector
$\mathbf{X}$	An N-dimensional transformed vector
j	Imaginary unit, $j = \sqrt{-1}$
$\mathbf{F}_N$	Matrix representation of the N-point DFT
$X_k^{Fourier}$	The kth coefficient of the DFT spectrum
$X_k^{Hartley}$	The kth coefficient of the DHT spectrum
$\mathbf{H}_N$	Matrix representation of the N-point DHT
$\mathbf{W}$	KLT matrix
$\mathbf{y}$	Output vector of the KLT transformation
$\mathbf{R_y}$	Covariance matrix of $\mathbf{y}$
$\mathbf{R_x}$	Covariance matrix of $\mathbf{x}$
ρ	Correlation coefficient
z_m	White noise process
$\mathbf{C}_N$	Matrix representation of the N-point DCT
$\mathbf{A}$	An arbitrary matrix
$\widehat{\mathbf{C}}_N$	Matrix representation of an approximation for the N-point DCT
$\hat{\mathbf{X}}$	Output vector of the transformation of $\mathbf{x}$ using an approximate transform
$\mathbf{T}$	Low-complexity multiplierless matrix
$\mathbf{D}$	Diagonal matrix used in the orthonormalization process

$\mathbf{a}_k$	The kth row of an arbitrary matrix $\mathbf{A}$
$\mathbf{t}_k$	The kth row of a low-complexity matrix $\mathbf{T}$
$\mathcal{P}$	Set of low-complexity elements
$\mathcal{D}_{\mathcal{P}}$	Search space generated from the set $\mathcal{P}$
$\mathcal{D}_{\mathcal{P}}^{(k)}$	Subset of the search space containing all the solution in $\mathcal{D}_{\mathcal{P}}$ for the optimization problem in Equation (4.3) for the kth row of the input matrix
$\wp_m$	The mth search sequence
$\epsilon(\cdot)$	Total error energy
$\mathrm{MSE}(\cdot)$	Mean square error
$C_g(\cdot)$	Coding gain
$\eta(\cdot)$	Transform efficiency
$C_g^*(\cdot)$	Unified coding gain
$\delta(\cdot)$	Orthogonality deviation
$\mathbf{T}_{y,z}$	Representative approximation of equivalence class z obtained using approximation Scheme y

TABLE OF CONTENTS

1 INTRODUCTION

1.1 MOTIVATION AND FRAMEWORK

A signal might be seen as a function that changes with time and/or space and transmits information about the behavior of the phenomenon under study (PRIEMER, 1990; MOURA, 2009). The *IEEE Transactions on Signal Processing* Internet page states that the "term 'signal' includes, among others, audio, video, speech, image, communication, geophysical, sonar, radar, medical and musical signals" (IEEE TRANSACTIONS ON SIGNAL PROCESSING,).

The field of signal processing comprises, among other things, a collection of techniques to obtain, manipulate, analyze, represent, transmit and extract information from an input signal (MOURA, 2009). In particular, transforms play an important role in this area of research. The use of transforms allow us to look at data from a different perspective, the transform domain, which often adds new interpretations to the data under analysis. For example, the Fourier transform decomposes an input signal into its frequency components and the Karhunen–Loève transform is capable of decorrelating data sequences (BRITANAK; YIP; RAO, 2007).

Among the possible transforms, the ones with sinusoidal kernels are particularly important (CINTRA, 2011). Special interest is given to the discrete transforms, because they are suitable for real-world applications using digital computers which are inherently capable of discrete, finite calculations only (BLAHUT, 2010). In this work, we separate three discrete transforms for analysis: the discrete Fourier transform (DFT), the discrete Hartley transform (DHT), and the discrete cosine transform (DCT).

The DFT is one of the most important discrete transforms (STRANG, 1994). It finds application in many different problems such as solving difference equations (HELMS, 1967), image processing (REDDY; CHATTERJI, 1996; GONZALEZ; WOODS, 2012), beamforming (GODARA, 1995; SEYDNEJAD; AKHZARI, 2016), analysis of radar signals (CHENG et al., 2016; SAPONARA; NERI, 2017), voice processing (KLATT; KLATT, 1990), time series (RANSOM; EIKENBERRY; MIDDLEDITCH, 2002; PERERA et al., 2015; CHEN; CHEN, 2014), spectral

estimation (KAY, 1993), harmonic regression (BÁRTFAI, 2016), and analysis of biomedical signals (FITZKE et al., 1997).

The DHT, introduced by Bracewell in 1983 (BRACEWELL, 1983), is also a relevant discrete transform (POULARIKAS, 2010). The DHT is an attractive discrete transformation mainly due to the following properties: (i) the DHT is isomorphic to the DFT (BRACEWELL, 1983); (ii) the multiplicative complexities of the DHT and the DFT are identical in the sense discussed in Heideman (HEIDEMAN; BURRUS, 1988); (iii) unlike the DFT, the DHT is a purely real-valued transform, which means that it does not require complex arithmetic for its computation (BRACEWELL, 1983); (iv) the forward and inverse transforms are the same; and (v) the DHT is more symmetric than other transforms (more symmetric than the DCT, for example), which facilitates its computation and implementation (BRACEWELL, 2000). Because of its similarities with the DFT, the DHT is also applied in many different fields of study. Some examples are: optics (VILLASENOR, 1994); image processing (TSENG; LEE, 2014; KASBAN, 2017); convolution computation (DUHAMEL; VETTERLI, 1987; PEI; JAW, 1989); audio processing (JLEED; BOUCHARD, 2017); biomedical image analysis (SHRUTHI et al., 2016); and solution of power system problems (HEYDT et al., 1991).

The DCT is applied, for example, in areas such as image processing (ZHANG; WU; MA, 2016; CAO et al., 2015; KOZHEMIAKIN et al., 2014), audio processing (NASSAR et al., 2016), watermarking (RAM, 2013; LEI et al., 2016), and gait recognition (FAN et al., 2016). However, its most popular use is in data compression (BRITANAK; YIP; RAO, 2007). In particular, the DCT is applied in several image and video compression patterns, such as the JPEG (WALLACE, 1992), MPEG (GALL, 1992), H.261 (International Telecommunication Union, 1990), H.263 (International Telecommunication Union, 1995), H.264/AVC (LUTHRA; SULLIVAN; WIEGAND, 2003), and HEVC (POURAZAD et al., 2012). The good performance of the DCT for data compression can be justified by the fact the the DCT is asymptotically equivalent to the Karhunen–Loève transform (KLT), which is the optimal transform for data compression, when the input signal has some specific features (BRITANAK; YIP; RAO, 2007).

Although these transforms are very popular, implementing them requires floating-point

arithmetic. Fast algorithms can dramatically reduce their computational cost. However, the number of calls in applications of these transforms can be extraordinarily high. For instance, a single image frame of high-definition TV (HDTV), that can be encoded with the DCT, contains 32.400 8×8 image subblocks. Therefore, computational savings in the transformation step may effect significant performance gains, both in terms of speed and power consumption (POTLURI et al., 2014; COUTINHO et al., 2015). One approach to further minimize the computational cost of computing the discrete transforms is the use of matrix approximations (BAYER; CINTRA, 2010; CINTRA; BAYER; TABLADA, 2014). Such approximations provide matrices with similar mathematical behavior to the exact transform while presenting a dramatically low arithmetic cost.

1.2 GOALS

In the sense of the previous discussion, our goals in this dissertation are:

- Introducing a greedy search algorithm for matrix approximation based on angular distance between vectors;
- Discussing how the proposed method can be applied to derive approximations for trigonometric discrete transforms;
- Applying the proposed algorithm to introduce new low-complexity approximations for the 8-point DCT;
- Assessing the efficiency of the proposed approximations on image and video compression experiments when compared to the exact DCT and other approximations in literature.

1.3 STRUCTURE

The present work is structured as follows. In Chapter 2, we present the discrete transforms discussed in this dissertation: the DFT, DHT, and DCT. An overview of the mathematical structure of these transforms is provided.

In Chapter 3, we present some popular fast algorithms and low-complexity approximations for the DCT. Such low-complexity approximations shown in this chapter are considered for comparison with the methods proposed in this work.

The search algorithm for matrix approximation is detailed in Chapter 4. The proposed method is based on an unconstrained optimization problem. Considering the orthogonality property, we derive a constrained optimization problem as well. We also discuss how the proposed approach can be tailored to obtain approximations for complex-valued matrices, such as the DFT.

Considering symmetries and redundancies of a given exact matrix, we show how to reduce the computational cost of the proposed approximation algorithm. In Chapter 5, we explore the structure of the DFT, DHT and DCT, and define approximations schemes based on the combination of procedures to reduce the computational cost of the algorithm and version of the proposed method used.

In Chapter 6, the approximation schemes defined in Chapter 5 are used to find new approximations for the 8-point DCT. The proposed approximations are evaluated according to popular figures of merit and compared to the exact DCT and other approximations in literature.

In Chapter 7, a JPEG-like experiment for image compression is described and used to evaluate the performance of the proposed approximations in comparison to the DCT and the other approximations in literature.

In Chapter 8, one of the proposed approximations that presented good results in the image compression experiments is select for further analysis. A fast algorithm for the chosen approximation is introduced. The 16- and 32-point scaled versions of the selected approximation, obtained by the Jridi-Alfalou-Meher method (JRIDI; ALFALOU; MEHER, 2015), are presented. The video compression experiment is described and performed. The FPGA implementation of the selected approximation is presented along with the implementation of two other approximations in literature for comparison.

In Chapter 9, an overview of the topics discussed and obtained results is presented.

2 DISCRETE TRIGONOMETRIC TRANSFORMS

Two N-dimensional vectors, say $\mathbf{x} = \begin{bmatrix} x_0 & x_1 & \dots & x_{N-1} \end{bmatrix}^\top$ and $\mathbf{X} = \begin{bmatrix} X_0 & X_1 & \dots & X_{N-1} \end{bmatrix}^\top$, relate to each other through a discrete transform according to the following expressions:

$$X_k \triangleq \sum_{i=0}^{N-1} x_i \cdot \ker(i,k,N), \quad k = 0, 1, \dots, N-1, \tag{2.1}$$

$$x_i = \sum_{k=0}^{N-1} X_k \cdot \ker^{-1}(i,k,N), \quad i = 0, 1, \dots, N-1, \tag{2.2}$$

where $\ker(\cdot,\cdot,\cdot)$ and $\ker^{-1}(\cdot,\cdot,\cdot)$ are the forward and inverse transformation kernels. In this work, although our main goal is to propose new approximations for the DCT, we also discuss the DFT and the DHT, which are related transforms. In the following, we present a brief mathematical overview of the DFT, DHT, and DCT.

2.1 DISCRETE FOURIER TRANSFORM

The N-point DFT has its coefficients defined as in Equation (2.1) with its kernel given by

$$\ker(i,k,N) \triangleq \cos\left(\frac{2\pi ik}{N}\right) - j\sin\left(\frac{2\pi ik}{N}\right) = e^{-j2\pi ik/N}, \quad i,k = 0, 1, \dots, N-1. \tag{2.3}$$

Its inverse kernel is furnished by

$$\ker^{-1}(i,k,N) = \frac{1}{N}\left[\cos\left(\frac{2\pi ik}{N}\right) + j\sin\left(\frac{2\pi ik}{N}\right)\right] = \frac{1}{N}e^{j2\pi ik/N}, \quad i,k = 0, 1, \dots, N-1,$$

where $j = \sqrt{-1}$.

2.1.1 Matrix representation of the DFT

The DFT of an input signal of length N can be calculated by a matrix operation as

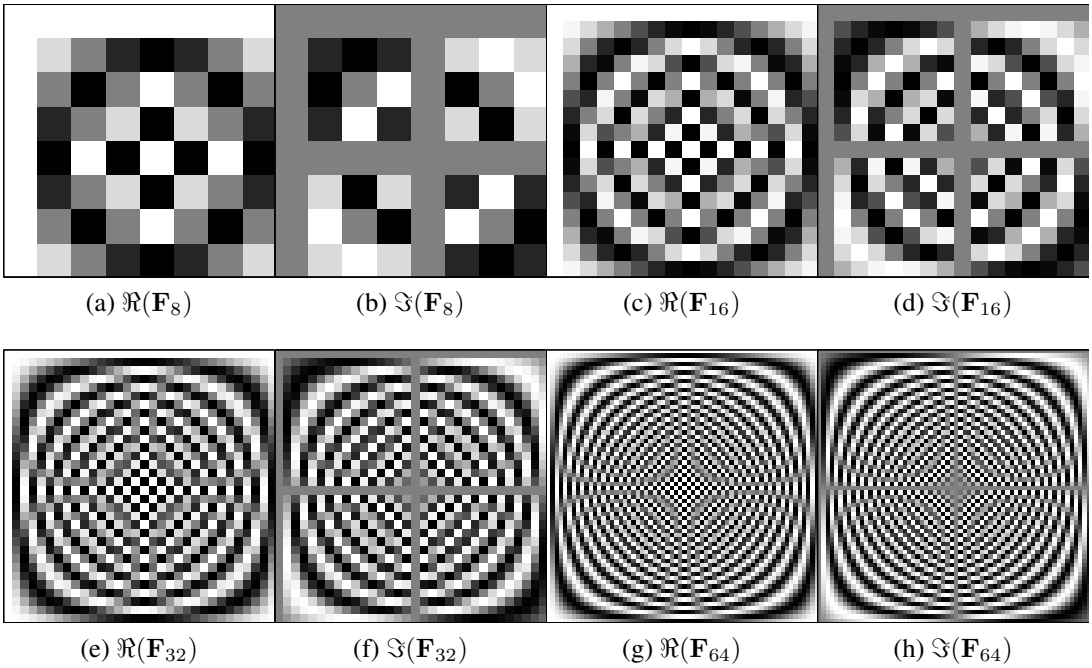
$$\mathbf{X} = \mathbf{F}_N \cdot \mathbf{x}, \tag{2.4}$$

where $\mathbf{F}_N$ is the DFT matrix given by

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

and $\omega_N \triangleq e^{-j2\pi/N}$. The matrix $\mathbf{F}_N$ is orthogonal. Then, we have that $\mathbf{F}_N^{-1} = \frac{1}{N}\mathbf{F}_N^*$, where $\mathbf{F}_N^*$ is the Hermitian matrix of $\mathbf{F}_N$ (SEBER, 2008).

Figure 1 displays, for some values of N, the image representation of the real and complex parts of $\mathbf{F}_N$. The darker shades of gray represent smaller values, whereas lighter shades represent larger values. This kind of representation is useful to visualize patterns and symmetries in the matrix, which can be explored to simplify computations and identify redundancies in the calculation of Equation (2.4).



(a) $\Re(\mathbf{F}_8)$ (b) $\Im(\mathbf{F}_8)$ (c) $\Re(\mathbf{F}_{16})$ (d) $\Im(\mathbf{F}_{16})$

(e) $\Re(\mathbf{F}_{32})$ (f) $\Im(\mathbf{F}_{32})$ (g) $\Re(\mathbf{F}_{64})$ (h) $\Im(\mathbf{F}_{64})$

Figura 1 – Image representation of the real and imaginary parts of the DFT matrix considering $N = 8, 16, 32, 64$. The functions $\Re(\cdot)$ and $\Im(\cdot)$ return the real and complex parts of its arguments, respectively.

2.1.2 Computational complexity

To calculate the DFT coefficients (Equation (2.4)), it is necessary to perform at most N^2 complex multiplications and $N(N-1)$ complex additions (BLAHUT, 2010). The complex multiplication, $(e+jf) = (a+jb)\cdot(c+jd)$, can be expressed in terms of real multiplications and real additions as (BLAHUT, 2010)

$$e = ac - bd, \tag{2.5}$$

$$f = ad + bc, \tag{2.6}$$

which requires four real multiplications and 2 real additions. As an alternative, e and f can be obtained as

$$e = (a-b)d + a(c-d), \tag{2.7}$$

$$f = (a-b)d + b(c+d), \tag{2.8}$$

whenever the multiplication operation is more computationally expensive than the addition operation. In this case, the complex product requires three real multiplications and five real additions. Besides that, if c and d are constants for a series of complex multiplications, for example, when transforming a series of input vectors using Equation (2.4), then the terms $c+d$ and $c-d$ are also constants and can be previously computed. By doing so, the computation cost becomes three real multiplications and three real additions. Each complex addition requires two real additions.

Therefore, if considering the last option described for the computation of the complex product, the direct computation of all the DFT coefficients requires at most $3N^2$ real multiplications and $N(5N-2)$ real additions.

2.2 DISCRETE HARTLEY TRANSFORM

The DHT forward and inverse kernels are given by:

$$\ker(i,k,N) \triangleq \frac{1}{N}\operatorname{cas}\left(\frac{2\pi ik}{N}\right),\ k = 0,1,\ldots,N-1, \tag{2.9}$$

$$\ker^{-1}(i,k,N) = \operatorname{cas}\left(\frac{2\pi ik}{N}\right),\ i = 0,1,\ldots,N-1.$$

where $\mathrm{cas}(x) \triangleq \cos(x) + \sin(x)$ (BRACEWELL, 1983).

Notice that $\cos(x)$ is an even function and $\sin(x)$ is an odd function (OPPENHEIM, 1999). Therefore, we can say that the even part of the DHT is the part that corresponds to the cosine function and the odd part of the DHT is the part that corresponds to the sine function.

The DFT and DHT relate to each other through a very simple expression. Let $X_k^{Fourier}$ and $X_k^{Hartley}$ be kth coefficient of the DFT and DHT spectrum, respectively, computed from $\mathbf{x}$ according to Equation (2.1). Then, the DHT coefficients are calculated in terms of the DFT coefficients as follows

$$X_k^{Hartley} = \Re(X_k^{Fourier}) - \Im(X_k^{Fourier}), \tag{2.10}$$

and conversely

$$\Re(X_k^{Fourier}) = \frac{1}{2}(X_{N-k}^{Hartley} + X_k^{Hartley}) \text{ and } \Im(X_k^{Fourier}) = \frac{1}{2}(X_{N-k}^{Hartley} - X_k^{Hartley}). \tag{2.11}$$

The matrix representation of the DHT is naturally derived from Equation (2.10) as (POULARIKAS, 2010)

$$\mathbf{H}_N = \Re(\mathbf{F}_N) - \Im(\mathbf{F}_N).$$

On the other hand, the DFT can be obtained from the DHT as follows (POULARIKAS, 2010)

$$\Re(\mathbf{F}_N) = \mathcal{E}(\mathbf{H}_N) \text{ and } \Im(\mathbf{F}_N) = \mathcal{O}(\mathbf{H}_N),$$

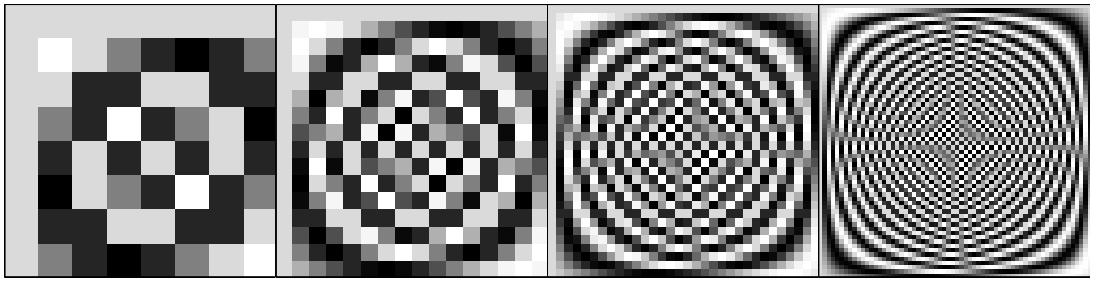
where $\mathcal{E}(\cdot)$ and $\mathcal{O}(\cdot)$ return the even and odd parts of its input, respectively. When applied to matrices, $\mathcal{E}(\cdot)$ and $\mathcal{O}(\cdot)$ act elementwise.

The image representations of the DHT for $N = 8, 16, 32, 64$ are shown in Figure 2. As expected, because the DFT and the DHT share similar mathematical definitions, the image patterns shown in Figures 2 and 1 are comparable. This is more evident for larger values of N.

2.2.1 Computational complexity

To transform an input signal, $\mathbf{x}$, using the DHT, the following matrix computation is performed (POULARIKAS, 2010):

$$\mathbf{X} = \frac{1}{N} \cdot \mathbf{H}_N \cdot \mathbf{x}.$$



(a) $\mathbf{H}_8$ (b) $\mathbf{H}_{16}$ (c) $\mathbf{H}_{32}$ (d) $\mathbf{H}_{64}$

Figura 2 – Image representation of the DHT matrix for $N = 8, 16, 32, 64$.

The arithmetic complexity of the above matrix multiplication requires at most N^2 real multiplications and $N(N-1)$ real additions.

Notice that the only difference between the DFT and DHT formulas (Equations (2.3) and (2.9)) is the imaginary unit j multiplying the sine term in the DFT. From this, we can see that there is a relationship between the two transforms, as shown in Equations (2.10) and (2.11). The existence of such this invertible rational transformation between the DFT and DHT implies that the two systems are equivalent in the sense that an algorithm capable of obtaining DFT can also be used to compute the DHT, and vice versa, with no additional multiplicative complexity (HEIDEMAN; BURRUS, 1988). Therefore, the DFT and DHT have identical multiplicative complexities (HEIDEMAN; BURRUS, 1988).

2.3 DISCRETE COSINE TRANSFORM

There are eight different variations of the DCT: DCT–I, DCT–II, DCT–III, DCT–IV, DCT–V, DCT–VI, DCT–VII, and DCT–III (BRITANAK; YIP; RAO, 2007). However, only the DCT–II is shown to optimally decorrelate Markov type-1 signals (BRITANAK; YIP; RAO, 2007). In this work, we only consider the DCT–II. Then, we refer to the DCT–II simply as DCT.

The DCT has its forward and inverse kernels defined as

$$\ker(i,k,N) = (1-(1-1/\sqrt{2})\delta_k)\sqrt{\frac{2}{N}}\cos\left((\frac{2i+k)\pi}{2N}\right),\ k=0,1,\ldots,N-1,$$

$$\ker^{-1}(i,k,N) = (1-(1-1/\sqrt{2})\delta_k)\sqrt{\frac{2}{N}}\cos\left((\frac{2i+k)\pi}{2N}\right),\ i=0,1,\ldots,N-1,$$

where

$$\delta_k = \begin{cases} 0, & \text{if } k = 0, \\ 1, & \text{otherwise.} \end{cases}$$

In fact, the DCT is an asymptotic case of the more general Karhunen-Loève transform (KLT). The analytical derivation of the DCT from the KLT is shown in the sequel.

2.3.1 The Karhunen–Loève transform

Let $\mathbf{x}$ be a random input vector with zero mean, which represents the input data to be decorrelated, where the superscript $^\top$ indicates the transposition operation. The KLT is a linear transformation represented by an orthogonal matrix $\mathbf{W}$ which decorrelates the variables in $\mathbf{x}$. The decorrelated output vector $\mathbf{y}$ is obtained according to the following operation:

$$\mathbf{y} = \begin{bmatrix} y_0 & y_1 & \dots & y_{N-1} \end{bmatrix}^\top = \mathbf{W}^\top \cdot \mathbf{x}. \tag{2.12}$$

If the transformation $\mathbf{W}^\top$ decorrelates the input variables, then the covariance matrix of the output vector $\mathbf{y}$ is given by the following diagonal matrix (BRITANAK; YIP; RAO, 2007):

$$\mathbf{R}_\mathbf{y} = \mathrm{E}\{\mathbf{y} \cdot \mathbf{y}^\top\} = \mathrm{diag}(\lambda_0, \lambda_1, \dots, \lambda_{N-1}), \tag{2.13}$$

where $\mathrm{E}(\cdot)$ represents the expectation operator, $\mathrm{diag}(\cdot)$ is the diagonal matrix generated by its arguments, and

$$\lambda_k = \mathrm{E}\left\{y_k^2\right\}, \quad k = 0, 1, \dots, N-1,$$

are the variances of the vector $\mathbf{y}$.

Replacing (2.12) in (2.13), it is possible to rewrite the covariance matrix of $\mathbf{y}$ as

$$\mathbf{R}_\mathbf{y} = \mathrm{E}\left\{\mathbf{W}^\top \cdot \mathbf{x} \cdot \mathbf{x}^\top \cdot \mathbf{W}\right\} = \mathbf{W}^\top \cdot \mathrm{E}\left\{\mathbf{x} \cdot \mathbf{x}^\top\right\} \cdot \mathbf{W} = \mathbf{W}^\top \cdot \mathbf{R}_\mathbf{x} \cdot \mathbf{W},$$

where $\mathbf{R}_\mathbf{x}$ is the covariance matrix of $\mathbf{x}$ which, by construction, is real and symmetric (GONZALEZ; WOODS, 2012; SEBER, 2008). Since $\mathbf{W}$ is intended to be orthogonal, it must satisfy $\mathbf{W}^{-1} = \mathbf{W}^\top$. Thus, we can write

$$\mathbf{R}_\mathbf{x} \cdot \begin{bmatrix} \mathbf{w}_0 | \mathbf{w}_1 | \cdots | \mathbf{w}_{N-1} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_0 | \mathbf{w}_1 | \cdots | \mathbf{w}_{N-1} \end{bmatrix} \cdot \mathbf{R}_\mathbf{y}, \tag{2.14}$$

where $\mathbf{w}_k$, $k = 0, 1, \ldots, N-1$, represents the kth column of the matrix $\mathbf{W}$. Therefore, Equation (2.14) can be rewritten as the following eigenvalue problem:

$$\mathbf{R}_{\mathbf{x}} \cdot \mathbf{w}_k = \lambda_k \cdot \mathbf{w}_k, \quad k = 0, 1, \ldots, N-1. \tag{2.15}$$

Note that the variances coincide with the eigenvalues. Solving Equation (2.15), we obtain the columns of $\mathbf{W}$, which are ordered according to the decreasing order of their respective eigenvalues (BRITANAK; YIP; RAO, 2007), thus resulting in the KLT.

2.3.2 Derivation of the discrete cosine transform

If the entries of the input vector $\mathbf{x}$ satisfy

$$x_m = \rho \cdot x_{m-1} + z_m,$$

where $\rho \in [0, 1]$ is the correlation coefficient and z_m is a white noise process, then we say that $\mathbf{x}$ is described by a first-order Markovian model (CINTRA; BAYER; TABLADA, 2014). In that case, the elements of the correlation matrix associated with $\mathbf{x}$ are given by (GONZALEZ; WOODS, 2012; BRITANAK; YIP; RAO, 2007)

$$[\mathbf{R}_{\mathbf{x}}]_{m,n} = \rho^{|m-n|}, \quad m, n = 0, 1, \ldots, N-1. \tag{2.16}$$

Solving Equation (2.15), we find that the mth component of the kth eigenvector $\mathbf{w}_k$, for $k, m = 0, 1, \ldots, N-1$, is given by (BRITANAK; YIP; RAO, 2007; RAY; DRIVER, 1970):

$$c_{k,m} = \sqrt{\frac{2}{N+\lambda_k}} \cdot \sin\left(\mu_k \left[(m+1) - \frac{N+1}{2}\right] + \frac{(k+1)\pi}{2}\right), \tag{2.17}$$

where

$$\lambda_k = \frac{1-\rho^2}{1 - 2\rho\cos(\mu_k) + \rho^2} \tag{2.18}$$

is the kth eigenvalue associated to $\mathbf{w}_k$ and μ_k, $k = 0, 1, \ldots, N-1$, are the real-valued roots of the following transcendental equation in μ:

$$\tan(N\mu) = -\frac{(1-\rho^2)\sin(\mu)}{(1+\rho^2)\cos(\mu) - 2\rho}. \tag{2.19}$$

Assuming highly correlated input data, that is, $\rho \to 1$, we notice that the right side of Equation (2.19) goes to zero. Therefore, the N real-valued positive roots of Equation (2.19) are given by

$$\mu_k = \frac{k\pi}{N}, \quad k = 0, 1, \ldots, N-1.$$

Thus, replacing the values of μ_k in Equation (2.18), we have that $\lambda_k = 0$ for $k \neq 0$. Now, there is only λ_0 left to compute in order to obtain a closed expression for Equation (2.17). From (STRANG, 1988, p. 251), we have that the trace (BRITANAK; YIP; RAO, 2007) of $\mathbf{R_x}$, defined as

$$\mathrm{tr}(\mathbf{R_x}) = \sum_{n=0}^{N-1} [\mathbf{R_x}]_{n,n},$$

equals the sum of the N eigenvalues. From Equation (2.16), we have that $\mathrm{tr}(\mathbf{R_x}) = N$. Thus

$$\mathrm{tr}(\mathbf{R_y}) = \sum_{k=0}^{N-1} \lambda_k = \lambda_0 = \mathrm{tr}(\mathbf{R_x}) = N \quad \therefore \quad \lambda_0 = N.$$

Finally, we obtain that

$$c_{0,m} = \frac{1}{\sqrt{N}}, \quad k = 0,$$
$$c_{k,m} = \sqrt{\frac{2}{N}} \sin\left(\frac{k(2m+1)}{2N} + \frac{\pi}{2}\right) = \sqrt{\frac{2}{N}} \cos\left(\frac{(2m+1)k\pi}{2N}\right), \quad k \neq 0.$$

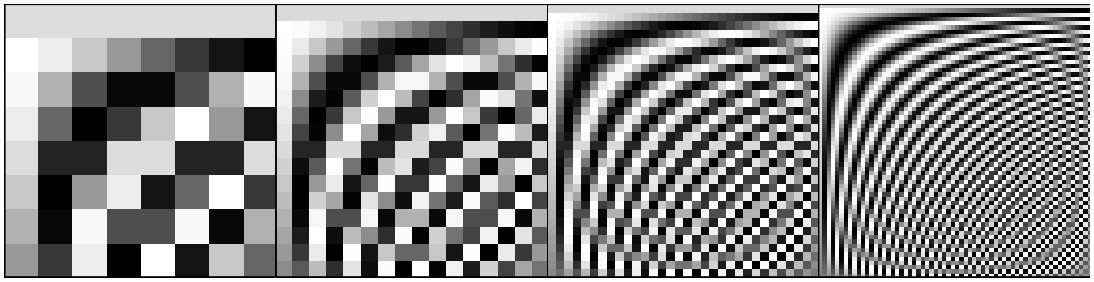
Introducing a constant α_k, we can combine the equations above, obtaining

$$c_{k,m} = \sqrt{\frac{2}{N}} \alpha_k \cos\left(\frac{(2m+1)k\pi}{2N}\right), \tag{2.20}$$

where $\alpha_0 = 1/\sqrt{2}$ and $\alpha_k = 1$, if $k \neq 0$.

The linear transformation whose matrix has elements defined as in Equation (2.20) is called the discrete cosine transform (ARAI; AGUI; NAKAJIMA, 1988; GONZALEZ; WOODS, 2012). Therefore, the DCT is asymptotically equivalent to the KLT when $\rho \to 1$. Such relationship justifies the good decorrelation and energy compression properties of the DCT when the input data follows a highly correlated first order stationary Markovian process.

Similar to the DFT and the DHT, the DCT matrix, $\mathbf{C}_N$, also shows some patterns that are easier to see in its image representation. Figure 3 shows those images. It is possible to see, specially for larger values of N, that the DCT matrix has a structure very similar to the top left quadrant of both the DFT and DHT.



(a) $\mathbf{C}_8$ (b) $\mathbf{C}_{16}$ (c) $\mathbf{C}_{32}$ (d) $\mathbf{C}_{64}$

Figura 3 – Image representation of the DCT matrix for $N = 8, 16, 32, 64$.

2.3.3 Computational complexity

The direct computation of the DCT transform of an input signal $\mathbf{x}$,

$$\mathbf{X} = \mathbf{C}_N \cdot \mathbf{x}$$

requires at most N^2 multiplications and $N(N-1)$ additions.

The arithmetic computational complexities of the DFT, DHT, and DCT discussed here refer only to its direct implementation. In practice, fast algorithms are employed, being able to drastically reduce the arithmetic cost of the transform computation.

3 FAST ALGORITHMS AND APPROXIMATIONS FOR THE 8-POINT DCT

An algorithm is a detailed description of a computational procedure (BLAHUT, 2010). A fast algorithm is an alternative, computationally efficient way or realizing the same procedure, which is not the obvious way to compute the output from the input (BLAHUT, 2010). In the context of discrete transforms, the performance of an algorithm is usually measured by the number of multiplications and additions it requires (BLAHUT, 2010). For example, in Chapter 2, Section 1, Equations 2.7 and 2.8 are fast algorithms for computing Equations 2.5 and 2.6, respectively.

If considering butterfly-based structures as commonly found in decimation-in-frequency algorithms, such as (HOU, 1987; YIP; RAO, 1988; RAO; YIP, 1990), a fast algorithm for a given matrix $\mathbf{A}$ can be the product of several matrices, for example, $\mathbf{A} = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3$. In this case, the computational cost of computing $\mathbf{A} \cdot \mathbf{x}$ is replaced by the cost of computing $\mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \mathbf{A}_3 \cdot \mathbf{x}$. Even though the number of matrices increase, the computational cost decreases because $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}_3$ are usually sparse matrices containing mostly 1's and -1's, which results in trivial multiplications (SALOMON; MOTTA; BRYANT, 2007).

Since one of our goals in this work is to derive new approximations for the 8-point DCT, we present next some popular fast algorithms and low-complexity approximations for this particular transform.

3.1 FAST ALGORITHMS FOR THE 8-POINT DCT

Let $\mathbf{C}_8$ be the 8-point DCT matrix. Because of its symmetries, $\mathbf{C}_8$ can be represented in the following way:

$$\mathbf{C}_8 = \frac{1}{2} \cdot \begin{bmatrix} \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 & \gamma_3 \\ \gamma_0 & \gamma_2 & \gamma_4 & \gamma_6 & -\gamma_6 & -\gamma_4 & -\gamma_2 & -\gamma_0 \\ \gamma_1 & \gamma_5 & -\gamma_5 & -\gamma_1 & -\gamma_1 & -\gamma_5 & \gamma_5 & \gamma_1 \\ \gamma_2 & -\gamma_6 & -\gamma_0 & -\gamma_4 & \gamma_4 & \gamma_0 & \gamma_6 & -\gamma_2 \\ \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 & \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 \\ \gamma_4 & -\gamma_0 & \gamma_6 & \gamma_2 & -\gamma_2 & -\gamma_6 & \gamma_0 & -\gamma_4 \\ \gamma_5 & -\gamma_1 & \gamma_1 & -\gamma_5 & -\gamma_5 & \gamma_1 & -\gamma_1 & \gamma_5 \\ \gamma_6 & -\gamma_4 & \gamma_2 & -\gamma_0 & \gamma_0 & -\gamma_2 & \gamma_4 & -\gamma_6 \end{bmatrix},$$

where

$$\gamma_0 = \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} \approx 0{,}9808\ldots, \quad \gamma_1 = \frac{\sqrt{2}}{2} \approx 0{,}707\ldots,$$
$$\gamma_2 = \frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2} \approx 0{,}8315\ldots, \quad \gamma_3 = \frac{\sqrt{2+\sqrt{2}}}{2} \approx 0{,}9239\ldots,$$
$$\gamma_4 = \frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2} \approx 0{,}5556\ldots, \quad \gamma_5 = \frac{\sqrt{2-\sqrt{2}}}{2} \approx 0{,}3827\ldots,$$
$$\gamma_6 = \frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2} \approx 0{,}1951\ldots$$

Thus, an 8-point input vector might have its components decorrelated by the following expression:

$$\mathbf{X} = \mathbf{C}_8 \cdot \mathbf{x},$$

where $\mathbf{X}$ is the decorrelated vector.

Explicitly, we have that

$$\begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_7 \end{bmatrix} = \begin{bmatrix} c_{0,0} & c_{0,1} & \cdots & c_{0,7} \\ c_{1,0} & c_{1,1} & \cdots & c_{1,7} \\ \vdots & \vdots & \ddots & \vdots \\ c_{7,0} & c_{7,1} & \cdots & c_{7,7} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_7 \end{bmatrix} = \begin{bmatrix} c_{0,0}x_0 + c_{0,1}x_1 + \cdots + c_{0,7}x_7 \\ c_{1,0}x_0 + c_{1,1}x_1 + \cdots + c_{1,7}x_7 \\ \vdots \\ c_{7,0}x_0 + c_{7,1}x_1 + \cdots + c_{7,7}x_7 \end{bmatrix}. \tag{3.1}$$

Therefore, the arithmetic cost to decorrelate the input vector using $\mathbf{C}_8$ is 64 multiplications and 56 additions.

Besides having a closed form, an import factor for the usage of the DCT is the existence of fast algorithms that allows its efficient calculation. Common fast algorithms for the computation of the 8-point DCT include: (i) Yuan *et al.* (YUAN; HAO; XU, 2006), (ii) Arai *et al.* (ARAI; AGUI; NAKAJIMA, 1988), (iii) Chen *et al.* (CHEN; SMITH; FRALICK, 1977), (iv) Feig–Winograd (FEIG; WINOGRAD, 1992) and, Loeffler *et al.* (LOEFFLER; LIGTENBERG; MOSCHYTZ, 1989). The arithmetic cost of those and other methods is listed in Table 1.

The theoretical minimum of multiplicative complexity for this length is 11 multiplications (**??**), which is attained, for example, by the Loeffler *et al.* (LOEFFLER; LIGTENBERG;

Algorithm	Multiplication	Additions
Loeffler	11	29
Suehiro	12	29
Yuan	12	29
Lee	12	29
Vetterli	12	29
Hou	12	29
Wang	13	29
Arai *et al.*	13	29
Chen *et al.*	16	26
Feig–Winograd	22	28

Tabela 1 – Arithmetic cost of the fast algorithms for the exact 8-point DCT

MOSCHYTZ, 1989) algorithm. This result is obtained when we consider: (i) the computation of the DCT as a cyclic convolution, and (ii) the results presented in (**??**), as demonstrated in (DUHAMEL; VETTERLI, 1987).

3.2 MATRIX APPROXIMATIONS

As shown on the previous section, the entries of the DCT matrix are irrational quantities. Thus, given the limited precision of computers, its practical implementation with exact numeric precision it is unfeasible (WALLACE, 1992). In this sense, the fast algorithms previously mentioned are implemented by means of truncation and/or rounding of its coefficients with its precision defined according to the desired application. Despite substantially reducing the computational cost of its implementation, the fast algorithms for the DCT considered do not eliminate the need for the use of floating-point or fixed-point with large integers. The cost of the elementary arithmetic operations in floating-point numeric representation is usually higher than the cost from operations in fixed-point arithmetic or simpler representations (BRITANAK; YIP; RAO, 2007). For this reason, the hardware implementation using floating-point arithmetic requires greater consumption of power and area resources. Additionally, given the maturity of the area of fast algorithms for the 8-point DCT, there is little space for improvement over the methods already archived in literature.

An alternative approach to further reduce the computational cost of the DCT over the methods already archived in literature is the use of matrix approximations (BAYER; CINTRA,

2010; CINTRA; BAYER; TABLADA, 2014). Such approximations are matrices with low computational cost that have similar mathematical structure to the exact transforms. That is, let $\mathbf{C}_N$ be the N-point DCT, an approximation for $\mathbf{C}_N$, $\widehat{\mathbf{C}}_N$, is a matrix such that

$$\widehat{\mathbf{X}} = \widehat{\mathbf{C}}_N \cdot \mathbf{x} \approx \mathbf{C}_N \cdot \mathbf{x} = \mathbf{X}.$$

Thus, $\widehat{\mathbf{X}} \approx \mathbf{X}$, according to some criteria, such as proximity or coding measures (BRITANAK; YIP; RAO, 2007). An approximation for $\mathbf{C}_N$ can be obtained from a low-complexity multiplierless matrix $\mathbf{T}$. That is, a matrix whose elements are zeros or powers of two. Notice that in binary representation, multiplications by powers of two can be performed by simple bit-shifting operation (BRITANAK; YIP; RAO, 2007). Such multiplications have no cost in hardware implementations (MADANAYAKE et al., 2012) and are regarded as trivial multiplications (BLAHUT, 2010).

The design of approximate transforms is often based on structural aspects of the exact transforms, such as: symmetries (CHAM, 1989), fast algorithms (HOU, 1987; LOEFFLER; LIGTENBERG; MOSCHYTZ, 1989), parametrization (FEIG; WINOGRAD, 1992), and numerical properties (YUAN; HAO; XU, 2006). In a general manner, the low-complexity matrix $\mathbf{T}$ from which we derive an approximation for a trigonometric transform, $\mathbf{C}$, is obtained by solving the optimization problem below:

$$\mathbf{T} = \arg\min_{\mathbf{T}'} \operatorname{approx}(\mathbf{T}', \mathbf{C}),$$

where $\operatorname{approx}(\cdot, \cdot)$ is a specific objective function—such as proximity or performance measures (BRITANAK; YIP; RAO, 2007)—submitted to several constraints, such as orthogonality and low-complexity of the candidate matrices $\mathbf{T}'$.

Two important concepts when discussing approximate transforms are presented next.

Definition 3.1 (Orthogonality)**.** *Matrix* $\mathbf{A}$ *is said to be row orthogonal or simply orthogonal if* $\mathbf{A} \cdot \mathbf{A}^\top$ *is a diagonal matrix.*

Definition 3.2 (Orthonormality)**.** *If* $\mathbf{A} \cdot \mathbf{A}^\top$ *is the identity matrix, then* $\mathbf{A}$ *is said to be orthonormal.*

If $\mathbf{T}$ is orthogonal, then its inverse is given by $\mathbf{T}^{-1} = \mathbf{T}^{\top} \cdot \mathbf{D}^{-1}$, where $\mathbf{D}$ is the diagonal matrix resulting from $\mathbf{T} \cdot \mathbf{T}^{\top}$. In particular, if $\mathbf{T}$ is orthonormal, then $\mathbf{T}^{-1} = \mathbf{T}^{\top}$. As a consequence of that, if $\mathbf{T}$ is orthogonal and can be decomposed into the product of p matrices, that is,

$$\mathbf{T} = \mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \cdots \cdot \mathbf{A}_p,$$

then

$$\mathbf{T}^{-1} = \mathbf{T}^{\top} \cdot \mathbf{D}^{-1} = (\mathbf{A}_1 \cdot \mathbf{A}_2 \cdot \cdots \cdot \mathbf{A}_p)^{\top} \cdot \mathbf{D}^{-1} = \mathbf{A}_p^{\top} \cdot \mathbf{A}_{p-1}^{\top} \cdot \cdots \cdot \mathbf{A}_1^{\top} \cdot \mathbf{D}^{-1}.$$

Another reason to pursue the orthogonality of $\mathbf{T}$ is that the exact DCT is an orthonormal matrix. Orthonormal approximations can be obtained when $\mathbf{T}$ is orthogonal.

In this work, we consider, among the existing orthonormalization methods (HIGHAM, 2008; WATKINS, 2004), the one based on polar decomposition (HIGHAM, 1986; HIGHAM; SCHREIBER, 1988), which preserves the low-complexity structure of the transform $\mathbf{T}$. In this case, the orthonormalization procedure only requires the computation of a diagonal matrix given by

$$\mathbf{D} = \sqrt{(\mathbf{T} \cdot \mathbf{T}^{\top})^{-1}}, \tag{3.2}$$

where $\sqrt{\cdot}$ represents the matrix square root operation (HIGHAM, 1987).

Lastly, as shown in (HIGHAM, 1986; CINTRA; BAYER, 2011; CINTRA, 2011; BRITANAK; YIP; RAO, 2007; BAYER; CINTRA, 2012), an orthonormal approximation for the DCT is given by

$$\widehat{\mathbf{C}}_N = \mathbf{D} \cdot \mathbf{T}. \tag{3.3}$$

3.2.1 Nonorthogonal case

Notice that Equations 3.2 and 3.3 are derived considering that $\mathbf{T}$ is orthogonal, i.e., that $\mathbf{T}$ satisfies Definition 3.1. When that is not the case, the elements outside the diagonal of $\mathbf{T} \cdot \mathbf{T}^{\top}$ result in an increase of the computational complexity of $\widehat{\mathbf{C}}_N$. In this scenario, a possible solution to obtain orthonormal approximations, is to approximate $\mathbf{D}$ itself by setting to zero the

off-diagonal elements. Therefore, the approximate diagonal is given by

$$\widehat{\mathbf{D}} = \sqrt{(\mathrm{diag}(\mathbf{T} \cdot \mathbf{T}^{\top}))^{-1}},$$

hence

$$\widehat{\mathbf{C}}_N = \widehat{\mathbf{D}} \cdot \mathbf{T}.$$

Next, we present a representative selection of low-complexity matrices from which approximations for the DCT can be derived.

3.2.2 Low-complexity matrices for DCT approximation

3.2.2.1 The Walsh-Hadamard transform:

The order N Walsh-Hadamard transform (WHT) (HORADAM, 2007) is given by a binary $N \times N$ matrix, $\mathbf{T}_{\text{WHT–}N}$, with entries in $\{\pm 1\}$ that satisfies:

$$\mathbf{T}_{\text{WHT–}N} \cdot \mathbf{T}_{\text{WHT–}N}^{\top} = N \cdot \mathbf{I}_N,$$

where $\mathbf{I}_N$ represents the identity matrix or order N. The 8-point WHT is given by

$$\mathbf{T}_{\text{WHT–8}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix},$$

with the diagonal matrix implied by Equation 3.2 given by

$$\mathbf{D}_{\text{WHT–8}} = \mathrm{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right).$$

The WHT is used in image processing due to its good performance and simplicity of implementation (HORADAM, 2007). Then, even though the WHT was not proposed as an approximation for the DCT, it is used as an alternative to the DCT.

3.2.2.2 The signed DCT (SCDT):

The first matrix in the literature proposed as an approximation for the DCT was introduced by Haweel in (HAWEEL, 2001). The signed DCT (SDCT) is a nonorthogonal matrix

obtained from the application of the sign function to each element of $\mathbf{C}_8$. The sign function is given by $\mathrm{sign}(x) = |x|/x$, $x \neq 0$ and $\mathrm{sign}(0) = 0$. Thus, the low-complexity matrix associated to the 8-point SDCT is given by

$$\mathbf{T}_{\text{SDCT}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

with

$$\mathbf{D}_{\text{SDCT}} = \mathrm{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right).$$

3.2.2.3 The level 1 approximation by Lengwehasatit and Ortega:

Lengwehasatit and Ortega proposed five levels of approximation for the DCT based on the input signal features (LENGWEHASATIT; ORTEGA, 2004). The level one approximation is generated by the low-complexity orthogonal matrix below:

$$\mathbf{T}_{\text{LO}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

with

$$\mathbf{D}_{\text{LO}} = \mathrm{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{6}}\right).$$

3.2.2.4 The series of approximations BAS:

The series of approximations BAS was proposed by Bouguezel, Ahmad, and Swamy (BOUGUEZEL; AHMAD; SWAMY, 2008a; BOUGUEZEL; AHMAD; SWAMY, 2008b; BOUGUEZEL; AHMAD; SWAMY, 2009; BOUGUEZEL; AHMAD; SWAMY, 2010; BOUGUEZEL; AHMAD; SWAMY, 2011; BOUGUEZEL; AHMAD; SWAMY, 2013). Many of these approximations were obtained from SDCT modifications (TABLADA; BAYER; CINTRA, 2015). Table 2 displays the matrices considered in this work.

Transform	Matrix	Orthogonal?	$\mathbf{D}$
$\mathbf{T}_{\text{BAS–1}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}\right)$
$\mathbf{T}_{\text{BAS–2}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 0 & 0 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$	No	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$
$\mathbf{T}_{\text{BAS–3}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}\right)$
$\mathbf{T}_{\text{BAS–4}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 1 & -1 & -2 & 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 1 & -2 & 2 & -1 & 1 & -2 & 2 & -1 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}\right)$
$\mathbf{T}_{\text{BAS–5}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$	No	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{2}\right)$
$\mathbf{T}_{\text{BAS–6}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$

Tabela 2 – BAS approximations for $\mathbf{C}_8$

3.2.2.5 The rounded DCT (RDCT):

Given $x \in \mathbb{R}$, let $\lfloor x \rfloor$ be the largest integer that does not exceed x. The round function, as implemented in Matlab/Octave, is defined by

$$\text{round}(x) = \text{sign}(x) \cdot \lfloor x + 0.5 \rfloor.$$

Applied to matrices, the round function operates elementwise.

The rounded DCT was proposed by Cintra and Bayer in (BAYER; CINTRA, 2010). The low-complexity orthogonal matrix RDCT is obtained by the application of the rounding

function to the DCT matrix entries as follows:

$$\mathbf{T}_{\text{RDCT}} = \text{round}(2 \cdot \mathbf{C}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

with

$$\mathbf{D}_{\text{RDCT}} = \text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}\right).$$

3.2.2.6 The modified RDCT:

The modified RDCT (MRDCT) was introduced by Bayer and Cintra in (BAYER; CINTRA, 2012). The MRDCT is an orthogonal matrix obtained by replacing some elements of the RDCT matrix by zeros. Its explicit form is presented next:

$$\mathbf{T}_{\text{MRDCT}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

with

$$\mathbf{D}_{\text{RDCT}} = \text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}\right)$$

The difference matrix is given by:

$$\mathbf{T}_{\text{RDCT}} - \mathbf{T}_{\text{MRDCT}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}.$$

3.2.2.7 The series of approximations CBT:

In (CINTRA; BAYER; TABLADA, 2014), Cintra, Bayer and Tablada obtained a series of approximations, which we are going to refer to as CBT, by means of applying several different rounding approximations to the DCT matrix. The matrices introduced in (CINTRA; BAYER; TABLADA, 2014) are shown in Table 3.

The quality of the approximations shown here, in terms of figures of merit common in literature, is going to be discussed in Chapter 6 along with the results for the new approximations proposed.

Transform	Matrix	Orthogonal?	$\mathbf{D}$
$\mathbf{T}_{\text{CBT–1}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{2}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{2}, \frac{1}{\sqrt{12}}\right)$
$\mathbf{T}_{\text{CBT–2}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 0 & 0 & -2 & -2 & 0 & 0 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 0 & -2 & 2 & 0 & 0 & 2 & -2 & 0 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{16}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{16}}, \frac{1}{\sqrt{12}}\right)$
$\mathbf{T}_{\text{CBT–3}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}\right)$
$\mathbf{T}_{\text{CBT–4}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}\right)$
$\mathbf{T}_{\text{CBT–5}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{bmatrix}$	Yes	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{12}}\right)$
$\mathbf{T}_{\text{CBT–6}}$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 \end{bmatrix}$	No	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$
$\mathbf{T}_{\text{CBT–7}}$	$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 1 & 1 & -1 & -1 & -2 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & -1 & -2 & -1 & 1 & 2 & 1 & -2 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & -2 \\ 1 & -2 & 1 & 2 & -2 & -1 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 1 & -1 & 2 & -2 & 2 & -2 & 1 & -1 \end{bmatrix}$	No	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{20}}\right)$

Tabela 3 – Series of approximations CBT for $\mathbf{C}_8$

4 SEARCH METHOD

In this chapter, we introduce a new search method for finding matrix approximations. Notice that, even though the development of the method was motivated by the trigonometric transforms, the proposed method is completely general and might be used to approximate any matrix. Although our main goal is to find new approximations for the DCT, we are going to explore in this chapter the ramifications of the proposed method when applied to the DHT and DFT also.

4.1 OVERALL STRUCTURE AND INITIAL CONCEPTS

Let $\mathbf{A}$ be an arbitrary $N \times M$ matrix with elements in $\mathbb{R}$, and $\mathbf{a}_k = \begin{bmatrix} a_{k,0} & a_{k,1} & \dots & a_{k,M-1} \end{bmatrix}$, $k = 0, 1, \dots, N-1$, be a row vector that represents the kth row of $\mathbf{A}$. Note that $\mathbf{A}$ might be described by its rows as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_0 \\ \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{N-1} \end{bmatrix}.$$

Aiming at finding a low-complexity approximation $\mathbf{T}$ for $\mathbf{A}$, we reduced the problem of approximating the whole matrix into the problem of approximating its rows by low-complexity row vectors. Such heuristic can be categorized as greedy (CORMEN et al., 2001). Therefore, our goal is to derive integer low-complexity matrices

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_{N-1} \end{bmatrix},$$

such that its rows $\mathbf{t}_k$, $k = 0, 1, \dots, N-1$, satisfy

$$\mathbf{t}_k = \arg \min_{\mathbf{t} \in \mathcal{D}_\mathcal{P}} \text{approx}(\mathbf{t}, \mathbf{a}_k), \quad k = 0, 1, \dots, N-1, \tag{4.1}$$

where $\mathcal{D}_\mathcal{P}$ is the search space. We characterize the search space in the next subsection.

4.1.1 Search Space

In order to obtain a low-complexity matrix $\mathbf{T}$, its entries must be computationally simple (BRITANAK; YIP; RAO, 2007; BLAHUT, 2010). We define the search space as the collection of M-point row vectors whose entries are in a set, say $\mathcal{P}$, of low-complexity elements. That is, the search space $\mathcal{D}_{\mathcal{P}}$ is composed by all the possible permutations of length M of the elements in $\mathcal{P}$. Therefore, the cardinality of the search space is given by $|\mathcal{D}_{\mathcal{P}}| = |\mathcal{P}|^M$. A particular vector in $\mathcal{D}_{\mathcal{P}}$ is denoted by $\mathcal{D}_{\mathcal{P}}(i)$, $i = 1, 2, \ldots, |\mathcal{D}_{\mathcal{P}}|$. Some choices for $\mathcal{P}$ include: $\mathcal{P}_1 = \{0, \pm 1\}$ and $\mathcal{P}_2 = \{0, \pm 1, \pm 2\}$, where all elements in $\mathcal{P}$ are trivial multiplicands as discussed in Chapter 3, Section 2.

To illustrate, approximating 8×8 matrices may require search spaces $\mathcal{D}_{\mathcal{P}_1}$ and $\mathcal{D}_{\mathcal{P}_2}$ as shown in Tables 4 and 5, respectively. Such search spaces have cardinality $|\mathcal{P}_1|^8 = 3^8 = 6,561$ and $|\mathcal{P}_2|^8 = 5^8 = 390,625$ elements, respectively.

i	$\mathcal{D}_{\mathcal{P}_1}(i)$
1	$\begin{bmatrix}-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1\end{bmatrix}$
2	$\begin{bmatrix}-1 & -1 & -1 & -1 & -1 & -1 & -1 & 0\end{bmatrix}$
⋮	⋮
3200	$\begin{bmatrix}0 & 0 & 0 & -1 & 0 & 0 & 0 & 1\end{bmatrix}$
3201	$\begin{bmatrix}0 & 0 & 0 & -1 & 0 & 0 & 1 & -1\end{bmatrix}$
⋮	⋮
6560	$\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 0\end{bmatrix}$
6561	$\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}$

Tabela 4 – Examples of approximated vectors from the search space $\mathcal{D}_{\mathcal{P}_1}$

i	$\mathcal{D}_{\mathcal{P}_2}(i)$
1	$\begin{bmatrix}-2 & -2 & -2 & -2 & -2 & -2 & -2 & -2\end{bmatrix}$
2	$\begin{bmatrix}-2 & -2 & -2 & -2 & -2 & -2 & -2 & -1\end{bmatrix}$
⋮	⋮
150000	$\begin{bmatrix}-1 & 2 & 1 & -2 & -2 & -2 & -2 & -2\end{bmatrix}$
150001	$\begin{bmatrix}-1 & 2 & 1 & -2 & -2 & -2 & -2 & -1\end{bmatrix}$
⋮	⋮
390624	$\begin{bmatrix}2 & 2 & 2 & 2 & 2 & 2 & 2 & 1\end{bmatrix}$
390625	$\begin{bmatrix}2 & 2 & 2 & 2 & 2 & 2 & 2 & 2\end{bmatrix}$

Tabela 5 – Examples of approximated vectors from the search space $\mathcal{D}_{\mathcal{P}_2}$

4.1.2 Objective Function

The problem posed in (4.1) requires the identification of an error function to quantify the "distance" between the candidate row vectors from $\mathcal{D}_{\mathcal{P}}$ and the rows of the exact matrix $\mathbf{A}$. Related literature often considers error functions based on matrix norms (CINTRA; BAYER, 2011), proximity to orthogonality measures (TABLADA; BAYER; CINTRA, 2015), and coding performance measures (BRITANAK; YIP; RAO, 2007).

In this work, we propose the utilization of a distance based on the angle between vectors as the objective function to be minimized. Let $\mathbf{u}$ and $\mathbf{v}$ be two M-dimensional vectors defined over $\mathbb{R}^M$. The angle between vectors is simply given by:

$$\text{angle}(\mathbf{u}, \mathbf{v}) = \arccos\left(\frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}\right), \tag{4.2}$$

where $\langle \cdot, \cdot \rangle$ is the inner product and $\|\cdot\|$ indicates the norm induced by the inner product (STRANG, 1988).

4.2 ANGLE BASED METHOD

Based on the previous discussion, we are able to propose the angle based method, which is based on the optimization problem stated as follows:

$$\mathbf{t}_k = \arg\min_{\mathbf{t} \in \mathcal{D}_{\mathcal{P}}} \text{angle}(\mathbf{a}_k, \mathbf{t}), \quad k = 0, 1, \ldots, N-1. \tag{4.3}$$

First, we select the set $\mathcal{P}$ and span the induced search space $\mathcal{D}_{\mathcal{P}}$. Then, for each row of $\mathbf{A}$, we generate a subset of the search space, $\mathcal{D}_{\mathcal{P}}^{(k)}$, $k = 0, 1, \ldots, N-1$, containing all the vectors in $\mathcal{D}_{\mathcal{P}}$ that are solutions to the problem in (4.3). Lastly, each approximate matrix is obtained as a combination of the vectors in $\mathcal{D}_{\mathcal{P}}^{(k)}$, $k = 0, 1, \ldots, N-1$. The number of matrices

obtained is given by $\prod_{k=0}^{N-1} |\mathcal{D}_{\mathcal{P}}^{(k)}|$. Therefore,

$$\mathbf{T}^{(i)} = \begin{bmatrix} \mathbf{t}_0 \\ \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_k \\ \vdots \\ \mathbf{t}_{N-1} \end{bmatrix}, \quad i = 1, 2, \ldots, \prod_{k=0}^{N-1} |\mathcal{D}_{\mathcal{P}}^{(k)}|,$$

where $\mathbf{t}_k \in \mathcal{D}_{\mathcal{P}}^{(k)}$.

The procedure for the angle based method is shown in Algorithm 1.

Example 4.1. *Let* $\mathbf{A}$ *be a* 4×4 *matrix. Suppose that, after applying Algorithm 1 to approximate* $\mathbf{A}$*, we obtained* $|\mathcal{D}_{\mathcal{P}}^{(1)}| = |\mathcal{D}_{\mathcal{P}}^{(3)}| = 1$*, and* $|\mathcal{D}_{\mathcal{P}}^{(2)}| = |\mathcal{D}_{\mathcal{P}}^{(4)}| = 2$*. In this case, we obtain* $\prod_{k=1}^{4} |\mathcal{D}_{\mathcal{P}}^{(k)}| = 1 \cdot 2 \cdot 1 \cdot 2 = 4$ *approximate matrices, given by:*

$$\mathbf{T}^{(1)} = \begin{bmatrix} \mathcal{D}_{\mathcal{P}}^{(1)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(2)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(3)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(4)}(1) \end{bmatrix}, \quad \mathbf{T}^{(2)} = \begin{bmatrix} \mathcal{D}_{\mathcal{P}}^{(1)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(2)}(2) \\ \mathcal{D}_{\mathcal{P}}^{(3)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(4)}(1) \end{bmatrix}, \quad \mathbf{T}^{(3)} = \begin{bmatrix} \mathcal{D}_{\mathcal{P}}^{(1)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(2)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(3)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(4)}(2) \end{bmatrix}, \quad \mathbf{T}^{(4)} = \begin{bmatrix} \mathcal{D}_{\mathcal{P}}^{(1)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(2)}(2) \\ \mathcal{D}_{\mathcal{P}}^{(3)}(1) \\ \mathcal{D}_{\mathcal{P}}^{(4)}(2) \end{bmatrix},$$

where $\mathcal{D}_{\mathcal{P}}^{(k)}(i)$*,* $k = 0, 1, \ldots, N-1$*,* $i = 1, 2, \ldots, |\mathcal{D}_{\mathcal{P}}^{(k)}|$*, represents the* i*th vector in* $\mathcal{D}_{\mathcal{P}}^{(k)}$*.*

loa 1 – Pseudo algorithm for angle based method

Input: $\mathbf{A}$, $\mathcal{D}_{\mathcal{P}}$
Output: **approximations** (3 dimensional array containing all the obtained approximate matrices)

for $k \leftarrow 0, 1, \ldots, N-1$ **do**
  $angles \leftarrow$ null vector of length $|\mathcal{D}_{\mathcal{P}}|$
  for $i \leftarrow 1, 2, \ldots, |\mathcal{D}_{\mathcal{P}}|$ **do**
    $angles(i) \leftarrow \text{angle}(\mathbf{a}_k, \mathcal{D}_{\mathcal{P}}(i))$;
  end for
  $indexes \leftarrow$ indexes of the vectors in $\mathcal{D}_{\mathcal{P}}$ for which $angles = \min(angles)$;
  $\mathcal{D}_{\mathcal{P}}^{(k)} \leftarrow \mathcal{D}_{\mathcal{P}}(indexes)$;
end for
approximations $\leftarrow$ Null array with dimensions $N \times M \times \prod_{k=0}^{N-1} |\mathcal{D}_{\mathcal{P}}^{(k)}|$;
approximations $\leftarrow$ All combinations of the vectors in $\mathcal{D}_{\mathcal{P}}^{(k)},\ k = 0, 1, \ldots, N-1$;

4.3 ANGLE BASED METHOD - CONSTRAINED TO ORTHOGONALITY

Note that the previous method does not guarantee that the obtained matrices are orthogonal. However, orthogonality is a desirable feature, as discussed in Chapter 3, Section 2. In order to ensure orthogonality, we reformulate the previously stated optimization problem based on the order that the rows of the exact matrix are approximated. Thereby, a constrained to orthogonality version of the proposed method is derived.

4.3.1 Search sequence

For the unconstrained method, since there is no constraints to the optimization problem, the rows are approximated independently from each other. However, if considering orthogonality as a constraint, we define a dependency relation among the rows. Hence, the sequence in which we approximate the rows must be considered.

There are N rows to be approximated. One way of doing it is—under some criteria–to approximate the rows following their natural order, i.e., first we approximate the 1st row, then the 2nd row, and so on. The 2nd row is approximated subject to the orthogonality constraint relative to the resulting approximate 1st row. The 3rd row is approximated considering orthogonality relative to approximate rows 1 and 2, and so on. This procedure corresponds to the sequence $\wp_1 = (1, 2, 3, \ldots, N)$. However, this is only a particular search sequence. Therefore, for a systematic procedure, we must consider all the $N!$ possible permutations of $\wp_1$. Let $\wp_m$, $m = 1, 2, \ldots, N!$, be the mth permutation of $\wp_1$, and $\wp_m(k)$, $k = 0, 1, \ldots, N-1$, be a particular element of $\wp_m$. For example, if $N = 8$, there are $N! = 40320$ possible search sequences. In this case, we have $\wp_{1250} = (1, 3, 7, 6, 5, 4, 8, 2)$ and $\wp_{1250}(2) = 7$.

4.3.2 Optimization problem

In view of the previous discussion, we can now fully specify the optimization problem suitable for the proposed algorithm under the orthogonality constraint. Fixing a search sequence $\wp_m$, the optimization problem is stated as follows:

$$\mathbf{t}_{\wp_m(k)} = \arg \min_{\mathbf{t} \in \mathcal{D}_\mathcal{P}} \text{angle}(\mathbf{a}_{\wp_m(k)}, \mathbf{t}), \quad k = 0, 1, \ldots, N-1, \tag{4.4}$$

subject to

$$\langle \mathbf{t}_{\wp_m(i)}, \mathbf{t}_{\wp_m(j)} \rangle = 0, \quad i \neq j. \tag{4.5}$$

The solution of the problem above returns N row vectors $\mathbf{t}_{\wp_m(0)}, \mathbf{t}_{\wp_m(1)}, \ldots, \mathbf{t}_{\wp_m(N-1)}$ that are taken as the rows of the approximate matrix $\mathbf{T}$.

Algorithm 2 displays the procedure for the constrained to orthogonality version of the proposed method.

loa 2 – Algorithm for the angle based method constrained to orthogonality

Input: $\mathbf{A}$; $\mathcal{D}_\mathcal{P}$; $\wp$ ($N! \times N$ matrix containing all the possible search sequences).
Output: **approximations** (3 dimensional array containing all the obtained approximate matrices).

approximations $\leftarrow$ Null array with dimensions $N \times M \times N!$;
for $m \leftarrow 1, 2, \ldots, N!$ **do**
  for $k \leftarrow 0, 1, \ldots, N-1$ **do**
    $\theta_{min} \leftarrow 2\pi$;
    $index \leftarrow 1$;
    for $i \leftarrow 1, 2, \ldots, |\mathcal{D}_\mathcal{P}|$ **do**
      $aux \leftarrow$ **approximations**$(:, :, m) \cdot (\mathcal{D}_\mathcal{P}(i))^\top$
      if $\mathrm{sum}(aux) = 0$ **then**
        $\theta \leftarrow \mathrm{angle}(\mathbf{a}_{\wp_m(k)}, \mathcal{D}_\mathcal{P}(i))$;
        if $\theta < \theta_{min}$ **then**
          $\theta_{min} \leftarrow \theta$;
          $index \leftarrow i$;
        end if
      end if
    end for
    approximations$(\wp_m(k), :, m) \leftarrow \mathcal{D}_\mathcal{P}(index)$;
  end for
end for

4.4 APPROXIMATIONS FOR COMPLEX-VALUED MATRICES

The introduced method is based on the calculation of the angle between two vectors whose elements are in $\mathbb{R}$: a row of the input matrix and the candidate approximate vector. However, one of the transforms we are considering is the DFT, which has its coefficients defined over the complex space, $\mathbb{C}$. In this case, algorithm modifications are necessary. Next, we describe two procedures, I and II, to obtain complex-valued approximations using the proposed

method. Each of them offers a way to decompose the complex-valued problem into real-valued problems that are suitable for the application of the proposed method.

4.4.1 Procedure I

A natural first option is to decompose the complex matrix in its real and complex components and approximate each one using any version (unconstrained or constrained) of the proposed method. Then, the DFT approximations are combinations of the approximations found for the real and complex components. Notice that, if using the constrained version of the method, the approximations for the real and complex parts are going to be orthogonal, but there is no guarantee that the resulting approximate matrices are orthogonal as well.

4.4.2 Procedure II

Another approach for approximating complex-values matrices is to calculate the angle in the complex space. According to Scharnhorst (SCHARNHORST, 2001), one way of computing the angle between two M-dimensional complex vectors, say $\mathbf{r}$ and $\mathbf{s}$, is by considering its isometric vector space $\mathbb{R}^{2M}$.

Then, the angle between $\mathbf{r}$ and $\mathbf{s}$ is calculated as in (4.2) with

$$\text{angle}(\mathbf{r}, \mathbf{s}) = \text{angle}(\mathbf{r}^\star, \mathbf{s}^\star),$$

where $\mathbf{r}^\star$ and $\mathbf{s}^\star$ are defined in $\mathbb{R}^{2N}$ by the relation

$$r^\star_{2k} = \Re(r_k) \text{ and } r^\star_{2k+1} = \Im(r_k),\ k = 0, 1, \ldots, M-1. \tag{4.6}$$

The inverse operation is given by:

$$r_k = r^\star_{2k} + j r^\star_{2k+1}\ k = 0, 1, \ldots, M-1. \tag{4.7}$$

Let $\mathbf{A}$ be an $N \times M$ matrix whose coefficients are complex, that is, $a_{i,j} \in \mathbb{C}$, $i = 0, 1, \ldots, N-1$, $j = 0, 1, \ldots, M-1$. By performing the mapping on (4.6) for each row of $\mathbf{A}$, we obtain a new real matrix $\mathbf{B}$ with dimensions $N \times 2M$, i.e.

$$\text{Complex } N \times M \text{ matrix } \mathbf{A} \xrightarrow{(4.6)} \text{ Real } N \times 2M \text{ matrix } \mathbf{B}. \tag{4.8}$$

Then, $\mathbf{B}$ can be approximated by any version of the proposed method as they were described earlier. Next, each approximate matrix obtained, $\hat{\mathbf{B}}$, must be converted from real to complex again by applying (4.7) to each of its rows, i.e.

$$\text{Real } N \times 2M \text{ matrix } \hat{\mathbf{B}} \xrightarrow{(4.7)} \text{ Complex } N \times N \text{ matrix } \hat{\mathbf{A}}. \tag{4.9}$$

The matrices obtained from (4.9) are the complex approximations for the input matrix $\mathbf{A}$.

In this case, the constrained version of the proposed method can not guarantee orthogonality of the approximate matrices obtained. The approximations for the real $N \times 2M$ matrix are going to be orthogonal. However, there is nothing that assures that its rows are still going to be orthogonal after being converted back to complex vectors. Also, due the mapping in (4.8), we are now approximating $2M$-dimensional vectors. Therefore, the cardinality of the search space is now given by $|\mathcal{D}_{\mathcal{P}}| = |\mathcal{P}|^{2M}$.

Table 6 summarizes the modifications in each version of the method so they can approximate complex matrices.

	Angle based method	
Procedure for complex-valued approximation	Unconstrained	Constrained to orthogonality
Procedure I	• Separate $\mathbf{A}$ into its real and complex components • Run Algorithm 1 for both components • Combine the approximations obtained for the real and complex components	• Separate $\mathbf{A}$ into its real and complex components • Run Algorithm 2 for both components • Combine the approximations obtained for the real and complex components
Procedure II	• Apply (4.8) to $\mathbf{A}$ • Run Algorithm 1 • Apply (4.9) to the approximations obtained	• Apply (4.8) to $\mathbf{A}$ • Run Algorithm 2 • Apply (4.9) to the approximations obtained

Tabela 6 – Procedures to approximate complex matrices using the unconstrained and constrained versions of the angle based method.

Since our focus here is to approximate the 8-point DCT, we have no data to support a suggestion of which procedure for complex-valued matrix approximation is better in terms of the obtained approximation quality. However, in terms of complexity, Procedure I, which

requires the approximation of two $N \times M$ matrices, seems to be a better option. Since the size of the search space grows exponentially, it is likely that the processing time to find the approximations also grow exponentially as M increases, which makes the processing time to approximate a $N \times 2M$ matrix significantly larger than the processing time to approximate two $N \times M$ matrices.

4.5 REMARKS

Here we list some observations about the proposed method. Some of the following notes are just general considerations while other points are going to be further explored in the next chapter.

4.5.1 General remarks

- If $\mathbf{A}$ already has any low-complexity rows, they might be previously fixed and any of the methods can be used to approximate only the remaining rows. This reduces processing time;
- Matrix symmetries may also be explored in order to reduce computational time.

4.5.2 Unconstrained version of the proposed method

- Since the unconstrained version of the method does not have to consider the search sequence, it is faster than the constrained version;
- The rows are approximated independently, which allows the use of parallelization in order to run it even faster;
- By construction, the approximations generated by a specific search space are all different, although different search spaces may generate the same approximations.

4.5.3 Constrained to orthogonality version of the proposed method

- For a particular search sequence, the algorithm may reach a point where it can not find a vector in the search space that is orthogonal to the vectors already fixed in the approxi-

mation matrix. From that point on, all the rows still to be approximated are going to be set as null vectors in the approximate matrix;

- Some search sequences may generate the same approximation matrix;
- As for the two items above, the 3-dimensional array obtained from the method must be "cleaned" in order to eliminate the singular matrices and the repeated ones. Only the remaining matrices are actually valid approximations;
- Notice in Algorithm 2 we only change the candidate vector in the approximate matrix if the angle between this vector and the matrix row is smaller than the previous minimum angle. By doing so, we fix in the approximated matrix the first vector in the search space which generates that minimum angle. Changing this condition may generate a different approximate matrix.

5 APPROXIMATION SCHEMES

As discussed in the end of the previous chapter, some features of the matrix we aim at approximating may be used in order to reduce the computational complexity of the approximation process. We show here: (i) how the search space might be reduced by only considering the positive elements of $\mathcal{P}$; (ii) how the native low-complexity rows of the discrete transforms we are considering can be used to reduce the number of rows to be approximated; and (iii) how to explore the symmetries of the discrete transform matrices in order to further reduce the complexity of the approximation process. Finally, we define the combination of versions of the method and features explored to reduce the computational complexity of the approximation process as the approximation schemes we can use to approximate the DFT, DHT and DCT.

5.1 SEARCH SPACE REDUCTION

The search space is generated from a set of low-complexity elements. Some common choices for this set and the size of the corresponding search spaces when approximating an $N \times M$ matrix are displayed in Table 7.

Set ($\mathcal{P}$)	Size of the corresponding search space
$\{-1, 0, 1\}$	3^M
$\{-2, -1, 0, 1, 2\}$	5^M
$\{-1, -1/2, 0, 1/2, 1\}$,	5^M
$\{-2, -1, -1/2, 0, 1/2, 1, 2\}$	7^M

Tabela 7 – Examples of common sets and the size of the corresponding search space

Observing the sets shown in Table 7 we can see that they are all symmetric around zero. It is important to have those negative and positive elements since the target matrices also have positive and negative entries. In this sense, one way to reduce the size of the search space, would be to consider only the nonnegative elements of those sets. Then, after the approximation process, restore the element signs according to the sign pattern from the exact matrix. Therefore, we propose the following procedure to approximate an input matrix $\mathbf{A}$:

1. Select a set $\mathcal{P}$ and remove its negative elements, e.g.: $\mathcal{P}^+ = \{0, 1, 2\}$;

2. Approximate $\mathrm{abs}(\mathbf{A})$ using the unconstrained version of the method, where $\mathrm{abs}(\cdot)$ returns the absolute value of its input. When applied to matrices, the abs function is an elementwise operation (SEBER, 2008);

3. Define the approximations for $\mathbf{A}$ as

$$\widehat{\mathbf{A}} = \widehat{\mathrm{abs}(\mathbf{A})} \odot \mathrm{sign}(\mathbf{A}), \tag{5.1}$$

where $\widehat{\mathrm{abs}(\mathbf{A})}$ is an approximation obtained from step 2, and $\odot$ represents the element wise multiplication.

By performing the procedure above, the size of the search space is reduced and the sign structure of the input matrix is preserved. As an example, Table 8 shows the proportional reduction of the search space when $\mathbf{A}$ is an $N \times 8$ matrix.

Set	Size of the original search space (Table 7)	Size of the reduced search space	Reduction of the search space
$\{0, 1\}$	$3^8 \approx 6.56 \times 10^3$	$2^8 = 2.56 \times 10^2$	96.10%
$\{0, 1, 2\}$	$5^8 \approx 3.90 \times 10^5$	$3^8 \approx 6.56 \times 10^3$	98.32%
$\{0, 1/2, 1\}$,	$5^8 \approx 3.90 \times 10^5$	$3^8 \approx 6.56 \times 10^3$	98.32%
$\{0, 1/2, 1, 2\}$	$7^8 \approx 5.76 \times 10^6$	$4^8 \approx 6.55 \times 10^4$	98.86%

Tabela 8 – Reduction of the size of the search space for some sets when $M = 8$

Note that the proposed procedure above can be used only in association with the unconstrained version of the method. This is due to the fact that for the unconstrained version the rows are approximated independently. Observe that, for the proposed procedure, we are approximating a nonnegative matrix with nonnegative elements. In this case, two row vectors are orthogonal if, and only if, they have nonzero elements in different positions, which causes the inner product to be zero. Thus, if using this procedure associated with the constrained version of the method, we have the following possible situations:

- If a row is approximated by a vector with more than one nonzero element, the output matrix is necessarily going to have at least one all zero row. In this case, it means that the matrix is singular and it is not interesting for our purposes;

- Otherwise, the output matrix is going to be a permuted and/or scaled version of the identity matrix, which is also not interesting for decorrelation purposes.

Therefore, for the constrained version of the method it is necessary to consider the original sets and the search space remains the same size.

5.2 FIXING LOW-COMPLEXITY ROWS

Notice that, if a given matrix already has low-complexity rows, then those rows do not need to be approximated. Therefore, the proposed method is only applied to the remaining rows.

The DFT, DHT, and DCT all have native low-complexity rows. Next, we identify for each considered transform which are these low-complexity rows that can be disregarded by the approximation algorithm.

5.2.1 DFT and DHT

The DFT and DHT have the same low-complexity rows, which is expected given their similar kernels. The kernels of both transforms are built as a combination of cosine and sine functions with their argument being $\frac{2\pi ik}{N}$. Then, for those two transforms, the low-complexity rows are the ones for which $k = 0, N/4, N/2, 3N/4$. For each of these values we have that

- If $k = 0$, then $\frac{2\pi ik}{N} = 0$;
- If $k = \frac{N}{4}$, then $\frac{2\pi ik}{N} = \frac{\pi}{2}i$;
- If $k = \frac{N}{2}$, then $\frac{2\pi ik}{N} = \pi i$;
- If $k = \frac{3N}{4}$, then $\frac{2\pi ik}{N} = \frac{3\pi}{2}i$.

Table 9 displays the sequences generated by the cosine ans sine functions when their argument are the ones obtained above and $i = 0, 1, 2, \ldots$.

As seen in Table 9, all the sequences have elements on the low-complexity set $\{-1, 0, 1\}$. For the DFT, its real and complex parts are given by the cosine and sine sequences shown in Table 9, respectively, as shown in Figure 4.

k	Argument	Generated sequences	
0	0	$\cos(0)$	$1, 1, 1, 1, 1, 1, 1, \ldots$
		$\sin(0)$	$0, 0, 0, 0, 0, 0, 0, \ldots$
$\frac{N}{4}$	$\frac{\pi}{2}i$	$\cos(\frac{\pi}{2}i)$	$1, 0, -1, 0, 1, 0, -1, \ldots$
		$\sin(\frac{\pi}{2}i)$	$0, 1, 0, -1, 0, 1, 0, \ldots$
$\frac{N}{2}$	πi	$\cos(\pi i)$	$1, -1, 1, -1, 1, -1, 1, \ldots$
		$\sin(\pi i)$	$0, 0, 0, 0, 0, 0, 0, \ldots$
$\frac{3N}{4}$	$\frac{3\pi}{2}i$	$\cos(\frac{3\pi}{2}i)$	$1, 0, -1, 0, 1, 0, -1, \ldots$
		$\sin(\frac{3\pi}{2}i)$	$0, -1, 0, 1, 0, -1, 0, \ldots$

Tabela 9 – Cosine and sine sequences generated when $k = 0, N/4, N/2, 3N/4$

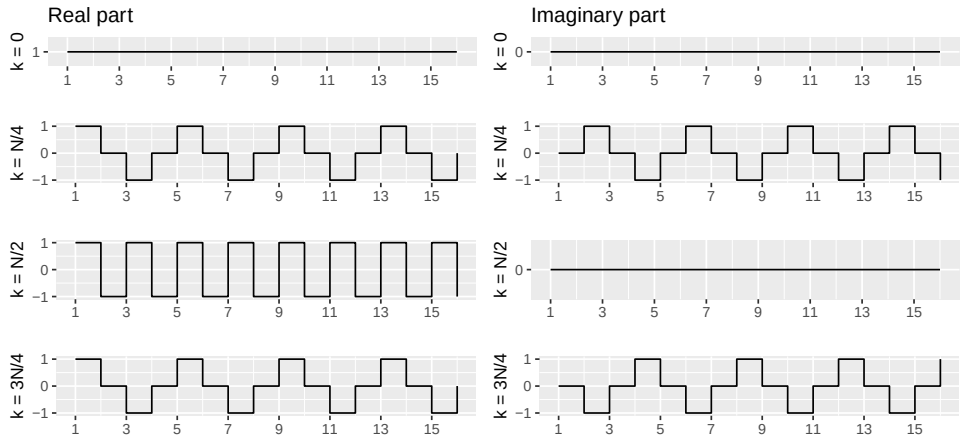


Figura 4 – Graphic representation of the real and imaginary parts of the low-complexity sequences that form the rows of the DFT matrix for $k = 0, N/4, N/2, 3N/4$.

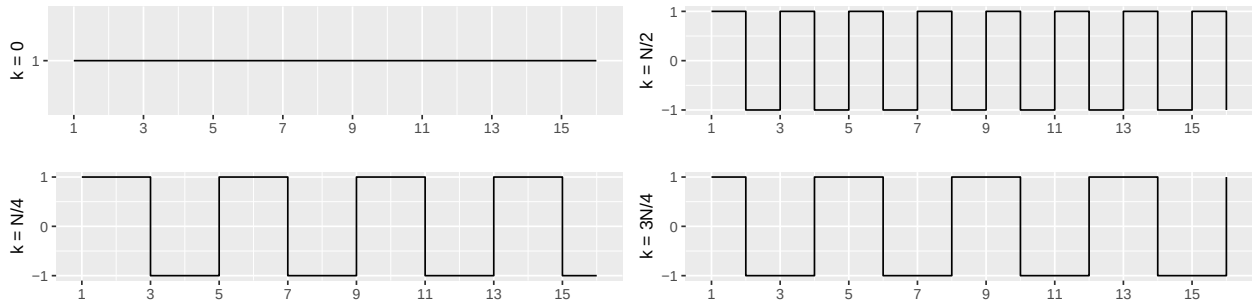


Figura 5 – Graphic representation of the low-complexity sequences that form the rows of the DHT matrix for $k = 0, N/4, N/2, 3N/4$.

For the DHT, its low-complexity rows are going to be the sum of those sequences. The graphic representation of those sequences is shown in Figure 5.

5.2.2 DCT

The DCT matrix does not have low-complexity rows. However, for $k = 0, \frac{N}{2}$, the DCT rows are scaled low-complexity sequences, as show in Table 10. In this case, we can define

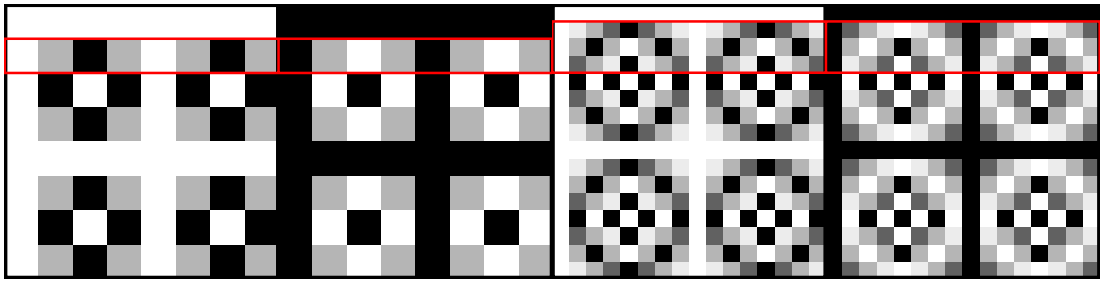
k	$c_{k,m}$	Row sequence ($m = 0, 1, 2, \ldots$)
0	$\frac{1}{\sqrt{N}}$	$\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots = \frac{1}{\sqrt{N}}(1, 1, 1, 1, 1, 1, 1 \ldots)$
$N/2$	$\sqrt{\frac{2}{N}}\cos((2m+1)\frac{\pi}{4})$	$\frac{1}{\sqrt{N}}, -\frac{1}{\sqrt{N}}, -\frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}}, \ldots = \frac{1}{\sqrt{N}}(1, -1, -1, 1, 1, -1, -1 \ldots)$

Tabela 10 – DCT row sequence for $k = 0, N/2$

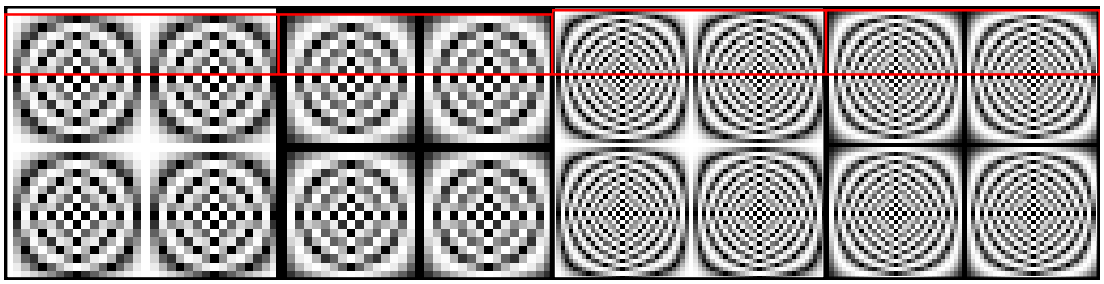
those rows as the low-complexity sequences in Table 10.

5.3 UNCONSTRAINED VERSION OF THE METHOD

For the unconstrained version of the proposed method, we can not only fix some rows but combine this with the search space reduction procedure proposed on the previous section. Figures 6, 7, and 8 display the image representation of the absolute value of the transforms considered for $N = 8, 16, 32, 64$.

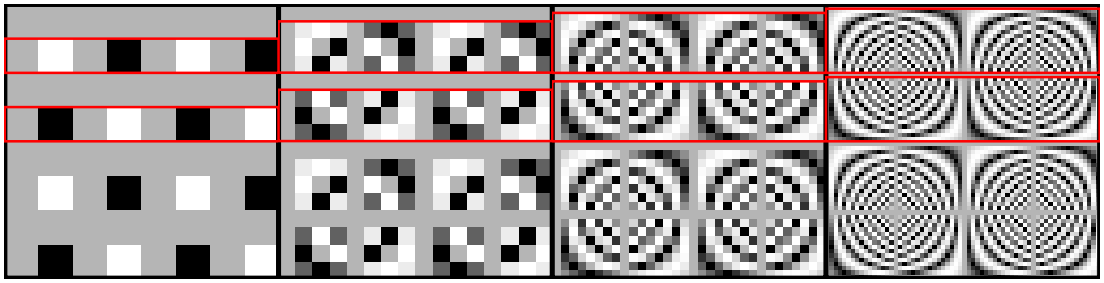


(a) $\mathrm{abs}(\Re(\mathbf{F}_8))$ (b) $\mathrm{abs}(\Im(\mathbf{F}_8))$ (c) $\mathrm{abs}(\Re(\mathbf{F}_{16}))$ (d) $\mathrm{abs}(\Im(\mathbf{F}_{16}))$



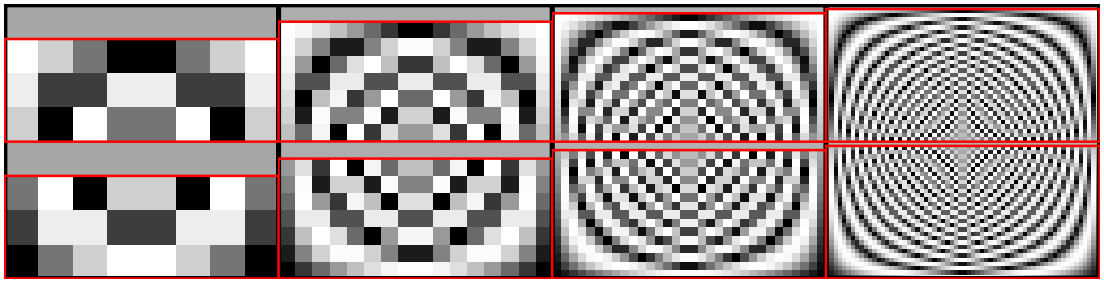
(e) $\mathrm{abs}(\Re(\mathbf{F}_{32}))$ (f) $\mathrm{abs}(\Im(\mathbf{F}_{32}))$ (g) $\mathrm{abs}(\Re(\mathbf{F}_{64}))$ (h) $\mathrm{abs}(\Im(\mathbf{F}_{64}))$

Figura 6 – Image representation for the absolute value of the real and complex parts of the DFT matrix for $N = 8, 16, 32, 64$.



(a) $\mathrm{abs}(\mathbf{H}_8)$ (b) $\mathrm{abs}(\mathbf{H}_{16})$ (c) $\mathrm{abs}(\mathbf{H}_{32})$ (d) $\mathrm{abs}(\mathbf{H}_{64})$

Figura 7 – Image representation for the absolute value of the DHT transform matrix for $N = 8, 16, 32, 64$.



(a) $\mathrm{abs}(\mathbf{C}_8)$ (b) $\mathrm{abs}(\mathbf{C}_{16})$ (c) $\mathrm{abs}(\mathbf{C}_{32})$ (d) $\mathrm{abs}(\mathbf{C}_{64})$

Figura 8 – Image representation for the absolute value of the DCT transform matrix for $N = 8, 16, 32, 64$.

In the images on Figures 6, 7, and 8, it is possible to identify the low-complexity rows we can previously fix on the approximation matrix. In particular, for the DFT and DHT matrices, we can also see that some of the remaining rows are repeated. For example, in $\mathrm{abs}(\Re(\mathbf{F}_{16}))$ (Figure 6(c)), rows $k = 1, 7, 9, 15$ are the same. Note that if $\mathrm{abs}(\mathbf{A})$ has repeated rows, then, by definition of the unconstrained version of the method, the optimal solutions for those rows are going to be the same. As a consequence, we only need to approximate the unique rows. In summary, if (i) using the unconstrained version; (ii) using the search space reduction procedure proposed; and (iii) fixing the low-complexity rows already in the transform matrix; it is only necessary to approximate the rows highlighted in Figures 6, 7, and 8.

5.3.1 Matrix symmetries

Looking a little bit further into Figures 6, 7, and 8, it is possible to identify, inside the highlighted regions, some symmetry patterns. That means it is possible to approximate only a portion of those rows and obtain the whole matrix by reflexions on the rows, columns, or both,

of the approximated partition.

5.3.1.1 DFT and DHT

Notice that for the absolute value of the DFT and DHT in Figures 6 and 7, that are not only low-complexity rows but also low-complexity columns, $i = 0, N/4, N/2, 3N/4$. Those columns may also be fixed previously on the approximation matrix. Figures 9 and 10 show which portion of the highlighted rows in Figures 6 and 7 we need to approximate. That is, which portion of the matrix we need to approximate in order to obtain the whole matrix apart from the low-complexity rows and columns already existent in the original matrix.

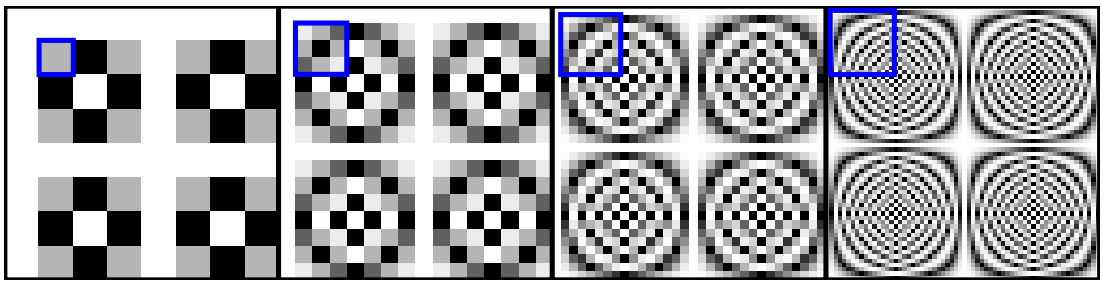


Figura 9 – Portion of the DFT matrix to be approximated for $N = 8, 16, 32, 64$.

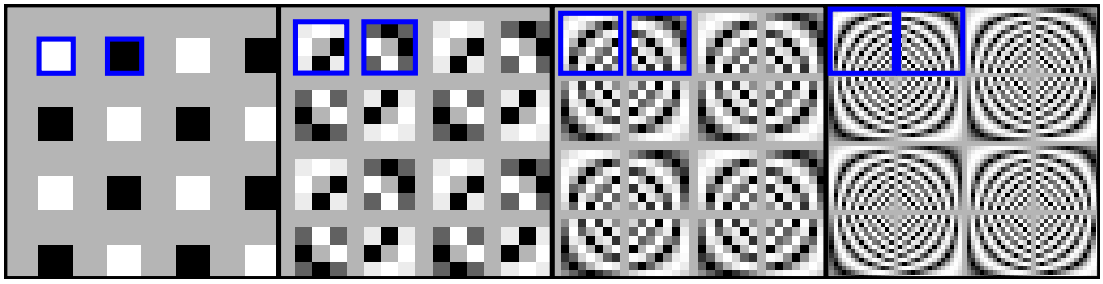


Figura 10 – Portion of the DHT matrix to be approximated for $N = 8, 16, 32, 64$.

5.3.1.2 DCT

For the DCT , there is no low-complexity columns that can be previously fixed. However, there are symmetries in the rows that can be explored, as shown in Figure 11.

Table 11 summarizes the information about the rows and columns to be approximated for both versions of the method considering all the possible modifications presented above to reduce the complexity of the approximation process. Table 12 also gives the same information

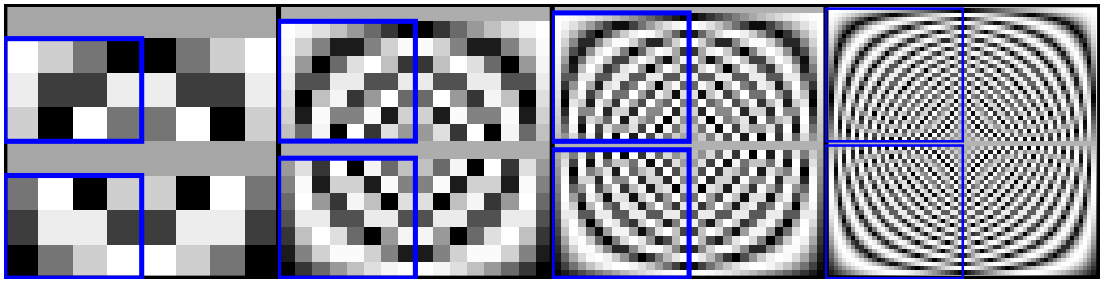


Figura 11 – Portion of the DCT matrix to be approximated for $N = 8, 16, 32, 64$.

of Table 11 for the case when we are only considering previously fixing the low-complexity rows. Table 13 summarizes which of the modifications presented above may be used for each

Transform	Rows to be approximated	Columns to be approximated	Number of rows to be approximated	Number of columns to be approximated
DFT	1 to $N/4-1$	1 to $N/4-1$	$N/4-1$	$N/4-1$
DHT	1 to $N/4-1$	1 to $N/4-1$ and $N/4+1$ to $N/2-1$	$N/4-1$	$N/2-2$
DCT	1 to $N/2-1$ and $N/2+1$ to $N-1$	0 to $N/2-1$	$N-2$	$N/2$

Tabela 11 – Summary of the rows and columns to be approximated when using the unconstrained version of the proposed method considering all the possible modifications to reduce the approximation procedure.

Transform	Fixed Rows	Columns to be approximated	Number of rows to be approximated	Number of columns to be approximated
DFT	$0, N/4, N/2, 3N/4$	All columns	$N-4$	N
DHT	$0, N/4, N/2, 3N/4$	All columns	$N-4$	N
DCT	$0, N/2$	All columns	$N-2$	N

Tabela 12 – Summary of the rows and columns to be approximated when using the constrained version of the proposed method and previously fixing the low-complexity rows of the original matrix.

method.

It is noteworthy that all the modifications can actually be used in association with the constrained version of the proposed method. But, in our case (the matrices we are interested are

	Version of the method	
	Unconstrained	Constrained to orthogonality
Search space reduction	✓	✗
Fix low-complexity rows	✓	✓ (only if the input matrix is orthogonal)
Explore matrix symmetries	✓	✗
Complex adaptation	✓	✗

Tabela 13 – Comparison of the unconstrained and constrained versions of the proposed method in terms of the complexity reduction procedures they admit.

orthogonal), only previously fixing the low-complexity rows guarantees that the output matrix is orthogonal (which is the whole point of this version of the method).

5.4 APPROXIMATION SCHEMES

Based on the discussion above, we can define the approximation schemes that can be used to approximate the DCT, DHT, and DFT. The DHT and DCT can be approximated using Schemes I and II, displayed in Figure 12. For the DFT, Schemes III and IV in Figure 13, which consider the complex adaptation, can be used.

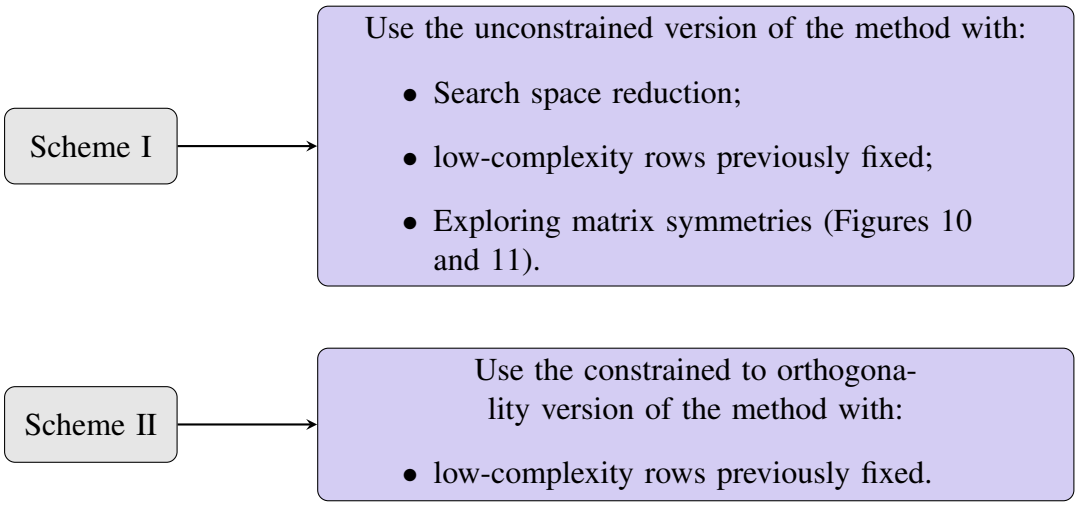


Figura 12 – Approximation schemes for the DHT and DCT.

Notice that the approximation schemes proposed for the DFT consider only the unconstrained version of the method. This is because none of the complex adaptations when used with the constrained version guarantees that the output matrices obtained are orthogonal. Then, the use of the constrained version loses its point. Also, when considering the complex adaptation A, which means we are going to approximate the real and imaginary parts independently, we

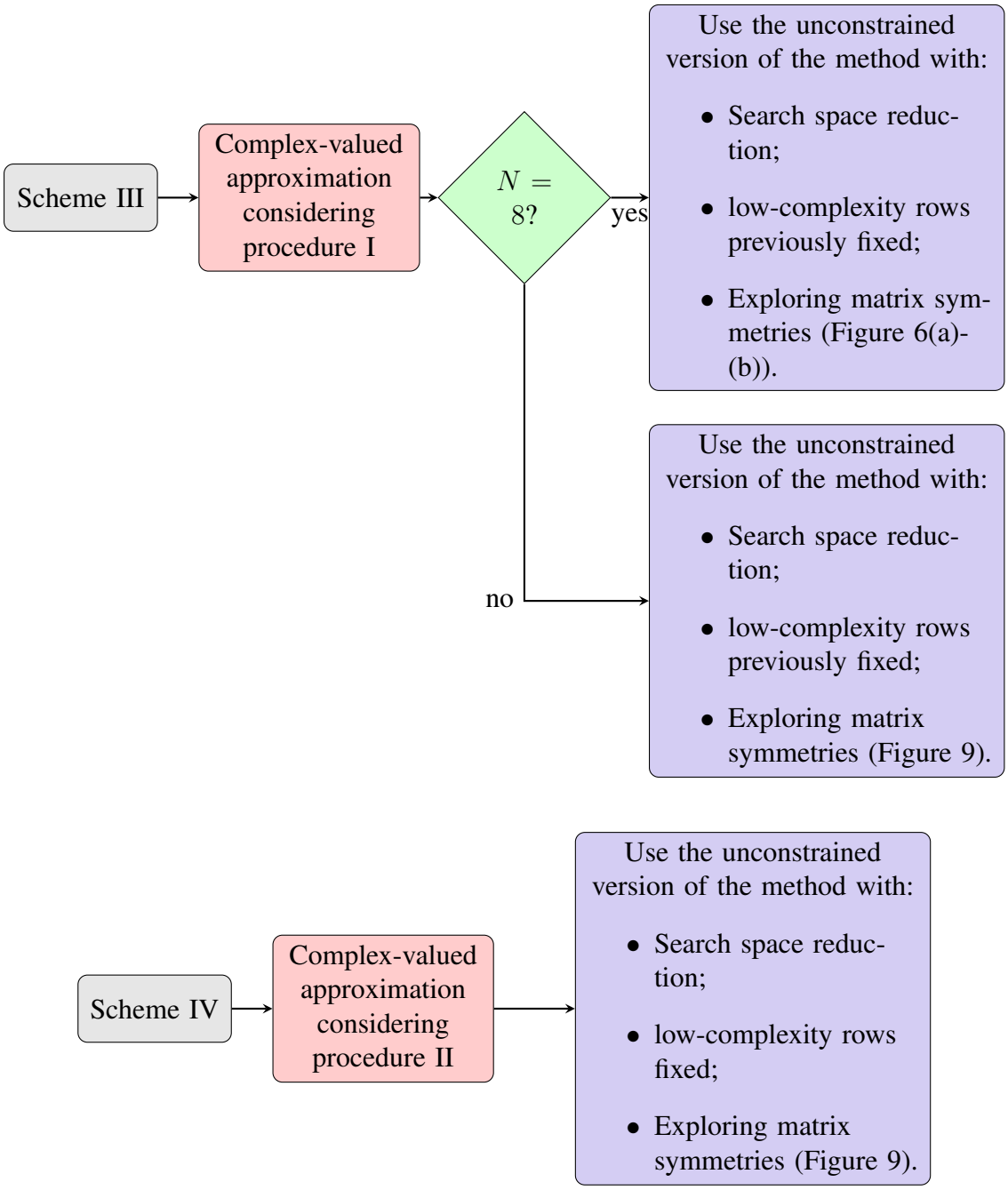


Figura 13 – Approximation schemes for the DFT.

need to verify if the size of the matrix is $N = 8$. In this case, we have two 8×8 matrices to approximate and, if we check the portion of those matrices that need to be approximated (Figure 9(a)), we can see that this region is reduced to a single element, not a vector, which is required by the method. Then, when $N = 8$, we are going to approximate the entire second row, as in Figure 6(a)-(b). For the approximation Scheme IV this is not a concern, since when $N = 8$, the 8×8 initial matrix is converted in a 8×16 matrix by the mapping in equation (4.6)

and that single element is now a vector with two elements (the real and imaginary parts of the original complex element).

In the next chapter, the approximation Schemes I and II are applied to obtain new approximations for the 8-point DCT.

6 NEW ANGLE-BASED APPROXIMATIONS FOR THE 8-POINT DCT

6.1 FIGURES OF MERIT

To evaluate the performance of the proposed approximations, we selected traditional figures of merit

- total error energy ($\epsilon(\cdot)$) (CINTRA; BAYER, 2011);
- mean square error ($\mathrm{MSE}(\cdot)$) (BRITANAK; YIP; RAO, 2007; WANG; BOVIK, 2009);
- coding gain ($C_g(\cdot)$) (BRITANAK; YIP; RAO, 2007; GOYAL, 2001; KATTO; YASUDA, 1991);
- transform efficiency ($\eta(\cdot)$) (BRITANAK; YIP; RAO, 2007).

The MSE and total error energy are suitable measures to quantify the difference between the exact DCT and its approximations (BRITANAK; YIP; RAO, 2007). The coding gain and transform efficiency are appropriate tools to quantify compression, redundancy removal, and data decorrelation capabilities (BRITANAK; YIP; RAO, 2007). Additionally, since for the unrestricted version of the method there is no guarantee of orthogonality, we also considered the orthogonality deviation measure (FLURY; GAUTSCHI, 1986).

Hereafter we adopt the following quantities and notation: the interpixel correlation is $\rho = 0.95$ (BRITANAK; YIP; RAO, 2007; GOYAL, 2001; LIANG; TRAN, 2001), $\widehat{\mathbf{C}}$ is an approximation for the DCT, and $\widehat{\mathbf{R}}_{\mathbf{y}} = \widehat{\mathbf{C}} \cdot \mathbf{R}_{\mathbf{x}} \cdot \widehat{\mathbf{C}}^{\top}$, where $\mathbf{R}_{\mathbf{x}}$ is the covariance matrix of $\mathbf{x}$, whose elements are given by $\rho^{|i-j|}$, $i, j = 1, 2, \ldots, 8$. We detail each of these measures below.

6.1.1 Total Energy Error

The total energy error is a similarity measure given by (CINTRA; BAYER, 2011):

$$\epsilon(\widehat{\mathbf{C}}) = \pi \cdot \|\mathbf{C} - \widehat{\mathbf{C}}\|_{\mathrm{F}}^2,$$

where $\|\cdot\|_{\mathrm{F}}$ represents the Frobenius norm (WATKINS, 2004).

6.1.2 Mean Square Error

The MSE of a given approximation $\widehat{\mathrm{C}}$ is furnished by (BRITANAK; YIP; RAO, 2007; WANG; BOVIK, 2009):

$$\mathrm{MSE}(\widehat{\mathbf{C}}) = \frac{1}{8} \cdot \mathrm{tr}\left((\mathbf{C} - \widehat{\mathbf{C}}) \cdot \mathbf{R}_{\mathbf{x}} \cdot (\mathbf{C} - \widehat{\mathbf{C}})^{\top} \right),$$

where $\mathrm{tr}(\cdot)$ represents the trace operator (BRITANAK; YIP; RAO, 2007). The total energy error and the mean square error are appropriate measures for capturing the approximation error in a Euclidean distance sense.

6.1.3 Coding Gain

The coding gain quantifies the energy compaction capability and is given by (BRITANAK; YIP; RAO, 2007):

$$C_g(\widehat{\mathbf{C}}) = 10 \cdot \log_{10} \left\{ \frac{\frac{1}{8} \sum_{i=1}^{8} r_{i,i}^2}{\left(\prod_{i=1}^{8} r_{i,i}^2 \cdot \|\widehat{\mathbf{c}}_i\|^2 \right)^{1/8}} \right\},$$

where $r_{i,i}$ is the ith element of the diagonal of $\widehat{\mathbf{R}}_{\mathbf{y}}$ (BRITANAK; YIP; RAO, 2007) and $\widehat{\mathbf{c}}_i$ is the ith row of $\widehat{\mathbf{C}}$.

However, as pointed in (KATTO; YASUDA, 1991), the previous definition is suitable for orthogonal transforms only. For nonorthogonal transforms, such as SDCT (HAWEEL, 2001) and MRDCT (BAYER; CINTRA, 2012), we adopt the unified coding gain (KATTO; YASUDA, 1991). For $i = 1, 2, \ldots, 8$, let $\widehat{\mathbf{c}}_i$ and $\widehat{\mathbf{g}}_i$ be the ith row of $\widehat{\mathbf{C}}$ and $\widehat{\mathbf{C}}^{-1}$, respectively. Then, the unified coding gain is given by

$$C_g^{*}(\widehat{\mathbf{C}}) = 10 \cdot \log_{10} \left\{ \prod_{i=1}^{8} \frac{1}{\sqrt[8]{A_i \cdot B_i}} \right\},$$

where $A_i = \mathrm{su}\left[\left(\widehat{\mathbf{c}}_i^{\top} \cdot \widehat{\mathbf{c}}_i \right) \odot \mathbf{R}_x \right]$, $\mathrm{su}(\cdot)$ returns the sum of all elements of the input matrix, the operator $\odot$ represents the elementwise product, and $B_i = \|\widehat{\mathbf{g}}_i\|^2$.

6.1.4 Transform Efficiency

The transform efficiency is an alternative measure to the coding gain, being expressed according to (BRITANAK; YIP; RAO, 2007)

$$\eta(\widehat{\mathbf{C}}) = \frac{\sum_{i=1}^{8} |r_{i,i}|}{\sum_{i=1}^{8}\sum_{j=1}^{8} |r_{i,j}|} \cdot 100,$$

where $r_{i,j}$ is the (i,j)th entry of $\widehat{\mathbf{R}}_{\mathbf{y}}$, $i,j = 1,2,\ldots,8$ (BRITANAK; YIP; RAO, 2007).

6.1.5 Orthogonality deviation

The orthogonality deviation (FLURY; GAUTSCHI, 1986) is a measure to quantify how close a matrix is from a diagonal matrix. It is given by:

$$\delta(\mathbf{T}) = 1 - \frac{\|\operatorname{diag}(\mathbf{T})\|_{\mathrm{F}}^2}{\|\mathbf{T}\|_{\mathrm{F}}^2}.$$

6.2 IMPORTANT DEFINITIONS

A large number of new approximations were obtained considering the approximation schemes introduced in the previous chapter. In order to optimize the presentation of those approximate matrices, the following definitions are required.

Definition 6.1 (Equivalence)**.** *We say that two matrices are equivalent to each other when they present the same results for a set of evaluation metrics considered.*

Definition 6.2 (Class of equivalence)**.** *A set of matrices equivalent to each other form a class of equivalence.*

Although for some cases the number of approximate matrices obtained was large, we were able to identify a reduced number of classes of equivalence. Then, instead of presenting all the matrices obtained, we present only one representative of each class of equivalence. The metrics that define the equivalence relationship between two matrices, the metric to select the representative matrix of each class, and the results obtained are discussed in the next section.

6.3 NEW APPROXIMATIONS

The new approximations were obtained running approximations Schemes I and II, which are the appropriate ones for the DCT, as explained in Chapter 5. The low-complexity sets considered to generate the search spaces are displayed in Table 14.

Set	Set elements
$\mathcal{P}_1$	$\{-1, 0, 1\}$
$\mathcal{P}_2$	$\{-1, -\frac{1}{2}, 0, -\frac{1}{2}, 1\}$
$\mathcal{P}_3$	$\{-2, -1, 0, 1, 2\}$
$\mathcal{P}_4$	$\{-3, -1, 0, 1, 3\}$
$\mathcal{P}_5$	$\{-1, -\frac{1}{2}, -\frac{1}{4}, 0, -\frac{1}{4}, -\frac{1}{2}, 1\}$
$\mathcal{P}_6$	$\{-2, -1, -\frac{1}{2}, 0, -\frac{1}{2}, 1, 2\}$
$\mathcal{P}_7$	$\{-3, -1, -\frac{1}{2}, 0, -\frac{1}{2}, 1, 3\}$
$\mathcal{P}_8$	$\{-2, -1, -\frac{1}{2}, -\frac{1}{4}, 0, -\frac{1}{4}, -\frac{1}{2}, 1, 2\}$
$\mathcal{P}_9$	$\{-3, -2, -1, -\frac{1}{2}, 0, -\frac{1}{2}, 1, 2, 3\}$

Tabela 14 – Low-complexity sets considered.

From this point on, the new approximation matrices proposed in this work are going to be referred to as $\mathbf{T}_{y,z}$, which means $\mathbf{T}$ is the representative approximation of equivalence class z obtained using approximation Scheme y. For example, $\mathbf{T}_{\mathrm{I},1}$ is the representative approximation of equivalence class 1 obtained using approximation Scheme I.

The following evaluation metrics were considered to define the classes of equivalence:

- Total error energy;
- Mean square error;
- Coding gain; and
- Transform efficiency.

For approximations obtained using the approximation Scheme I, which considers the version of the method not constrained to orthogonality, the approximate matrix chosen to be the representative of each class was the one with the minimum orthogonality deviation. For the ones obtained using the approximation Scheme II, the representative matrix of each class was

the one with the minimum arithmetic complexity. Table 15 summarizes the results obtained. The actual matrices obtained are presented in Appendix A of this work.

Approximation scheme	Number of matrices obtained	Number of classes of equivalence
Scheme I	151	6
Scheme II	15	10

Tabela 15 – Total matrices and classes of equivalence obtained for the 8-point DCT.

Among the matrices obtained, three had already been introduced in literature. We verified that $\mathbf{T}_{\text{I},1} = \mathbf{T}_{\text{II},1} = \mathbf{T}_{\text{RDCT}}$ and $\mathbf{T}_{\text{II},2} = \mathbf{T}_{\text{CBT–4}}$. Thus, for further analysis we focus on the 13 new approximations obtained. Table 16 displays an overview of the representative matrices of each class of equivalence.

Approximation scheme	Class of equivalence	Representative matrix	Representative approximation	$\delta(\mathbf{T})$	Additions	Bit-shiftings
Scheme I	C2	$\mathbf{T}_{\text{I},2}$	$\widehat{\mathbf{C}}_{\text{I},2}$	0.0300	48	16
Scheme I	C3	$\mathbf{T}_{\text{I},3}$	$\widehat{\mathbf{C}}_{\text{I},3}$	0.0130	80	24
Scheme I	C4	$\mathbf{T}_{\text{I},4}$	$\widehat{\mathbf{C}}_{\text{I},4}$	0.0005	56	32
Scheme I	C5	$\mathbf{T}_{\text{I},5}$	$\widehat{\mathbf{C}}_{\text{I},5}$	0.0086	52	16
Scheme I	C6	$\mathbf{T}_{\text{I},6}$	$\widehat{\mathbf{C}}_{\text{I},6}$	0.0017	76	40
Scheme II	C3	$\mathbf{T}_{\text{II},3}$	$\widehat{\mathbf{C}}_{\text{II},3}$	0	48	24
Scheme II	C4	$\mathbf{T}_{\text{II},4}$	$\widehat{\mathbf{C}}_{\text{II},4}$	0	48	16
Scheme II	C5	$\mathbf{T}_{\text{II},5}$	$\widehat{\mathbf{C}}_{\text{II},5}$	0	80	24
Scheme II	C6	$\mathbf{T}_{\text{II},6}$	$\widehat{\mathbf{C}}_{\text{II},6}$	0	80	24
Scheme II	C7	$\mathbf{T}_{\text{II},7}$	$\widehat{\mathbf{C}}_{\text{II},7}$	0	56	32
Scheme II	C8	$\mathbf{T}_{\text{II},8}$	$\widehat{\mathbf{C}}_{\text{II},8}$	0	56	32
Scheme II	C9	$\mathbf{T}_{\text{II},9}$	$\widehat{\mathbf{C}}_{\text{II},9}$	0	72	40
Scheme II	C10	$\mathbf{T}_{\text{II},10}$	$\widehat{\mathbf{C}}_{\text{II},10}$	0	72	40

Tabela 16 – Overview of the new approximations obtained from the angle based method

In Table 17, the measurements obtained for the approximations in literature along with the results for the new approximations for the figures of merit considered to define the classes of equivalence are shown. The DCT and integer DCT (IDCT) (OHM et al., 2012) results were included as reference. The top five results for each measure are displayed in bold and were *all obtained from new approximations proposed in this work.* We can also highlight approximations $\widehat{\mathbf{C}}_{\text{I},4}$ and $\widehat{\mathbf{C}}_{\text{I},6}$, which are among the top five for all measures considered.

Approximation	$\epsilon(\widehat{\mathbf{C}})$	$\mathrm{MSE}(\widehat{\mathbf{C}})$	$C_g^*(\widehat{\mathbf{C}})$	$\eta(\widehat{\mathbf{C}})$
DCT	0	0	8.8259	93.9912
IDCT (HEVC)	0.0020	8.66×10^{-6}	8.8248	93.8236
$\widehat{\mathbf{C}}_{\text{WHT}}$	47.6126	0.2241	7.9461	85.3138
$\widehat{\mathbf{C}}_{\text{LO}}$	0.8695	0.0061	8.3902	88.7023
$\widehat{\mathbf{C}}_{\text{SDCT}}$	3.3158	0.0207	6.0261	82.6190
$\widehat{\mathbf{C}}_{\text{RDCT}}$	1.7945	0.0098	8.1827	87.4297
$\widehat{\mathbf{C}}_{\text{MRDCT}}$	8.6592	0.0594	7.3326	80.8969
$\widehat{\mathbf{C}}_{\text{BAS-2008a}}$	5.9294	0.0238	8.1194	86.8626
$\widehat{\mathbf{C}}_{\text{BAS-2008b}}$	4.1875	0.0191	6.2684	83.1734
$\widehat{\mathbf{C}}_{\text{BAS-2009}}$	6.8543	0.0275	7.9126	85.3799
$\widehat{\mathbf{C}}_{\text{BAS-2010}}$	4.0935	0.0210	8.3251	88.2182
$\widehat{\mathbf{C}}_{\text{BAS-2011}}$	26.8462	0.0710	7.9118	85.6419
$\widehat{\mathbf{C}}_{\text{BAS-2013}}$	35.0639	0.1023	7.9461	85.3138
$\widehat{\mathbf{C}}_{\text{CBT–1}}$	8.5953	0.0375	8.1361	86.8051
$\widehat{\mathbf{C}}_{\text{CBT–2}}$	1.7945	0.0100	8.1361	86.8051
$\widehat{\mathbf{C}}_{\text{CBT–3}}$	1.7945	0.0098	8.1834	87.1567
$\widehat{\mathbf{C}}_{\text{CBT–4}}$	1.7945	0.0100	8.1369	86.5359
$\widehat{\mathbf{C}}_{\text{CBT–5}}$	0.8695	0.0062	8.3437	88.0594
$\widehat{\mathbf{C}}_{\text{CBT–6}}$	3.3158	0.0208	6.0462	83.0814
$\widehat{\mathbf{C}}_{\text{CBT–7}}$	2.1473	0.0665	6.4434	63.7855
$\widehat{\mathbf{C}}_{\text{I,2}}$	**0.4022**	**0.0028**	8.4721	90.1603
$\widehat{\mathbf{C}}_{\text{I,3}}$	**0.5765**	0.0040	**8.4412**	90.5152
$\widehat{\mathbf{C}}_{\text{I,4}}$	**0.1691**	**0.0011**	**8.7184**	**91.9696**
$\widehat{\mathbf{C}}_{\text{I,5}}$	**0.4022**	**0.0028**	8.4520	**90.6123**
$\widehat{\mathbf{C}}_{\text{I,6}}$	**0.1272**	**0.0008**	**8.7654**	**92.8767**
$\widehat{\mathbf{C}}_{\text{II,3}}$	1.2194	0.0046	8.6337	90.4615
$\widehat{\mathbf{C}}_{\text{II,4}}$	1.2194	0.0127	8.1024	87.2275
$\widehat{\mathbf{C}}_{\text{II,5}}$	2.4482	0.0084	8.4301	90.5362
$\widehat{\mathbf{C}}_{\text{II,6}}$	2.4482	0.0265	7.8837	87.7395
$\widehat{\mathbf{C}}_{\text{II,7}}$	1.5452	0.0043	**8.6693**	**91.4370**
$\widehat{\mathbf{C}}_{\text{II,8}}$	1.5452	0.0176	8.0161	88.4340
$\widehat{\mathbf{C}}_{\text{II,9}}$	1.0145	**0.0029**	**8.7393**	**92.3530**
$\widehat{\mathbf{C}}_{\text{II,10}}$	1.0145	0.0114	8.1454	88.5210

Tabela 17 – Performance measures for the DCT approximations in literature and the new approximations proposed

To provide another way of visualizing the data in Table 17, we generated the plot in Figure 14. In that (i) the axis contain the information about the MSE and total error energy measures; (ii) the color express the coding gain information; and (iii) the size of each point represent the transform efficiency. The ideal transform in terms of MSE and total energy error

is the one localized at the point $(0, 0)$. The larger and bluer the point, the better the coding performance. The first thing that one can noticed is that there are two approximations, $\widehat{\mathrm{C}}_{\mathrm{BAS-2013}}$ and $\widehat{\mathrm{C}}_{\mathrm{BAS-2011}}$, that have such high MSE results that makes it hard to evaluate the differences between the remaining approximations. Therefore, we filtered the data in Figure 14 to keep only the approximations with MSE lower than 2 and total energy error lower than 0.02. The approximations that satisfy those conditions are shown in Figure 15.

In this Figure 15, we can observe that from the 31 approximations, considering the ones in literature and the proposed ones, only 17 showed up in the filtered plot. From those 17 approximations, 11 of them are approximations proposed in this work. Also, the five transforms closest to the DCT in terms of all measures considered were proposed in this work and the approximations that overall presented the best results were $\widehat{\mathrm{C}}_{\mathrm{I},4}$ and $\widehat{\mathrm{C}}_{\mathrm{I},6}$, as highlighted before.

In the next chapter, the obtained approximations are evaluated in the context of image compression.

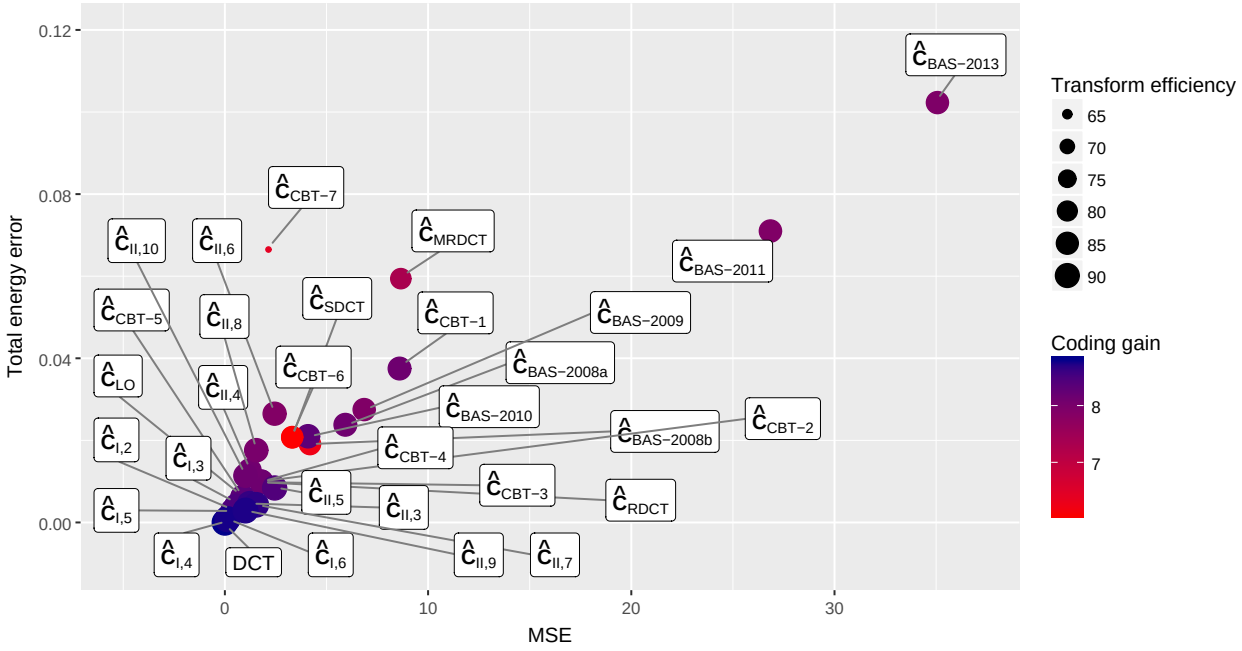


Figura 14 – Visual representation of Table 17.

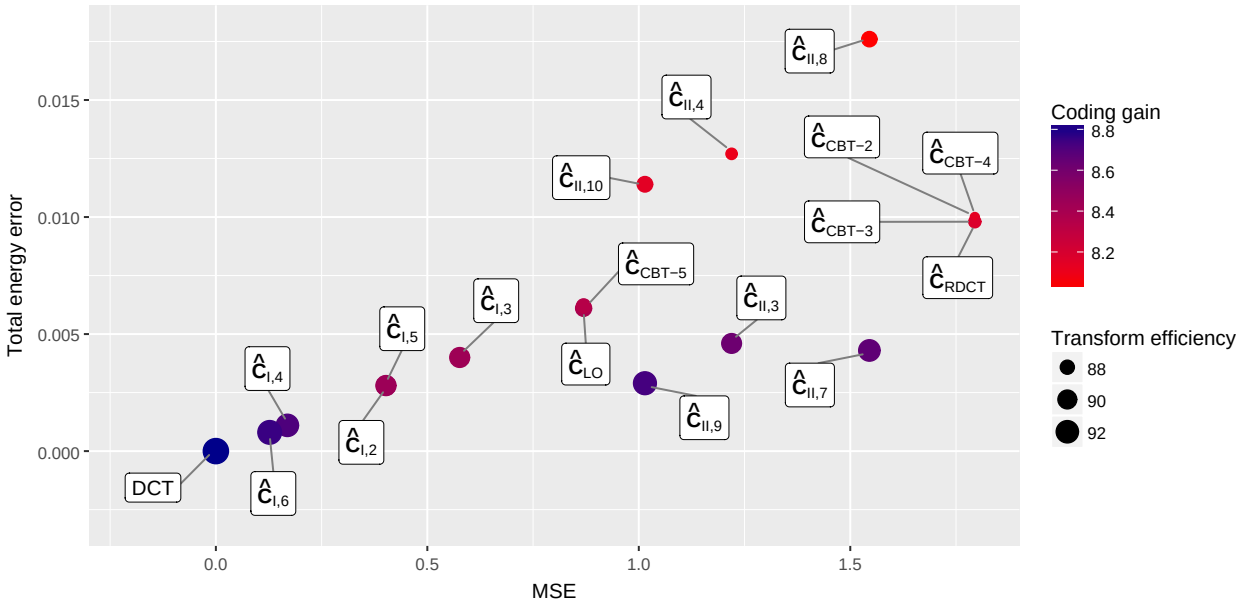


Figura 15 – Zoomed in visual representation of Table 17.

7 APPROXIMATIONS PERFORMANCE ON IMAGE PROCESSING

7.1 IMAGE COMPRESSION EXPERIMENTS

To evaluate the efficiency of the proposed transformation matrices, we performed a JPEG-like image compression experiment as described in (CINTRA; BAYER; TABLADA, 2014; POTLURI et al., 2014; BAYER; CINTRA, 2010). Input images were divided into sub-blocks of size 8×8 pixels and submitted to a bi-dimensional (2-D) transformation, such as the DCT or one of its approximations. Let $\mathbf{A}$ be a sub-block of size 8×8. The result of the 2-D transformation of $\mathbf{A}$ is an 8×8 sub-block $\mathbf{B}$ obtained as follows (CINTRA; BAYER; TABLADA, 2014; CINTRA; BAYER, 2011):

$$\mathbf{B} = \widehat{\mathbf{C}} \cdot \mathbf{A} \cdot \widehat{\mathbf{C}}^{\top}.$$

Considering the zig-zag scan pattern as detailed in (PAO; SUN, 1998) and shown in Figure 16, the initial r, $r = 1, 2, 3, \ldots, 64$, elements of $\mathbf{B}$ were retained; whereas the remaining $(64 - r)$ elements were discarded. The previous operation results in a matrix $\mathbf{B}'$ populated with

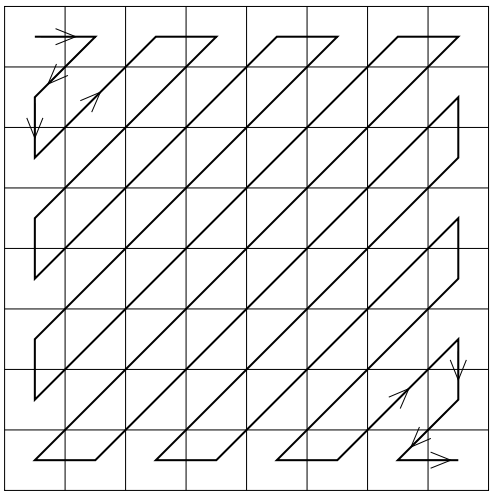


Figura 16 – Zig-zag pattern.

zeros which is suitable for entropy encoding (WALLACE, 1992). Each processed sub-block was submitted to the corresponding 2-D inverse transformation and the image was reconstructed.

The 2-D inverse transform is given by:

$$\mathbf{A} = \begin{cases} \widehat{\mathbf{C}}^{\top} \cdot \mathbf{B} \cdot \widehat{\mathbf{C}}, & \text{if } \mathbf{T} \text{ for orthogonal,} \\ \widehat{\mathbf{C}}^{-1} \cdot \mathbf{B} \cdot \left(\widehat{\mathbf{C}}^{-1}\right)^{\top}, & \text{otherwise.} \end{cases}$$

We considered 44 8-bit standardized images obtained from the USC-SIPI image bank (UNIVERSITY OF SOUTHERN CALIFORNIA,) (cf. Appendix B) and submitted them to the above described procedure. The reconstructed images were compared with the original images and evaluated quantitatively according to popular figures of merit: the mean square error (MSE) (BRITANAK; YIP; RAO, 2007), the peak signal-to-noise ratio (PSNR) (SALOMON; MOTTA; BRYANT, 2007) and the structural similarity index (SSIM) (WANG et al., 2004). We consider the MSE and PSNR measures because of its good properties and historical usage. However, as discussed in (WANG; BOVIK, 2009), the MSE and PSNR are not the best measures when it comes to predict human perception of image fidelity and quality, for which SSIM has been shown to be a better measure (WANG et al., 2004; WANG; BOVIK, 2009).

Additionally, for better visualization of the results, we considered the relative difference for each measures. The relative difference is given by:

$$\mathrm{RDiff}_{\mu} = \frac{\mu(\mathbf{C}) - \mu(\widehat{\mathbf{C}})}{\mu(\mathbf{C})} = 1 - \frac{\mu(\widehat{\mathbf{C}})}{\mu(\mathbf{C})},$$

where $\mu(\mathbf{C})$ and $\mu(\widehat{\mathbf{C}})$ indicate the exact DCT measure and the measure of an approximation, respectively, and $\mu \in \{\text{MSE, PSNR, SSIM}\}$.

For the MSE, we aim at the lowest possible results. That is, we look for approximations whose MSE is the closest possible to the DCT MSE or even smaller. In general, for the approximations in literature, $\mathrm{MSE}(\widehat{\mathbf{C}}) > \mathrm{MSE}(\mathbf{C})$ or, equivalently, $\frac{\mathrm{MSE}(\widehat{\mathbf{C}})}{\mathrm{MSE}(\mathbf{C})} > 1$. In this sense, we search for approximations such that,

$$\frac{\mathrm{MSE}(\widehat{\mathbf{C}})}{\mathrm{MSE}(\mathbf{C})} \to 1^{+},$$

where $\to 1^{+}$ represents right convergence. In other words, $\mathrm{RDiff}_{\mathrm{MSE}} \to 0$, or even $\widehat{\mathbf{C}}$ and $\mathbf{C}$ are equivalents. Ideally, we want approximations such that

$$\frac{\mathrm{MSE}(\widehat{\mathbf{C}})}{\mathrm{MSE}(\mathbf{C})} < 1.$$

That is, $\mathrm{RDiff}_{\mathrm{MSE}} > 0$, which means $\widehat{\mathbf{C}}$ presents better results than $\mathbf{C}$ in terms of MSE.

On the other hand, for the PSNR and SSIM, we aim at the largest possible values. So in this case, we want approximations such that the PSNR or SSIM are as large as DCT PSNR or SSIM or even larger. Let $\mu \in \{\text{PSNR, SSIM}\}$. Usually, approximations in literature satisfy $\mu(\widehat{\mathbf{C}}) < \mu(\mathbf{C})$, i.e., $\mu(\widehat{\mathbf{C}})/\mu(\mathbf{C}) < 1$. In this sense, we want approximations such that

$$\frac{\mu(\widehat{\mathbf{C}})}{\mu(\mathbf{C})} \to 1^{-},$$

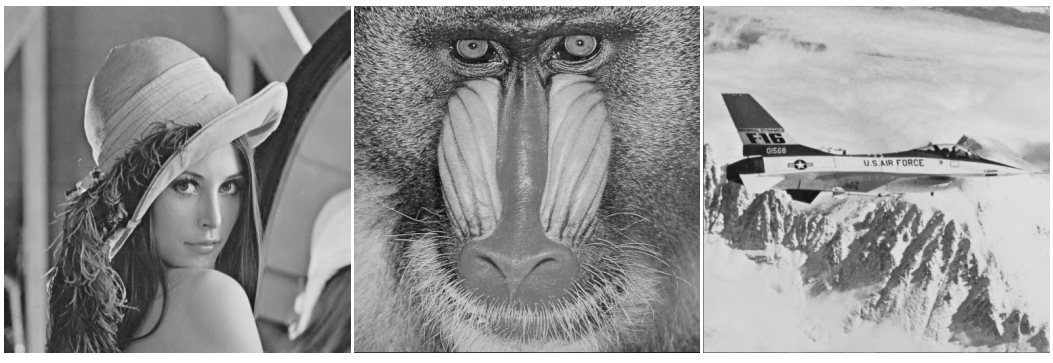
where $\to 1^{-}$ represents left convergence. That is the same as saying that $\mathrm{RDiff}_{\mu} \to 0$. Ideally, we look for approximations such that

$$\frac{\mu(\widehat{\mathbf{C}})}{\mu(\mathbf{C})} > 1,$$

which indicates $\mathrm{RDiff}_{\mu} < 0$. That means that $\widehat{\mathbf{C}}$ presents better results than $\mathbf{C}$ in terms of PSNR or SSIM.

7.2 RESULTS

First, we selected three images from the USC-SIPI image bank (UNIVERSITY OF SOUTHERN CALIFORNIA,) and performed the procedure describe above. Figure 17 displays the selected images.



(a) Lena (b) Baboon (c) Airplane

Figura 17 – Sample images.

The MSE, PSNR and SSIM of the reconstructed images obtained from using each approximation presented in this work are shown in Table 18.

For the experiments, we considered $r = 10$. In Table 18, the 5 best results for each measure and sample image are highlighted. All the approximations among the 5 best for all three images besides the DCT ($\widehat{\mathrm{C}}_{\mathrm{I},4}$, $\widehat{\mathrm{C}}_{\mathrm{I},6}$, $\widehat{\mathrm{C}}_{\mathrm{II},7}$, $\widehat{\mathrm{C}}_{\mathrm{II},9}$) were introduced in this work.

Image	Lena			Baboon			Airplane		
Transform	MSE	PSNR	SSIM	MSE	PSNR	SSIM	MSE	PSNR	SSIM
DCT	**40.2377**	**32.0845**	**0.9763**	**338.3289**	**22.8374**	0.9064	**57.3786**	**30.5433**	**0.9832**
$\widehat{\mathrm{C}}_{\mathrm{WHT}}$	62.0565	30.2029	0.9737	377.2737	22.3642	**0.9138**	103.5323	27.9800	0.9792
$\widehat{\mathrm{C}}_{\mathrm{SDCT}}$	110.2814	27.7058	0.9225	455.8198	21.5429	0.8304	158.1937	26.1389	0.9363
$\widehat{\mathrm{C}}_{\mathrm{LO}}$	52.2906	30.9466	0.9722	350.1966	22.6877	0.9008	78.0402	29.2076	0.9789
$\widehat{\mathrm{C}}_{\mathrm{RDCT}}$	58.6558	30.4477	0.9658	363.8937	22.5211	0.8723	85.8379	28.7940	0.9727
$\widehat{\mathrm{C}}_{\mathrm{MRDCT}}$	129.0950	27.0217	0.8935	449.8371	21.6003	0.7437	179.5797	25.5882	0.9018
$\widehat{\mathrm{C}}_{\mathrm{BAS-2008a}}$	59.8314	30.3615	0.9633	376.4999	22.3732	0.8845	93.3886	28.4279	0.9710
$\widehat{\mathrm{C}}_{\mathrm{BAS-2008b}}$	53.1955	30.8721	0.9725	362.1343	22.5421	0.8931	84.4295	28.8659	0.9787
$\widehat{\mathrm{C}}_{\mathrm{BAS-2009}}$	66.2534	29.9187	0.9638	389.3617	22.2273	0.8876	107.8282	27.8035	0.9714
$\widehat{\mathrm{C}}_{\mathrm{BAS-2010}}$	49.9871	31.1422	0.9728	358.4581	22.5864	0.9088	78.4957	29.1823	0.9787
$\widehat{\mathrm{C}}_{\mathrm{BAS-2011}}$	65.7003	29.9551	0.9567	389.5474	22.2252	0.8561	100.7354	28.0990	0.9647
$\widehat{\mathrm{C}}_{\mathrm{BAS-2013}}$	62.0565	30.2029	0.9737	377.2737	22.3642	**0.9138**	103.5323	27.9800	0.9792
$\widehat{\mathrm{C}}_{\mathrm{CBT-1}}$	93.8702	28.4055	0.9418	437.7631	21.7184	0.8257	153.5626	26.2680	0.9424
$\widehat{\mathrm{C}}_{\mathrm{CBT-2}}$	61.1506	30.2668	0.9641	372.2386	22.4226	0.8676	86.9480	28.7382	0.9711
$\widehat{\mathrm{C}}_{\mathrm{CBT-3}}$	59.1153	30.4138	0.9727	363.6265	22.5242	0.9038	92.9502	28.4483	0.9792
$\widehat{\mathrm{C}}_{\mathrm{CBT-4}}$	61.7111	30.2272	0.9711	372.0019	22.4254	0.8996	94.0452	28.3974	0.9778
$\widehat{\mathrm{C}}_{\mathrm{CBT-5}}$	55.0551	30.7228	0.9707	358.9443	22.5805	0.8966	79.3029	29.1379	0.9777
$\widehat{\mathrm{C}}_{\mathrm{CBT-6}}$	113.2215	27.5915	0.9113	433.0143	21.7658	0.7985	166.6957	25.9116	0.9178
$\widehat{\mathrm{C}}_{\mathrm{CBT-7}}$	50.1714	31.1262	0.9744	349.7947	22.6927	0.9023	71.2007	29.6060	0.9810
$\widehat{\mathrm{C}}_{\mathrm{I},2}$	51.1731	31.0404	0.9647	350.2942	22.6865	0.8950	70.3013	29.6612	0.9756
$\widehat{\mathrm{C}}_{\mathrm{I},3}$	50.5512	31.0935	0.9736	350.4899	22.6840	0.8967	69.7947	29.6926	0.9811
$\widehat{\mathrm{C}}_{\mathrm{I},4}$	**42.7374**	**31.8227**	0.9754	**340.5167**	**22.8094**	0.9063	**60.9795**	**30.2790**	0.9825
$\widehat{\mathrm{C}}_{\mathrm{I},5}$	51.1138	31.0454	0.9647	350.6485	22.6821	0.8942	**69.5870**	**29.7055**	0.9756
$\widehat{\mathrm{C}}_{\mathrm{I},6}$	**40.9147**	**32.0120**	**0.9760**	**338.3480**	**22.8372**	0.9071	**58.9464**	**30.4262**	**0.9829**
$\widehat{\mathrm{C}}_{\mathrm{II},3}$	46.8882	31.4202	0.9745	349.1215	22.7010	0.9097	72.4104	29.5328	0.9817
$\widehat{\mathrm{C}}_{\mathrm{II},4}$	68.2633	29.7889	0.9601	370.5620	22.4422	0.8767	97.3407	28.2479	0.9663
$\widehat{\mathrm{C}}_{\mathrm{II},5}$	52.3678	30.9402	**0.9765**	360.4583	22.5623	**0.9118**	82.1987	28.9822	**0.9834**
$\widehat{\mathrm{C}}_{\mathrm{II},6}$	106.0005	27.8777	0.9166	412.5139	21.9764	0.8030	146.8858	26.4610	0.9243
$\widehat{\mathrm{C}}_{\mathrm{II},7}$	**45.0054**	**31.5982**	**0.9774**	**348.9218**	**22.703**5	**0.9165**	71.7508	29.5725	**0.9840**
$\widehat{\mathrm{C}}_{\mathrm{II},8}$	85.7326	28.7993	0.9421	388.7694	22.2339	0.8477	120.5415	27.3194	0.9491
$\widehat{\mathrm{C}}_{\mathrm{II},9}$	**43.1098**	**31.7850**	**0.9772**	**343.9778**	**22.7655**	**0.9141**	**67.6742**	**29.8266**	**0.9839**
$\widehat{\mathrm{C}}_{\mathrm{II},10}$	69.7013	29.6984	0.9584	370.1255	22.4473	0.8748	99.3286	28.1601	0.9652

Tabela 18 – MSE, PSNR and SSIM of each sample image compressed and reconstructed considering the approximations 8-point DCT and $r = 10$

Among the approximations in literature, only $\widehat{\mathrm{C}}_{\mathrm{WHT}}$ and $\widehat{\mathrm{C}}_{\mathrm{BAS-2013}}$ showed up in the top 5, although it only happened for the Baboon image and SSIM measure.

In order to have a more general idea about the behavior of those transforms, we carried out the experiments described before for all the 44 images in the dataset, considering all the va-

lues of $r = 1, 2, \ldots, 64$. The MSE, PSNR and SSIM were calculated in each case. The average curves obtained are displayed in the plots in Figure 18. The curves for all the approximations were calculated, but only the ones with better results, i.e., the ones with results closer to the DCT results were kept in the plots in order to provide a clear visualization.

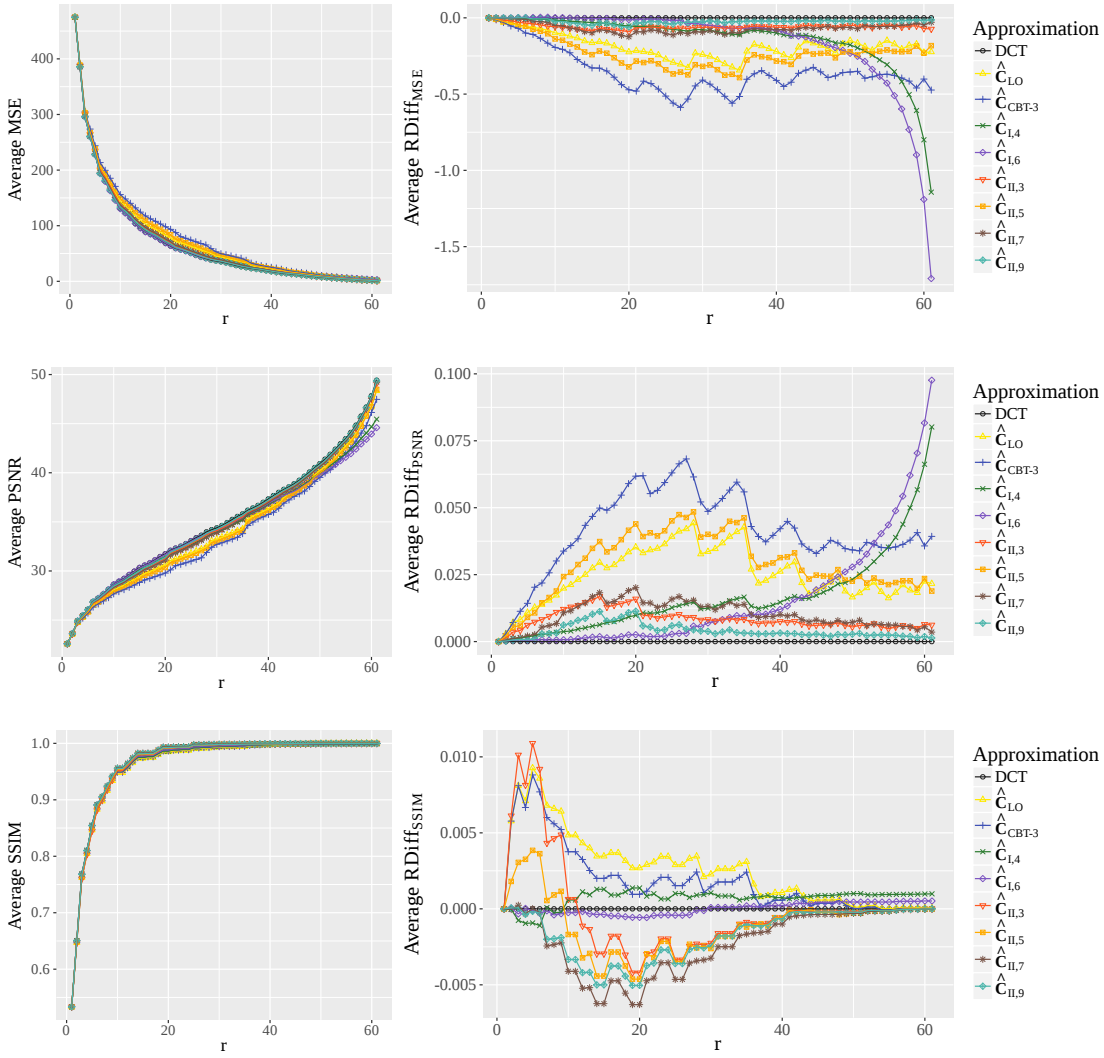


Figura 18 – Average curves for the MSE, PSNR and SSIM.

From the plots in Figure 18, we can observe that for the MSE and PSNR: (i) although the approximations in literature considered present results very close to the DCT, none of them overcomes the DCT; (ii) the proposed approximations $\widehat{\mathrm{C}}_{\mathrm{I},4}$ and $\widehat{\mathrm{C}}_{\mathrm{I},6}$, that presented the best results in terms of the metrics considered in Chapter 6, showed the closest results to the DCT for small values of r. However, as r grows, their curves tend to distance themselves from the DCT; (iii) Approximations $\widehat{\mathrm{C}}_{\mathrm{II},3}$, $\widehat{\mathrm{C}}_{\mathrm{II},7}$ and $\widehat{\mathrm{C}}_{\mathrm{II},9}$ are consistently closer to the DCT than the approximations in literature considered in the plots, $\widehat{\mathrm{C}}_{\mathrm{LO}}$ and $\widehat{\mathrm{C}}_{\mathrm{CBT-3}}$, and, for larger values of

r, they show the closest results to the DCT.

On other hand, for the SSIM, several approximations presented results better than the DCT. In particular, all the approximations proposed in the work show in the plots of Figure 18 have results better than the DCT for, at least, 7 values of r ($\widehat{\mathrm{C}}_{\mathrm{I},4}$), and, at most, 59 values of r ($\widehat{\mathrm{C}}_{\mathrm{II},7}$ and $\widehat{\mathrm{C}}_{\mathrm{II},9}$). However, $\widehat{\mathrm{C}}_{\mathrm{LO}}$ and $\widehat{\mathrm{C}}_{\mathrm{CBT–3}}$, only showed results better than the DCT for one and four values of r, respectively.

8 FAST ALGORITHM, VIDEO CODING, AND HARDWARE REALIZATION

Comparing the computational cost of its direct implementation, performance measures and results in the image compression experiments, we selected one of the new approximations to further analyze. The chosen approximation was $\mathbf{T}_{\text{II},3}$. It is an orthogonal matrix, and it is among the less complex approximations proposed. Also, it overcomes all the approximations in literature in the performance measures and overcomes the DCT in terms of SSIM in the image compression experiments for several values of r.

Subsection 8.1.1 was written in colaboration with Thiago L. T. da Silveira (Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande do Sul), who is part of the paper *Low-complexity 8-point DCT Approximation Based on Angle Similarity for Image and Video Coding* accepted for publication on the Journal of Multidimensional Systems and Signal Processing.

8.1 FAST ALGORITHM

The direct implementation of $\mathbf{T}_{\text{II},3}$ requires 48 additions and 24 bit-shifting operations. However, such computational cost can be significantly reduced by means of sparse matrix factorization. Considering butterfly-based structures we could derive the following factorization for $\mathbf{T}_{\text{II},3}$:

$$\mathbf{T}_{\text{II},3} = \mathbf{D} \cdot \mathbf{A}_4 \cdot \mathbf{A}_3 \cdot \mathbf{A}_2 \cdot \mathbf{A}_1,$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1 & & & & & & & 1 \\ & 1 & & & & & 1 & \\ & & 1 & & & 1 & & \\ & & & 1 & 1 & & & \\ & & & 1 & -1 & & & \\ & & 1 & & & -1 & & \\ & 1 & & & & & -1 & \\ 1 & & & & & & & -1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 1 & & & 1 & & & & \\ & 1 & 1 & & & & & \\ & 1 & -1 & & & & & \\ 1 & & & -1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix}, \quad \mathbf{A}_4 = \begin{bmatrix} 1 & & & & & & & \\ & & & & \frac{1}{2} & 1 & 1 & \\ & 1 & 2 & & & & & \\ & & & -1 & -1 & & \frac{1}{2} & \\ 1 & & & & & & & \\ & & & \frac{1}{2} & & -1 & 1 & \\ & -2 & 1 & & & & & \\ & & & -1 & 1 & -\frac{1}{2} & & \end{bmatrix},$$

and $\mathbf{D} = \mathrm{diag}(1, 2, 1, 2, 1, 2, 1, 2)$. Figure 19 shows the signal flow graph (SFG) related to the above factorization. The computational cost of this algorithm is only 24 additions and six multiplications by two. The multiplications by two are extremely simple to be performed, requiring only bit-shifting operations (BRITANAK; YIP; RAO, 2007). The fast algorithm proposed requires 50% less additions and 75% less bit-shifting operations when compared to the direct implementation. The computational costs of the considered methods are shown in Table 19, the exact DCT and IDCT (OHM et al., 2012) were included as reference.

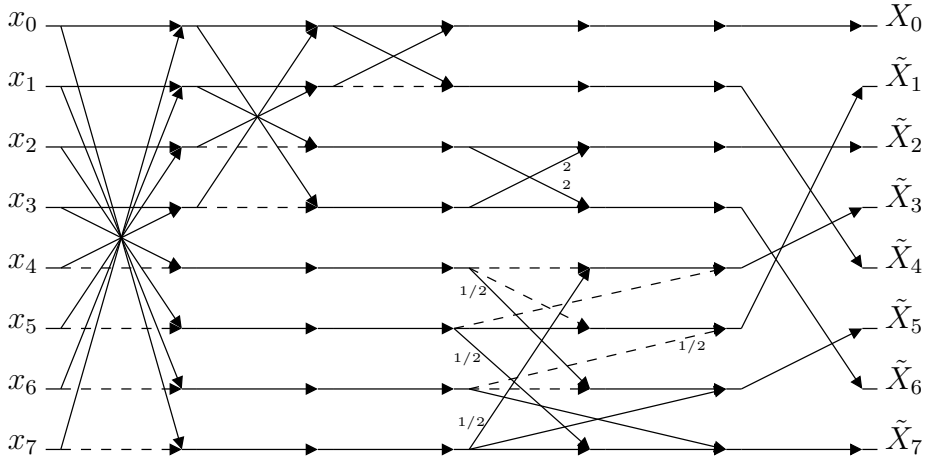


Figura 19 – SFG of the proposed transform, relating the input data x_n, $n = 0, 1, \ldots, 7$, to its correspondent coefficients $\tilde{X}_k$, $k = 0, 1, \ldots, 7$, where $\tilde{\mathbf{X}} = \mathbf{T}_{\mathrm{II},3} \cdot \mathbf{x}$. Dashed arrows represent multiplication by -1.

In general terms, DCT approximations exhibit a trade-off between computational cost and transform performance (TABLADA; BAYER; CINTRA, 2015), i.e., less complex matrices effect poor spectral approximations (BRITANAK; YIP; RAO, 2007). Departing from this general behavior, the proposed transformation $\mathbf{T}_{\mathrm{II},3}$ has (i) excelling performance measures and (ii) lower or similar arithmetic cost when compared to competing methods, as shown in Table 19.

8.1.1 Video coding

In order to assess the proposed transform $\widehat{\mathbf{C}}_{\mathrm{II},3}$ as a tool for video coding, we embedded it into a public available HEVC reference software (Joint Collaborative Team on Video Coding (JCT-VC), 2013). The HEVC presents several improvements relative to its predecessors (SULLIVAN et al., 2012) and aims at providing high compression rates (POURAZAD et al., 2012).

Method	Multiplications	Additions	Bit-shifts
DCT (LOEFFLER; LIGTENBERG; MOSCHYTZ, 1989)	11	29	0
IDCT (HEVC) (OHM et al., 2012)	0	50	30
$\mathbf{T}_{\text{II},3}$ (proposed)	0	24	6
$\widehat{\mathbf{C}}_{\text{LO}}$ (LENGWEHASATIT; ORTEGA, 2004)	0	24	2
$\mathbf{T}_{\text{SDCT}}$ (HAWEEL, 2001)	0	24	0
$\mathbf{T}_{\text{RDCT}}$ (BAYER; CINTRA, 2010)	0	22	0
$\mathbf{T}_{\text{MRDCT}}$ (BAYER; CINTRA, 2012)	0	14	0
$\mathbf{T}_{\text{BAS-2008a}}$ (BOUGUEZEL; AHMAD; SWAMY, 2008a)	0	18	2
$\mathbf{T}_{\text{BAS-2008b}}$ (BOUGUEZEL; AHMAD; SWAMY, 2008b)	0	21	0
$\mathbf{T}_{\text{BAS-2009}}$ (BOUGUEZEL; AHMAD; SWAMY, 2009)	0	18	0
$\mathbf{T}_{\text{BAS-2011}}$ (BOUGUEZEL; AHMAD; SWAMY, 2011)	0	16	0
$\mathbf{T}_{\text{BAS-2013}}$ (BOUGUEZEL; AHMAD; SWAMY, 2013)	0	24	0
$\mathbf{T}_{\text{CBT–1}}$ (CINTRA; BAYER; TABLADA, 2014)	0	22	4
$\mathbf{T}_{\text{CBT–2}}$ (CINTRA; BAYER; TABLADA, 2014)	0	22	6
$\mathbf{T}_{\text{CBT–3}}$ (CINTRA; BAYER; TABLADA, 2014)	0	24	0
$\mathbf{T}_{\text{CBT–4}}$ (CINTRA; BAYER; TABLADA, 2014)	0	24	4
$\mathbf{T}_{\text{CBT–5}}$ (CINTRA; BAYER; TABLADA, 2014)	0	24	6
$\mathbf{T}_{\text{CBT–6}}$ (CINTRA; BAYER; TABLADA, 2014)	0	18	0
$\mathbf{T}_{\text{CBT–7}}$ (CINTRA; BAYER; TABLADA, 2014)	0	28	12

Tabela 19 – Computational cost comparison

Differently from other standards, HEVC employs not only an 8-point IDCT but also transforms of size 4, 16, and 32 (OHM et al., 2012). Such feature effects a series of optimization routines allowing the processing of big smooth or textureless areas (POURAZAD et al., 2012).

For this reason, aiming to derive large blocklength transforms for HEVC embedding, we submitted the proposed transform matrix $\mathbf{T}_{\text{II},3}$ to the Jridi–Alfalou–Meher (JAM) scalable algorithm (JRIDI; ALFALOU; MEHER, 2015). Such method resulted in 16- and 32-point versions of the proposed matrix $\mathbf{T}_{\text{II},3}$ that are suitable for the sought video experiments. Although the JAM algorithm is similar to Chen's DCT (CHEN; SMITH; FRALICK, 1977), it exploits redundancies allowing concise and high parallelizable hardware implementations (JRIDI; ALFALOU; MEHER, 2015). From a low-complexity $N/2$-point transform, the JAM algorithm generates an $N \times N$ matrix transformation by combining two instantiations of the smaller one. The larger N-point transform is recursively defined by

$$\mathbf{T}_{(N)} = \frac{1}{\sqrt{2}} \mathrm{M}_N^{\text{per}} \begin{bmatrix} \mathbf{T}_{\left(\frac{N}{2}\right)} & \mathbf{Z}_{\frac{N}{2}} \\ \mathbf{Z}_{\frac{N}{2}} & \mathbf{T}_{\left(\frac{N}{2}\right)} \end{bmatrix} \mathrm{M}_N^{\text{add}}, \tag{8.1}$$

where $\mathbf{Z}_{\frac{N}{2}}$ is a matrix of order $N/2$ with all zeroed entries. Matrices $\mathrm{M}_N^{\text{add}}$ and $\mathrm{M}_N^{\text{per}}$ are, respec-

tively, obtained according to

$$\mathbf{M}_N^{\text{add}} = \begin{bmatrix} \mathbf{I}_{\frac{N}{2}} & \bar{\mathbf{I}}_{\frac{N}{2}} \\ \bar{\mathbf{I}}_{\frac{N}{2}} & -\mathbf{I}_{\frac{N}{2}} \end{bmatrix}$$

and

$$\mathbf{M}_N^{\text{per}} = \begin{bmatrix} \mathbf{P}_{N-1,\frac{N}{2}} & \mathbf{Z}_{1,\frac{N}{2}} \\ \mathbf{Z}_{1,\frac{N}{2}} & \mathbf{P}_{N-1,\frac{N}{2}} \end{bmatrix},$$

where $\mathbf{I}_{\frac{N}{2}}$ and $\bar{\mathbf{I}}_{\frac{N}{2}}$ are, respectively, the identity and counter-identity matrices of order $N/2$ and $\mathbf{P}_{N-1,\frac{N}{2}}$ is an $(N-1) \times (N/2)$ matrix whose ith row vectors are defined by

$$\mathbf{P}_{N-1,\frac{N}{2}}^{(i)} = \begin{cases} \mathbf{Z}_{1,\frac{N}{2}}, & \text{if } i = 1, 3, 5, \ldots, N-1 \\ \mathbf{I}_{\frac{N}{2}}^{(i/2)}, & \text{if } i = 0, 2, 4, \ldots, N-2. \end{cases}$$

The scaling factor $1/\sqrt{2}$ of (8.1) can be merged into the image/video compression quantization step. Furthermore, Equation 3.3 can be applied to generate orthogonal versions of larger transforms. The computational cost of the resulting N-point transform is given by twice the number of bit-shifting operations of the original $N/2$-point transform; and twice the number of additions plus N extra additions. Following the described algorithm, we obtained the 16- and 32-point low-complexity transform matrices proposed.

Figures 20 and 21 display the SFG for the low-complexity transform matrices $\mathbf{T}_{\text{II,3–(16)}}$ and $\mathbf{T}_{\text{II,3–(32)}}$ derived from $\mathbf{T}_{\text{II,3}}$. Table 20 lists the computational costs of the proposed transform for sizes $N = 8, 16, 32$ compared to an efficient implementation of the IDCT (MEHER et al., 2014).

N	IDCT (MEHER et al., 2014)		$\mathbf{T}_{\text{II,3}}$		Reduction from IDCT to $\mathbf{T}_{\text{II,3}}$	
	Additions	Bit-shifts	Additions	Bit-shifts	Additions	Bit-shifts
8	50	30	24	6	52%	80%
16	186	86	64	12	65.6%	86%
32	682	278	160	24	76.5%	91.3%

Tabela 20 – Computational cost comparison for 8-, 16-, and 32-point transforms embedded in HEVC reference software.

In our experiments, the original 8-, 16-, and 32-point integer transforms of HEVC were substituted by $\widehat{\mathbf{C}}_{\text{II,3}}$ and its scaled versions. The original 4-point transform was kept unchanged

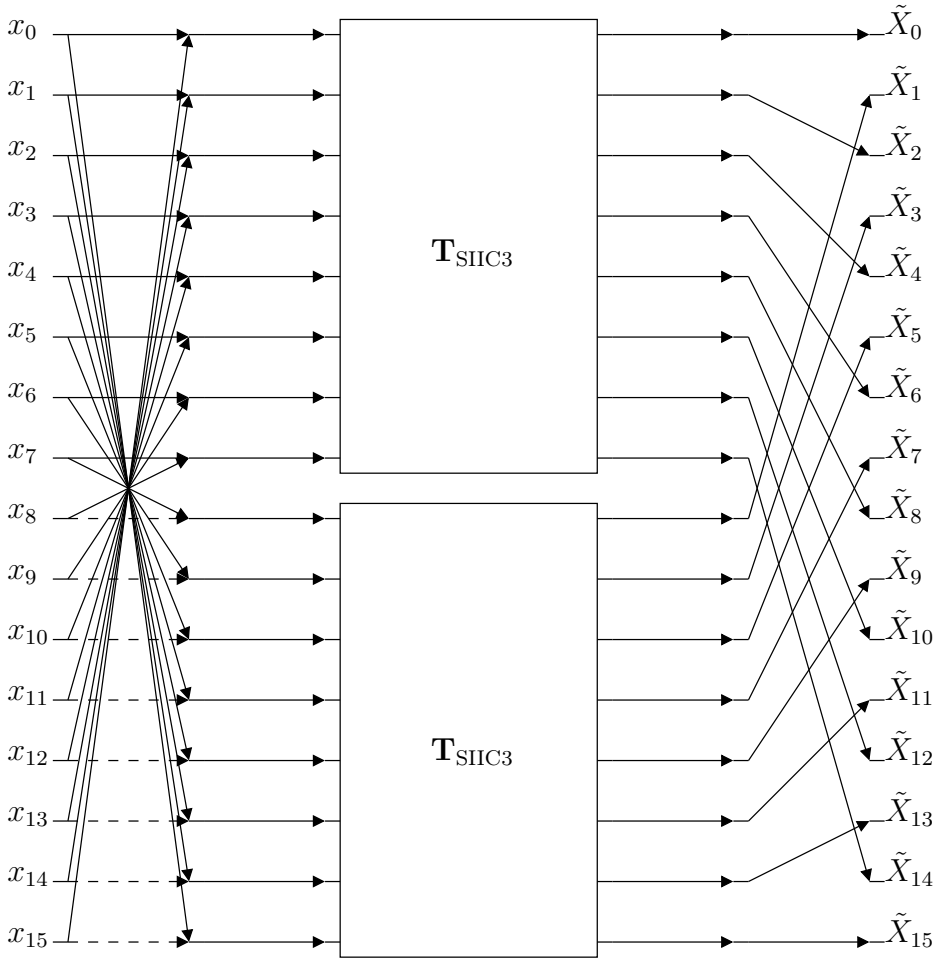


Figura 20 – SFG for the proposed 16-point low-complexity transform matrix, $\mathbf{T}_{\text{II,3–(16)}}$.

because it is already a very low-complexity transformation. We encoded the first 100 frames of one video sequence of each A to F class in accordance with the common test conditions (CTC) documentation (BOSSEN, 2013). Namely we used the 8-bit videos: `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). As suggested in (JRIDI; ALFALOU; MEHER, 2015), all the test parameters were set according to the CTC documentation. We tested the proposed transforms in All Intra (AI), `Random Access` (RA), `Low Delay B` (LD-B), and `Low Delay P` (LD-P) configurations, all in the `Main` profile.

We selected the frame-by-frame MSE and PSNR (OHM et al., 2012) for each YUV color channel as figures of merit. Then, for all test videos, we computed the rate distortion (RD) curve considering the recommended quantization parameter (QP) values, i.e. 22, 27, 32,

Video sequence	AI		RA		LD-B		LD-P	
	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate	BD-PSNR	BD-Rate
PeopleOnStreet	0.2999	−5.5375	0.1467	−3.4323	N/A	N/A	N/A	N/A
BasketballDrive	0.1692	−6.1033	0.1412	−6.1876	0.1272	−5.2730	0.1276	−5.2407
RaceHorses	0.4714	−5.8250	0.5521	−8.6149	0.5460	−7.9067	0.5344	−7.6868
BlowingBubbles	0.0839	−1.4715	0.0821	−2.1612	0.0806	−2.1619	0.0813	−2.2370
KristenAndSara	0.2582	−5.0441	N/A	N/A	0.1230	−4.1823	0.1118	−4.0048
BasketballDrillText	0.1036	−1.9721	0.1372	−3.2741	0.1748	−4.3383	0.1646	−4.1509

Tabela 21 – BD-PSNR (dB) and BD-Rate (%) of the modified HEVC reference software for tested video sequences.

and 37 (BOSSEN, 2013). The resulting RD curves are depicted in Figure 22. We have also measured the Bjøntegaard's delta PSNR (BD-PSNR) and delta rate (BD-Rate) (BJØNTEGAARD, 2001; HANHART; EBRAHIMI, 2014) for the modified HEVC software. These values are summarized in Table 21. We demonstrate that replacing the IDCT by the proposed transform and its scaled versions results in a loss in quality of at most 0.47dB for the AI configuration, which corresponds to an increase of 5.82% in bitrate. The worst performance for the other configurations—RA, LD-B, and LD-P—are found for the KristenAndSara video sequence, where approximately 0.55dB are lost if compared to the original HEVC implementation.

Despite the very low computational cost when compared to the IDCT (cf. Table 20), the proposed transform does not introduce significant errors. Figure 23 illustrates the tenth frame of the BasketballDrive video encoded according to the default HEVC IDCT and $\widehat{\mathrm{C}}_{\mathrm{II},3}$ and its scaled versions for each coding configuration. The QP was set to 32. Visual degradations are virtually nonperceptible demonstrating real-world applicability of the proposed DCT approximations for high resolution video coding.

8.2 FPGA IMPLEMENTATION

The proposed design along with $\mathbf{T}_{\mathrm{LO}}$ and $\mathbf{T}_{\mathrm{CBT\text{–}3}}$ were implemented on a FPGA chip using the Xilinx ML605 board. Considering hardware co-simulation the FPGA realization was tested with 100,000 random 8-point input test vectors. The test vectors were generated from within the MATLAB environment and, using JTAG based hardware co-simulation, routed to

the physical FPGA device where each algorithm was realized in the reconfigurable logic fabric. Then the computational results obtained from the FPGA algorithm implementations were routed back to the MATLAB memory space. The diagrams for the designs can be seen in Figure 24.

The metrics employed to evaluate the FPGA implementations were: configurable logic blocks (CLB), flip-flop (FF) count, and critical path delay (T_{cpd}), in ns. The maximum operating frequency was determined by the critical path delay as $F_{\text{max}} = (T_{\text{cpd}})^{-1}$, in MHz. Values were obtained from the Xilinx FPGA synthesis and place-route tools by accessing the `xflow.results` report file. Using the Xilinx XPower Analyzer, we estimated the static (Q_p in W) and dynamic power (D_p in $\mathrm{mW/MHz}$) consumption. In addition, we calculated area-time (AT) and area-time-square (AT^2) figures of merit, where area is measured as the CLBs and time as the critical path delay. The values of those metrics for each design are shown in Table 22.

Approximation	CLB	FF	T_{cpd} (ns)	F_{max} (MHz)	D_p (mW/GHz)	Q_p (W)	AT	AT^2
$\mathbf{T}_{\text{II,3}}$ (proposed)	135	408	1.750	571	2.74	3.471	236	413
$\mathbf{T}_{\text{LO}}$	114	349	1.900	526	2.82	3.468	217	412
$\mathbf{T}_{\text{CBT–3}}$	125	389	2.100	476	2.57	3.460	262	551

Tabela 22 – Hardware resource consumption and power consumption using Xilinx Virtex-6 XC6VLX240T 1FFG1156 device.

The design linked to the proposed design approximation $\mathbf{T}_{\text{II,3}}$ possesses the smallest T_{cpd} among the considered methods. Such critical path delay allows for operations at a 8.55% and 19.96% higher frequency than the designs associated to $\mathbf{T}_{\text{LO}}$ and $\mathbf{T}_{\text{CBT–3}}$, respectively. In terms of area-time and are-time-square measures, the design linked to the approximation $\mathbf{T}_{\text{LO}}$ presents the best results, followed closely by the one associated to $\mathbf{T}_{\text{II,3}}$.

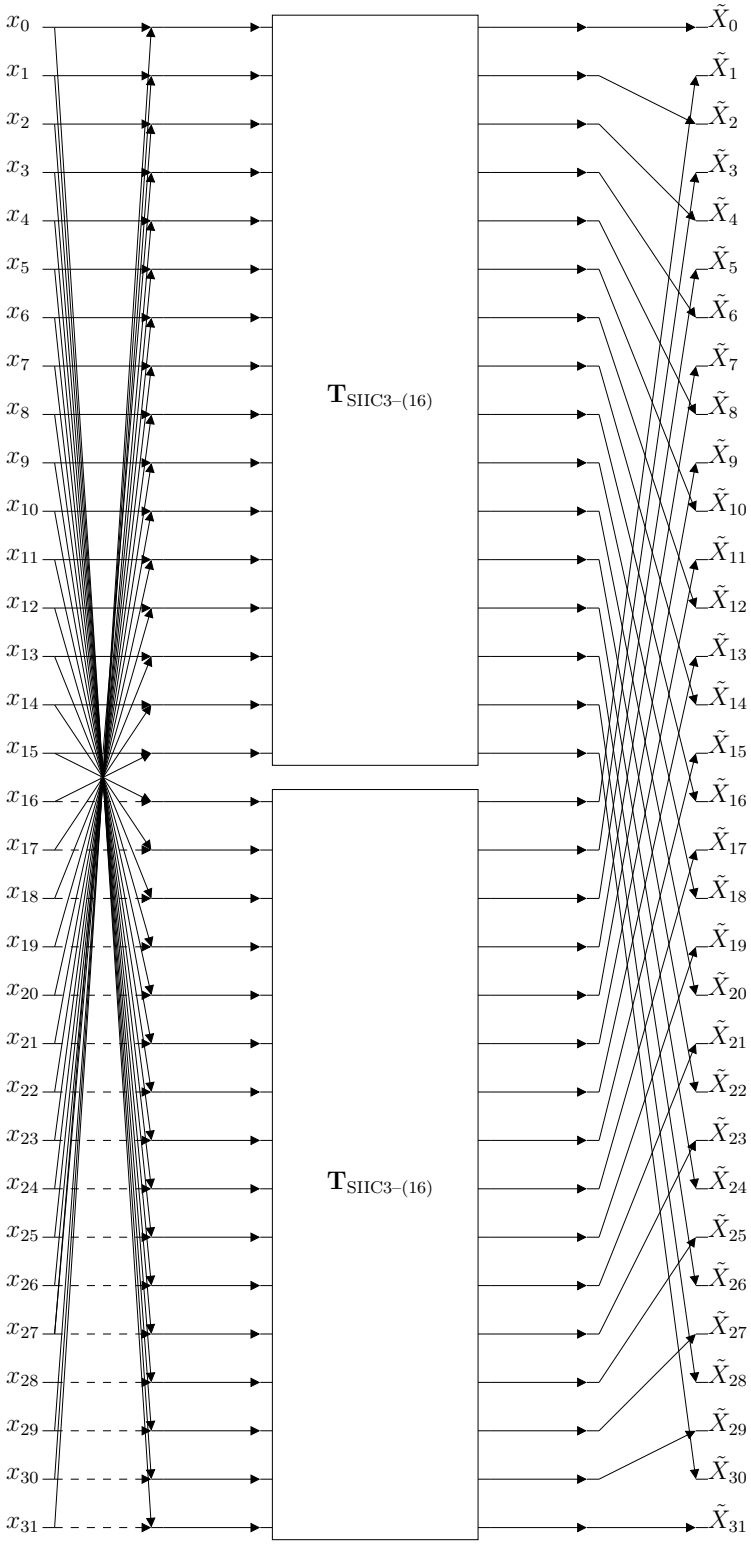


Figura 21 – SFG for the proposed 32-point low-complexity transform matrix, $\mathbf{T}_{\text{II,3–(32)}}$, where $\mathbf{T}_{\text{II,3–(16)}}$ is the 16-point matrix presented in Figure 20.

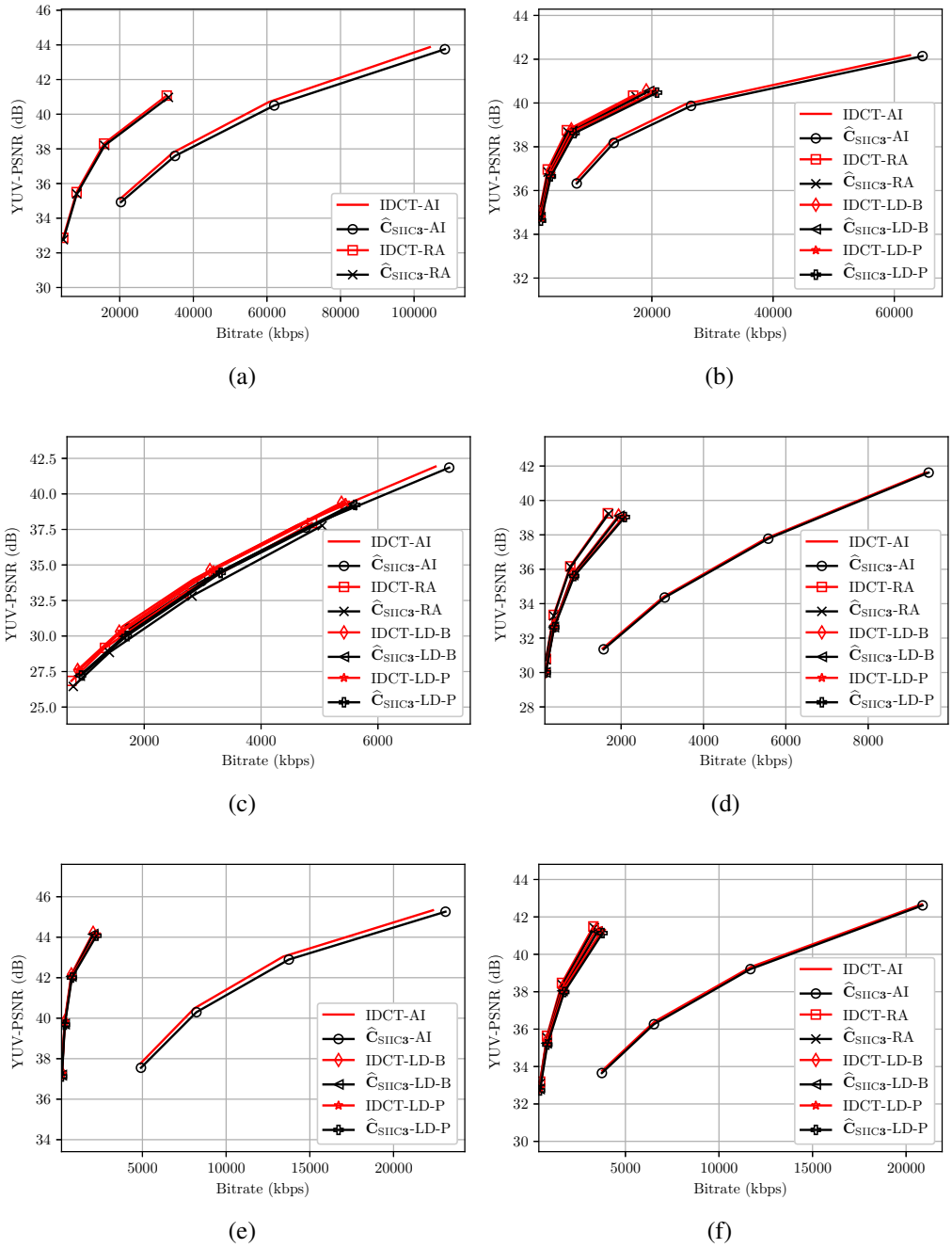


Figura 22 – Rate distortion curves of the modified HEVC software for test sequences: (a) `PeopleOnStreet`, (b) `BasketballDrive`, (c) `RaceHorses`, (d) `BlowingBubbles`, (e) `KristenAndSara`, and (f) `BasketbalDrillText`.

(a) MSE-Y = 10.4097, MSE-U = 3.5872, MSE-V = 3.3079, PSNR-Y = 37.9564, PSNR-U = 42.5832, PSNR-V = 42.9353

(b) MSE-Y = 10.8159, MSE-U = 3.8290, MSE-V = 3.5766, PSNR-Y = 37.7902, PSNR-U = 42.2999, PSNR-V = 42.5961

(c) MSE-Y = 10.1479, MSE-U = 3.4765, MSE-V = 3.1724, PSNR-Y = 38.0670, PSNR-U = 42.7194, PSNR-V = 43.1170

(d) MSE-Y = 10.3570, MSE-U = 3.6228, MSE-V = 3.3113, PSNR-Y = 37.9785, PSNR-U = 42.5403, PSNR-V = 42.9308

(e) MSE-Y = 14.0693, MSE-U = 4.0741, MSE-V = 4.4404, PSNR-Y = 36.6481, PSNR-U = 42.0304, PSNR-V = 41.6566

(f) MSE-Y = 14.5953, MSE-U = 4.1377, MSE-V = 4.6053, PSNR-Y = 36.4887, PSNR-U = 41.9632, PSNR-V = 41.4982

(g) MSE-Y = 14.6155, MSE-U = 4.1349, MSE-V = 4.5502, PSNR-Y = 36.4827, PSNR-U = 41.9661, PSNR-V = 41.5505

(h) MSE-Y = 15.0761, MSE-U = 4.2812, MSE-V = 4.6444, PSNR-Y = 36.3479, PSNR-U = 41.8151, PSNR-V = 41.4615

Figura 23 – Compression of the tenth frame of `BasketballDrive` using (a),(c),(e) the default and (b),(d),(f) the modified versions of the HEVC software for QP = 32, and AI, RA, LD-B, and LD-P coding configurations, respectively.

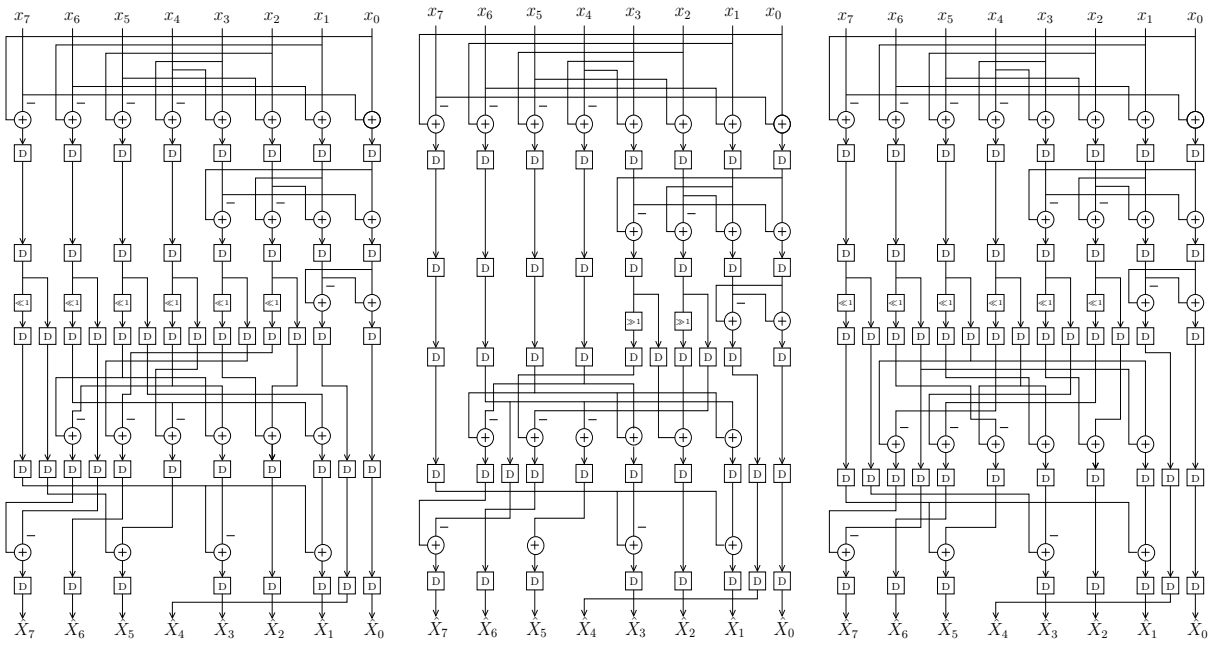


Figura 24 – Architectures for (a) $\mathbf{T}_{\text{II},3}$, (b) $\mathbf{T}_{\text{LO}}$, and (c) $\mathbf{T}_{\text{CBT–3}}$.

9 CONCLUSIONS

9.1 OVERVIEW

In this work, we introduced a greedy algorithm to find low-complexity approximations for a given matrix based on angular distance. The initial version of the method had no constraints. Thus, in order to guarantee the orthogonality of the obtained approximations, we proposed a constrained version of the proposed algorithm. Then, we discussed several ways to reduce the algorithm complexity by exploring the DFT, DHT and DCT structural features and defined approximation schemes.

The defined approximation schemes were used to derive new approximations for the 8-point DCT. Thirteen new approximations were obtained. All of them had outstanding results in terms of performance measures and six of them had also great results in the image compression experiments, *overcoming the exact DCT in several compression levels for the SSIM measure.* This is a relevant result because it directly offers counter-examples to the belief that the coding performance of an approximation is supposed to always be inferior to the exact DCT. We show that this is not always the case.

Although the problem of matrix approximation is quite simple to state, it is also very tricky and offers several non-linearities. Notice that finding low-complexity matrices is an integer optimization problem. Thus, navigating in the low-complexity matrix search space might generate non-trivial performance curves, usually leading to discontinuities. Indeed, it is very hard to tell beforehand whether an approximation method will deliver extremely good results.

However, the rationale of how to navigate in the low-complexity matrix search space matters. In the present work, we did as much as possible to furnish a sound theoretical analysis capable of capturing good approximations.

Based on the results from the evaluation steps, we took $\mathbf{T}_{\text{II},3}$ and proposed a fast algorithm for it that requires only 24 additions and 6 bit-shifthings operations. The FPGA implementation of $\mathbf{T}_{\text{II},3}$ was also made and compared to $\mathbf{T}_{\text{LO}}$ and $\mathbf{T}_{\text{CBT–3}}$. In this case, $\mathbf{T}_{\text{II},3}$ also

overcame the approximations in literature, being able to work at a 8.55% and 19.96% higher frequency than the designs associated to $\mathbf{T}_{\text{LO}}$ and $\mathbf{T}_{\text{CBT–3}}$, respectively.

For the video experiments, we scaled $\mathbf{T}_{\text{II,3}}$ using the JAM method (JRIDI; ALFALOU; MEHER, 2015), to obtain 16– and 32-point approximations. The obtained approximations were embedded into the HEVC reference software. The replacement of the IDCT by $\mathbf{T}_{\text{CBT–3}}$ and its scaled versions resulted in a quality loss of, at most, 0.55dB, observed for the RA configuration in the RaceHorses video sequence.

9.2 PUBLISHED PAPERS

Based on this research, the following work was accepted for publication:

- Oliveira, R. S., Cintra, R. J., Bayer, F. M., Silveira, T. L. T., Madanayake, A., and Leite, A. Low-complexity 8-point DCT Approximation Based on Angle Similarity for Image and Video Coding. *Multidimensional Systems and Signal Processing*, 2018.

The author of this Dissertation also colaborated in the following papers working on the hardware realization sections:

- Coutinho, Vítor A.; Cintra, Renato J.; Bayer, Fábio M.; Oliveira, Paulo A. M.; Oliveira, Raíza S.; Madanayake, Arjuna. Pruned Discrete Tchebichef Transform Approximation for Image Compression. *Circuits Systems and Signal Processing*, vol. 37, pp. 1-21, 2018.
- Oliveira, Paulo A. M.; Oliveira, Raíza S.; Cintra, Renato J.; Bayer, Fábio M.; Madanayake, Arjuna. JPEG quantisation requires bit-shifts only. *Electronics Letters*, vol. 53, pp. 588-590, 2017.
- Silveira, Thiago L. T.; Oliveira, Raíza S.; Bayer, Fábio M.; Cintra, Renato J.; Madanayake, Arjuna. Multiplierless 16-point DCT approximation for low-complexity image and video coding. *Signal, Image and Video Processing*, vol. 11, pp. 1-7, 2016.

9.3 FUTURE WORKS

For future works, we suggest the following lines of research:

- The main restriction of the proposed unconstrained method is the fact that the seach space grows very quick as M increases. In this sense, we suggest the investigation of efficient search over the matrix space, possibly without having to consider all the elements in it;

- For the constrained to orthogonality version of the proposed method, we have a search space that grows exponentially and $M!$ optimization problems to solve. Thus, we suggest the study of not only a better way of navigating the search space but also the investigation of ways to previously select the best approximation orders;

- Since we focused on finding approximations for the 8-point DCT, a natural next step would be to approximate larger DCT matrices, such as the 16–, 32–, and 64-point DCT;

- We proposed in this work schemes to approximate not only the DCT but also the DFT and DHT. Then, for future works, we suggest using the proposed approximation schemes to find low-complexity approximations for the DFT and DHT;

- The convolution operation can be represented as a matrix–vector product. Such representation may pave the way to the design of low-complexity architectures for digital (convolutional) filters.

REFERENCES

ARAI, Y.; AGUI, T.; NAKAJIMA, M. A fast DCT-SQ scheme for images. *Transactions of the IEICE*, E-71, n. 11, p. 1095–1097, nov. 1988. Citado 2 vezes nas páginas 30 e 33.

BÁRTFAI, I. An illustration of harmonic regression based on the results of the fast Fourier transformation. *Yugoslav Journal of Operations Research*, v. 12, n. 2, p. 185–201, 2016. Citado na página 20.

BAYER, F. M.; CINTRA, R. J. Image compression via a fast DCT approximation. *IEEE Latin America Transactions*, v. 8, n. 6, p. 708–713, dez. 2010. ISSN 1548-0992. Citado 5 vezes nas páginas 21, 34, 39, 71 e 79.

BAYER, F. M.; CINTRA, R. J. DCT-like transform for image compression requires 14 additions only. *Electronics Letters*, v. 48, n. 15, p. 919–921, jul. 2012. ISSN 0013-5194. Citado 4 vezes nas páginas 36, 40, 64 e 79.

BJØNTEGAARD, G. Calculation of average PSNR differences between RD-curves. In: *13th VCEG Meeting*. Austin, TX, USA: [s.n.], 2001. Document VCEG-M33. Citado na página 82.

BLAHUT, R. E. *Fast algorithms for signal processing*. 2nd. ed. [S.l.]: Cambridge University Press, 2010. Citado 5 vezes nas páginas 19, 25, 32, 35 e 43.

BOSSEN, F. *Common Test Conditions and Software Reference Configurations*. San Jose, CA, USA: [s.n.], 2013. Document JCT-VC L1100. Citado 2 vezes nas páginas 81 e 82.

BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. Low-complexity 8×8 transform for image compression. *Electronics Letters*, v. 44, n. 21, p. 1249–1250, 2008. ISSN 0013-5194. Citado 2 vezes nas páginas 38 e 79.

BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. Low-complexity 8×8 transform for image compression. *Electronics Letters*, v. 44, n. 21, p. 1249–1250, set. 2008. ISSN 0013-5194. Citado 2 vezes nas páginas 38 e 79.

BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. A fast 8×8 transform for image compression. In: *International Conference on Microelectronics (ICM)*. [S.l.: s.n.], 2009. p. 74–77. Citado 2 vezes nas páginas 38 e 79.

BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. A novel transform for image compression. In: *53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. [S.l.: s.n.], 2010. p. 509–512. ISSN 1548-3746. Citado na página 38.

BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. A low-complexity parametric transform for image compression. In: *Proceedings of the 2011 IEEE International Symposium on Circuits and Systems*. [S.l.: s.n.], 2011. Citado 2 vezes nas páginas 38 e 79.

BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. Binary discrete cosine and Hartley transforms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 60, n. 4, p. 989–1002, abr. 2013. Citado 2 vezes nas páginas 38 e 79.

BRACEWELL, R. N. Discrete Hartley transform. *Journal of the Optical Society of America*, Optical Society of America, v. 73, n. 12, p. 1832–1835, 1983. Citado 2 vezes nas páginas 20 e 26.

BRACEWELL, R. N. *The Fourier transform and its applications*. 3rd. ed. [S.l.]: McGraw-Hill New York, 2000. Citado na página 20.

BRITANAK, V.; YIP, P.; RAO, K. R. *Discrete Cosine and Sine Transforms*. [S.l.]: Academic Press, 2007. Citado 16 vezes nas páginas 19, 20, 27, 28, 29, 30, 34, 35, 36, 43, 44, 63, 64, 65, 72 e 78.

CAO, L. et al. Multi-focus image fusion based on spatial frequency in discrete cosine transform domain. *IEEE signal processing letters*, IEEE, v. 22, n. 2, p. 220–224, 2015. Citado na página 20.

CHAM, W. K. Development of integer cosine transforms by the principle of dyadic symmetry. In: *IEE Proceedings I Communications, Speech and Vision*. [S.l.: s.n.], 1989. v. 136, n. 4, p. 276–282. ISSN 0956-3776. Citado na página 35.

CHEN, M.-Y.; CHEN, B.-T. Online fuzzy time series analysis based on entropy discretization and a fast Fourier transform. *Applied Soft Computing*, Elsevier, v. 14, p. 156–166, 2014. Citado na página 19.

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, set. 1977. ISSN 0090-6778. Citado 2 vezes nas páginas 33 e 79.

CHENG, C.-H. et al. Chirp signal detection using FFT peak frequency difference [correspondence]. *IEEE Transactions on Aerospace and Electronic Systems*, IEEE, v. 52, n. 3, p. 1449–1453, 2016. Citado na página 19.

CINTRA, R. J. An integer approximation method for discrete sinusoidal transforms. *Journal of Circuits, Systems, and Signal Processing*, v. 30, n. 6, p. 1481–1501, December 2011. Citado 2 vezes nas páginas 19 e 36.

CINTRA, R. J.; BAYER, F. M. A DCT approximation for image compression. *IEEE Signal Processing Letters*, v. 18, n. 10, p. 579–582, out. 2011. Citado 4 vezes nas páginas 36, 44, 63 e 71.

CINTRA, R. J.; BAYER, F. M.; TABLADA, C. J. Low-complexity 8-point DCT approximations based on integer functions. *Signal Processing*, 2014. ISSN 0165-1684. Citado 6 vezes nas páginas 21, 29, 34, 40, 71 e 79.

CORMEN, T. et al. Introduction to algorithms. In: ____. [S.l.]: MIT Press, 2001. cap. 16. Citado na página 42.

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. Citado na página 21.

DUHAMEL, P.; VETTERLI, M. Improved Fourier and Hartley transform algorithms: Application to cyclic convolution of real data. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, IEEE, v. 35, n. 6, p. 818–824, 1987. Citado 2 vezes nas páginas 20 e 34.

FAN, Z. et al. Human gait recognition based on discrete cosine transform and linear discriminant analysis. In: IEEE. *IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. [S.l.], 2016. p. 1–6. Citado na página 20.

FEIG, E.; WINOGRAD, S. Fast algorithms for the discrete cosine transform. *IEEE Transactions on Signal Processing*, v. 40, n. 9, p. 2174–2193, set. 1992. Citado 2 vezes nas páginas 33 e 35.

FITZKE, F. et al. Fourier transform analysis of human corneal endothelial specular photomicrographs. *Experimental Eye Research*, Elsevier, v. 65, n. 2, p. 205–214, 1997. Citado na página 20.

FLURY, B.; 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*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 7, n. 1, p. 169–184, jan. 1986. ISSN 0196-5204. Disponível em: <http://dx.doi.org/10.1137/0907013>. Citado 2 vezes nas páginas 63 e 65.

GALL, D. J. L. The MPEG video compression algorithm. *Signal Processing: Image Communication*, Elsevier, v. 4, n. 2, p. 129–140, 1992. Citado na página 20.

GODARA, L. C. Application of the fast Fourier transform to broadband beamforming. *The Journal of the Acoustical Society of America*, ASA, v. 98, n. 1, p. 230–240, 1995. Citado na página 19.

GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. [S.l.]: Upper Saddle River, NJ: Prentice Hall, 2012. Citado 4 vezes nas páginas 19, 28, 29 e 30.

GOYAL, V. K. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, v. 18, n. 5, p. 9–21, set. 2001. ISSN 1053-5888. Citado na página 63.

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. Citado na página 82.

HAWEEL, T. I. A new square wave transform based on the DCT. *Signal Processing*, v. 82, p. 2309–2319, nov. 2001. Citado 3 vezes nas páginas 37, 64 e 79.

HEIDEMAN, M. T.; BURRUS, C. S. *Multiplicative complexity, convolution, and the DFT*. [S.l.]: Springer-Verlag, 1988. (Signal Processing and Digital Filtering). ISBN 978-0-387-96810-0. Citado 2 vezes nas páginas 20 e 27.

HELMS, H. Fast Fourier transform method of computing difference equations and simulating filters. *IEEE Transactions on Audio and Electroacoustics*, IEEE, v. 15, n. 2, p. 85–90, 1967. Citado na página 19.

HEYDT, G. et al. Application of the Hartley transform for the analysis of the propagation of nonsinusoidal waveforms in power systems. *IEEE Transactions on Power Delivery*, IEEE, v. 6, n. 4, p. 1862–1868, 1991. Citado na página 20.

HIGHAM, N. J. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, v. 7, n. 4, p. 1160–1174, out. 1986. Citado na página 36.

HIGHAM, N. J. Computing real square roots of a real matrix. *Linear Algebra and its Applications*, v. 88–89, p. 405–430, abr. 1987. ISSN 0024-3795. Citado na página 36.

HIGHAM, N. J. *Functions of Matrices: Theory and Computation*. [S.l.]: Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2008. (SIAM e-books). ISBN 978-0-89871-777-8. Citado na página 36.

HIGHAM, N. J.; SCHREIBER, R. S. *Fast Polar Decomposition of an Arbitrary Matrix*. Ithaca, NY, USA, 1988. Citado na página 36.

HORADAM, K. J. *Hadamard Matrices and Their Applications*. [S.l.]: Princeton University Press, 2007. ISBN 978-0-691-11921-2. Citado na página 37.

HOU, H. S. A fast recursive algorithm for computing the discrete cosine transform. *IEEE Transactions on Acoustic, Signal, and Speech Processing*, v. 6, n. 10, p. 1455–1461, out. 1987. Citado 2 vezes nas páginas 32 e 35.

IEEE TRANSACTIONS ON SIGNAL PROCESSING. Disponível em: <http://ieeexplore.ieee.org/xpl/aboutJournal.jsp?punumber=78.> Citado na página 19.

International Telecommunication Union. *ITU-T Recommendation H.261 Version 1: Video Codec for Audiovisual Services at* $p \times 64$ *kbits*. [S.l.], 1990. Citado na página 20.

International Telecommunication Union. *ITU-T Recommendation H.263 version 1: Video Coding for Low Bit Rate Communication*. [S.l.], 1995. Citado na página 20.

JLEED, H.; BOUCHARD, M. Acoustic environment classification using discrete Hartley transform features. In: IEEE. *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. [S.l.], 2017. p. 1–4. Citado na página 20.

Joint Collaborative Team on Video Coding (JCT-VC). *HEVC Reference Software documentation*. 2013. Fraunhofer Heinrich Hertz Institute. Disponível em: <https://hevc.hhi.fraunhofer.de/>. Citado na página 78.

JRIDI, M.; ALFALOU, A.; MEHER, P. K. A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal approximation of DCT. *IEEE Trans. Circuits Syst.*, v. 62, n. 2, p. 449–457, 2015. Citado 4 vezes nas páginas 22, 79, 81 e 89.

KASBAN, H. A spiral based image watermarking scheme using Karhunen–Loève and discrete Hartley transforms. *Multidimensional Systems and Signal Processing*, Springer, v. 28, n. 2, p. 573–595, 2017. Citado na página 20.

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. Citado 2 vezes nas páginas 63 e 64.

KAY, S. M. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Upper Saddle River, NJ: Prentice Hall, 1993. (Prentie Hall Signal Processing Series). Citado na página 20.

KLATT, D. H.; KLATT, L. C. Analysis, synthesis, and perception of voice quality variations among female and male talkers. *The Journal of the Acoustical Society of America*, ASA, v. 87, n. 2, p. 820–857, 1990. Citado na página 19.

KOZHEMIAKIN, R. A. et al. Filtering of dual-polarization radar images based on discrete cosine transform. In: IEEE. *15th International Radar Symposium (IRS), 2014*. [S.l.], 2014. p. 1–4. Citado na página 20.

LEI, M. et al. Audio zero-watermark scheme based on discrete cosine transform-discrete wavelet transform-singular value decomposition. *China Communications*, IEEE, v. 13, n. 7, p. 117–121, 2016. Citado na página 20.

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, nov. 2004. ISSN 1051-8215. Citado 2 vezes nas páginas 38 e 79.

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, dez. 2001. Citado na página 63.

LOEFFLER, C.; LIGTENBERG, A.; MOSCHYTZ, G. Practical fast 1D DCT algorithms with 11 multiplications. In: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. [S.l.: s.n.], 1989. p. 988–991. Citado 3 vezes nas páginas 33, 35 e 79.

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. ISSN 1051-8215. Citado na página 20.

MADANAYAKE, A. et al. A row-parallel 8×8 2-D DCT architecture using algebraic integer-based exact computation. *IEEE transactions on circuits and systems for video technology*, IEEE, v. 22, n. 6, p. 915–929, 2012. Citado na página 35.

MEHER, P. K. et al. Efficient integer DCT architectures for HEVC. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 24, n. 1, p. 168–178, Jan 2014. Citado na página 80.

MOURA, J. What is signal processing? [president's message]. *IEEE Signal Processing Magazine*, v. 26, n. 6, p. 6, Nov 2009. ISSN 1053-5888. Citado na página 19.

NASSAR, S. S. et al. Efficient audio integrity verification algorithm using discrete cosine transform. *International Journal of Speech Technology*, Springer, v. 19, n. 1, p. 1–8, 2016. Citado na página 20.

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. ISSN 1051-8215. Citado 4 vezes nas páginas 67, 78, 79 e 81.

OPPENHEIM, A. V. *Discrete-time signal processing*. 3rd. ed. [S.l.]: Pearson Education India, 1999. Citado na página 26.

PAO, I.-M.; SUN, M.-T. Approximation of calculations for forward discrete cosine transform. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 8, n. 3, p. 264–268, jun. 1998. ISSN 1051-8215. Citado na página 71.

PEI, S.-C.; JAW, S.-B. Computation of discrete Hilbert transform through fast Hartley transform. *IEEE Transactions on Circuits and Systems*, IEEE, v. 36, n. 9, p. 1251–1252, 1989. Citado na página 20.

PERERA, K. S. et al. Modeling large time series for efficient approximate query processing. In: SPRINGER. *International Conference on Database Systems for Advanced Applications*. [S.l.], 2015. p. 190–204. Citado na página 19.

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*, IEEE, v. 61, n. 6, p. 1727–1740, 2014. Citado 2 vezes nas páginas 21 e 71.

POULARIKAS, A. D. *Transforms and applications handbook*. [S.l.]: CRC press, 2010. Citado 2 vezes nas páginas 20 e 26.

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, jul. 2012. ISSN 2162-2248. Citado 3 vezes nas páginas 20, 78 e 79.

PRIEMER, R. *Introductory signal processing*. [S.l.]: World Scientific Publishing Company, 1990. v. 6. Citado na página 19.

RAM, B. Digital image watermarking technique using discrete wavelet transform and discrete cosine transform. *International journal of Advancements in Research & technology*, v. 2, n. 4, p. 19–27, 2013. Citado na página 20.

RANSOM, S. M.; EIKENBERRY, S. S.; MIDDLEDITCH, J. Fourier techniques for very long astrophysical time-series analysis. *The Astronomical Journal*, IOP Publishing, v. 124, n. 3, p. 1788, 2002. Citado na página 19.

RAO, K. R.; YIP, P. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. San Diego, CA: Academic Press, 1990. Citado na página 32.

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, November 1970. ISSN 0018-9448. Citado na página 29.

REDDY, B. S.; CHATTERJI, B. N. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Tansactions on Image Processing*, IEEE, v. 5, n. 8, p. 1266–1271, 1996. Citado na página 19.

SALOMON, D.; MOTTA, G.; BRYANT, D. *Data Compression: The Complete Reference*. [S.l.]: Springer, 2007. (Molecular biology intelligence unit). ISBN 9781846286032. Citado 2 vezes nas páginas 32 e 72.

SAPONARA, S.; NERI, B. Radar sensor signal acquisition and multidimensional FFT processing for surveillance applications in transport systems. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 66, n. 4, p. 604–615, 2017. Citado na página 19.

SCHARNHORST, K. Angles in complex vector spaces. *Acta Applicandae Mathematica*, Springer, v. 69, n. 1, p. 95–103, 2001. Citado na página 48.

SEBER, G. A. F. *A Matrix Handbook for Statisticians*. [S.l.]: Wiley, 2008. (Wiley Series in Probability and Mathematical Statistics). ISBN 9780470226780. Citado 3 vezes nas páginas 24, 28 e 53.

SEYDNEJAD, S. R.; AKHZARI, S. Performance evaluation of pre-and post-FFT beamforming methods in pilot-assisted SIMO-OFDM systems. *Telecommunication Systems*, Springer, v. 61, n. 3, p. 471–487, 2016. Citado na página 19.

SHRUTHI, K. et al. Comparison analysis of a biomedical image for compression using various transform coding techniques. In: IEEE. *IEEE 6th International Conference on Advanced Computing (IACC), 2016*. [S.l.], 2016. p. 297–303. Citado na página 20.

STRANG, G. *Linear Algebra and Its Applications*. [S.l.]: Brooks Cole, 1988. Hardcover. ISBN 0-15-551005-3. Citado 2 vezes nas páginas 30 e 44.

STRANG, G. Wavelets. *American Scientist*, Sigma Xi, The Scientific Research Society, v. 82, n. 3, p. 250–255, 1994. ISSN 00030996. Citado na página 19.

SULLIVAN, G. J. et al. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, v. 22, n. 12, p. 1649–1668, dez. 2012. Citado na página 78.

TABLADA, C. J.; 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. Citado 3 vezes nas páginas 38, 44 e 78.

TSENG, C.-C.; LEE, S.-L. Digital image sharpening using Riesz fractional order derivative and discrete Hartley transform. In: IEEE. *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2014*. [S.l.], 2014. p. 483–486. Citado na página 20.

UNIVERSITY OF SOUTHERN CALIFORNIA. *USC-SIPI Image Database*. Disponível em: <http://sipi.usc.edu/database/>. Citado 3 vezes nas páginas 72, 73 e 101.

VILLASENOR, J. D. Optical Hartley transforms. *Proceedings of the IEEE*, IEEE, v. 82, n. 3, p. 391–399, 1994. Citado na página 20.

WALLACE, G. K. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, v. 38, n. 1, p. xviii–xxxiv, February 1992. ISSN 0098-3063. Citado 3 vezes nas páginas 20, 34 e 71.

WANG, Z.; BOVIK, A. C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, v. 26, n. 1, p. 98–117, jan. 2009. ISSN 1053-5888. Citado 3 vezes nas páginas 63, 64 e 72.

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, abr. 2004. ISSN 1057-7149. Citado na página 72.

WATKINS, D. S. *Fundamentals of Matrix Computations*. [S.l.]: Wiley, 2004. (Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts). ISBN 978-0-471-46167-8. Citado 2 vezes nas páginas 36 e 63.

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. Citado na página 32.

YUAN, W.; HAO, P.; XU, C. Matrix factorization for fast DCT algorithms. In: *IEEE International Conference on Acoustic, Speech, Signal Processing (ICASSP)*. [S.l.: s.n.], 2006. v. 3, p. 948–951. ISSN 1520-6149. Citado 2 vezes nas páginas 33 e 35.

ZHANG, X.; WU, H.; MA, Y. A new auto-focus measure based on medium frequency discrete cosine transform filtering and discrete cosine transform. *Applied and Computational Harmonic Analysis*, Elsevier, v. 40, n. 2, p. 430–437, 2016. Citado na página 20.

APÊNDICE A – NEW APPROXIMATIONS FOR THE 8-POINT DCT

In this appendix, the representative approximations for the 8-point DCT matrix derived from the proposed method are displayed.

A.1 NEW APPROXIMATIONS OBTAINED FROM SCHEME I

$$\mathbf{T}_{\mathrm{I},2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1/2 & 0 & 0 & -1/2 & -1 & -1 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 1 & 0 & -1 & -1/2 & 1/2 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1/2 & -1 & 0 & 1 & -1 & 0 & 1 & -1/2 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ 0 & -1/2 & 1 & -1 & 1 & -1 & 1/2 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{I},3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 1 & 1 & -1 & -1 & -3 & -3 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\ 3 & -1 & -3 & -1 & 1 & 3 & 1 & -3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -3 & 1 & 3 & -3 & -1 & 3 & -1 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -1 & 3 & -3 & 3 & -3 & 1 & -1 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{I},4} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1/2 & 1/4 & -1/4 & -1/2 & -1 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & -1/4 & -1 & -1/2 & 1/2 & 1 & 1/4 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1/4 & 1 & -1 & -1/4 & 1 & -1/2 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 1/4 & -1/2 & 1 & -1 & 1 & -1 & 1/2 & -1/4 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{I},5} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1/2 & 0 & 0 & -1/2 & -1 & -1 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 1 & 0 & -1 & -1/2 & 1/2 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1/2 & -1 & 0 & 1 & -1 & 0 & 1 & -1/2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 0 & -1/2 & 1 & -1 & 1 & -1 & 1/2 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{I},6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 2 & 1/2 & -1/2 & -2 & -3 & -3 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 3 & -1/2 & -3 & -2 & 2 & 3 & 1/2 & -3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 2 & -3 & 1/2 & 3 & -3 & -1/2 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1/2 & -2 & 3 & -3 & 3 & -3 & 2 & -1/2 \end{bmatrix}$$

A.2 NEW APPROXIMATIONS OBTAINED FROM SCHEME II

$$\mathbf{T}_{\mathrm{II},3} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 & -1 & -2 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -2 & 2 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 2 & -2 & 0 & 1 & -1 & 0 & 2 & -2 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{II},4} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1/2 & 1 & 0 & 0 & -1 & -1/2 & -1 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 1 & 0 & -1 & -1/2 & 1/2 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1/2 & -1 & 0 & 1 & -1 & 0 & 1 & -1/2 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ 0 & -1 & 1/2 & -1 & 1 & -1/2 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{II},5} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 1 & 1 & -1 & -1 & -3 & -3 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\ 1 & 1 & -3 & -3 & 3 & 3 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 3 & -3 & -1 & 1 & -1 & 1 & 3 & -3 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -1 & 3 & -3 & 3 & -3 & 1 & -1 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{II},6} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 1 & 3 & -1 & 1 & -3 & -1 & -3 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\ 3 & -1 & -3 & -1 & 1 & 3 & 1 & -3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -3 & 1 & 3 & -3 & -1 & 3 & -1 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ -1 & -3 & 1 & -3 & 3 & -1 & 3 & 1 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{II},7} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 1 & 1/2 & -1/2 & -1 & -2 & -2 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 1 & 1/2 & -2 & -2 & 2 & 2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 2 & -2 & -1/2 & 1 & -1 & 1/2 & 2 & -2 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ 1/2 & -1 & 2 & -2 & 2 & -2 & 1 & -1/2 \end{bmatrix}$$

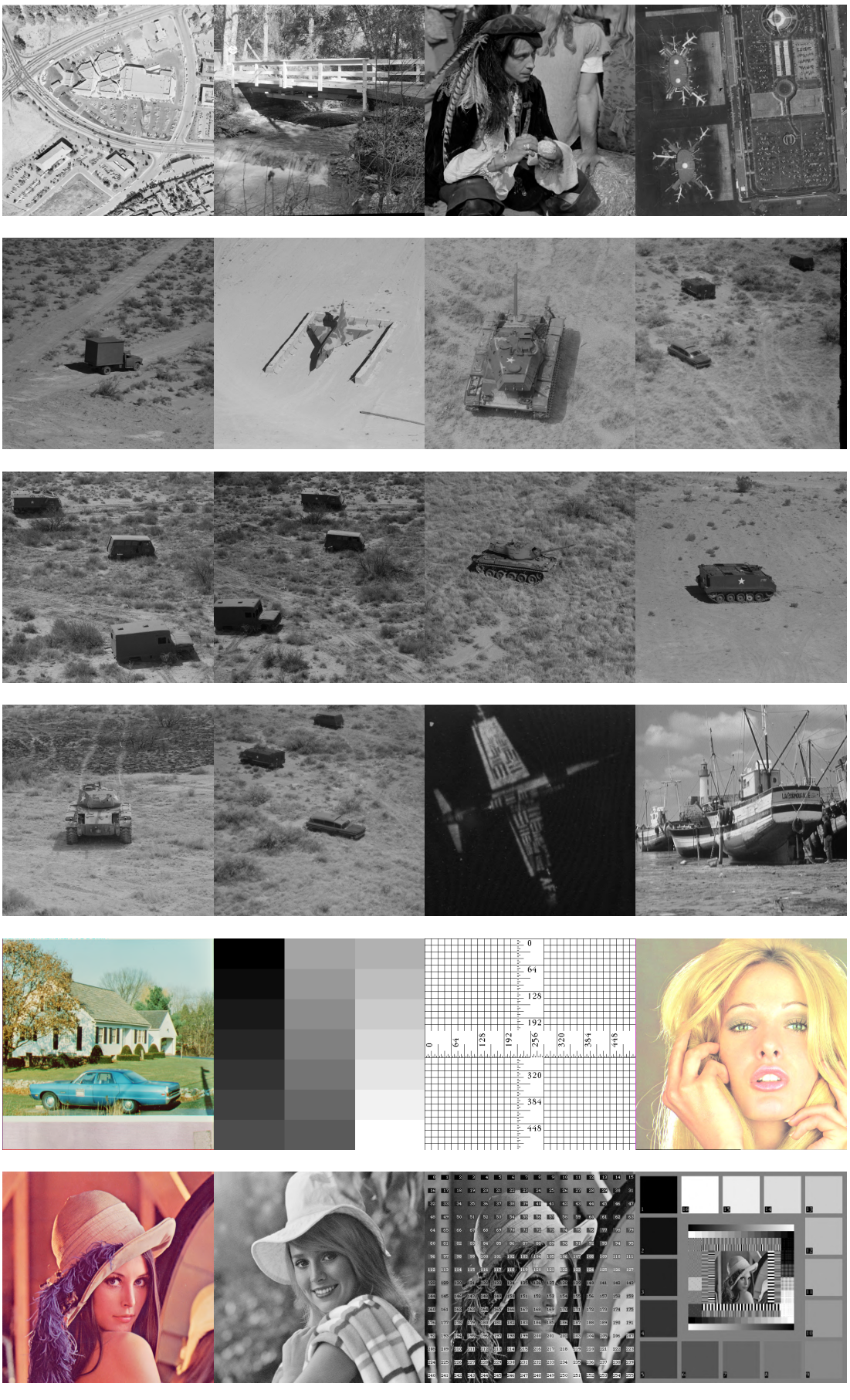
$$\mathbf{T}_{\mathrm{II},8} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 2 & -1/2 & 1/2 & -2 & -1 & -2 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 2 & -1/2 & -2 & -1 & 1 & 2 & 1/2 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 1/2 & 2 & -2 & -1/2 & 2 & -1 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ -1/2 & -2 & 1 & -2 & 2 & -1 & 2 & 1/2 \end{bmatrix}$$

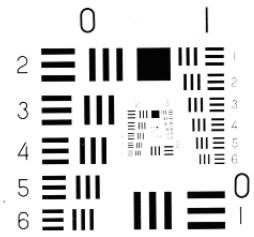
$$\mathbf{T}_{\mathrm{II},9} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 2 & 1/2 & -1/2 & -2 & -3 & -3 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 2 & 1/2 & -3 & -3 & 3 & 3 & -1/2 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 3 & -3 & -1/2 & 2 & -2 & 1/2 & 3 & -3 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ 1/2 & -2 & 3 & -3 & 3 & -3 & 2 & -1/2 \end{bmatrix}$$

$$\mathbf{T}_{\mathrm{II},10} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 2 & 3 & -1/2 & 1/2 & -3 & -2 & -3 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 3 & -1/2 & -3 & -2 & 2 & 3 & 1/2 & -3 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 2 & -3 & 1/2 & 3 & -3 & -1/2 & 3 & -2 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ -1/2 & -3 & 2 & -3 & 3 & -2 & 3 & 1/2 \end{bmatrix}$$

APÊNDICE B – IMAGE DATABASE

In this appendix, the images considered for the image compression experiments, from the USC-SIPI Database (UNIVERSITY OF SOUTHERN CALIFORNIA,), are displayed for reference.


0
64
128
192
256
320
384
448


0
1
2
3
4
5
6
0
1