

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



RECIFE, FEVEREIRO DE 2015

VÍTOR DE ANDRADE COUTINHO

APROXIMAÇÕES PARA A DCT VIA
“PRUNING” COM APLICAÇÕES EM
CODIFICAÇÃO DE IMAGEM E VÍDEO

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**

ORIENTADOR: PROF. DR. RENATO J. CINTRA

COORIENTADOR: PROF. DR. RICARDO M. CAMPELLO DE SOUZA

Recife, fevereiro de 2015.

©Vitor de Andrade Coutinho, 2015

Catálogo na fonte
Bibliotecária Valdicéa Alves, CRB-4 / 1260.

C871a Coutinho, Vítor de Andrade.
Aproximações para DCT via pruning com aplicações em codificação de imagem e vídeo / Vítor de Andrade Coutinho - Recife: O Autor, 2015.
153folhas, Il. e Tabs.

Orientador: Prof. Dr. Renato J. Cintra.
Coorientador: Prof. Dr. Ricardo M. Campello de Souza.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG.
Programa de Pós-Graduação em Engenharia Elétrica, 2015.
Inclui Referências, Apêndices, glossário e Acrônimo.

1. Engenharia Elétrica. 2. Compressão de dados. 3. DCT aproximada.
4. DCT pruning. 5. Compressão de imagem. I. Cintra, Renato J. (Orientador). II.
Souza, Ricardo M. Campello de. (Coorientador). III. Título.

UFPE

621.3 CDD (22. ed.)

BCTG/2015 - 247



Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE
DISSERTAÇÃO DO MESTRADO ACADÊMICO DE

VÍTOR DE ANDRADE COUTINHO

TÍTULO

**“APROXIMAÇÕES PARA A DCT VIA *PRUNING*
COM APLICAÇÕES EM CODIFICAÇÃO DE IMAGEM E VÍDEO”**

A comissão examinadora composta pelos professores: RICARDO MENEZES CAMPELLO DE SOUZA, DES/UFPE, JOÃO HENRIQUE RANHEL RIBEIRO, DES/UFPE e FÁBIO MARIANO BAYER, DE/UFMS aob a presidência do primeiro, consideram o candidato **VÍTOR DE ANDRADE COUTINHO APROVADO.**

Recife, 23 de fevereiro de 2015.

CECILIO JOSÉ LINS PIMENTEL
Coordenador do PPGE

**RICARDO MENEZES CAMPELLO DE
SOUZA**
Membro Titular Interno

FÁBIO MARIANO BAYER
Membro Titular Externo

JOÃO HENRIQUE RANHEL RIBEIRO
Membro Titular Externo

Dedico a Glauce, Henrique, Clara,
Gusta, Ana, Memena e Tonho.

AGRADECIMENTOS

Primeiramente, à minha mãe, Glauce Gouveia, ao meu pai, Henrique Guimarães, e à minha irmã, Clara de Andrade, pela dedicação, suporte, incentivo, amizade e amor incondicional.

Aos meus avós, Maria Augusta, Ana Maria e Francisco, por todo o carinho e amor que me proporcionam e pelo acolhimento de sempre. Ao meu avô Antônio Carlos (*in memoriam*) pelos poucos e preciosos momentos em que pude desfrutar de sua companhia.

Aos meus tios Marcus, Raisia, Polyana, Ramon e Pablo pelo companheirismo, amizade e inúmeros bons momentos vividos juntos.

A Fabíola, pelos imprescindíveis incentivos, carinho, compreensão e companhia constante na reta final do trabalho.

À família Holanda Tavares, pelos momentos inesquecíveis, pela grande influência na minha personalidade e a quem tenho eterno amor, admiração e gratidão.

Aos colegas do mestrado, do grupo de processamento de sinais e de graduação, especialmente aos “Moscas Brancas”, originalmente um time de futebol, atualmente uma família.

A Sil, Max, Andiará, José, Zé, Bruno, Vinícius, Leo, Neto, Pedrinho, Juliano, Méfner, Allissa, Ju, Anax, Pedro, Savinho *et al.*

A toda a minha família e amigos, por tudo o que representam para mim.

Ao meu orientador, Prof. Dr. Renato J. Cintra, pelos ensinamentos, dedicação, motivação, atenção, paciência, pelas cobranças oportunas e por acreditar na minha capacidade, me propondo um desafio extraclasse durante disciplina e que viria a resultar no presente trabalho.

Ao meu coorientador, Prof. Dr. Ricardo M. Campello de Souza, pelos ensinamentos durante minha graduação e pós-graduação, pela ajuda fundamental para ingressar no PPGEE, pela grande influência no meu interesse por processamento de sinais e pela disponibilidade.

Ao meu orientador de IC durante a graduação, Prof. Dr. Hélio M. de Oliveira, pelos ensinamentos científicos iniciais, pelo entusiasmo, disponibilidade e incentivo.

Aos Prof. Dr. Fábio Bayer, Dr. Arjuna Madanayake e Sunera Kulasekera, pelas contribuições e pelos trabalhos realizados em parceria.

A todos os meus professores da pós-graduação, da graduação e do período escolar, que fizeram parte da minha formação.

Ao CNPq, pelo suporte financeiro.

VÍTOR DE ANDRADE COUTINHO

Universidade Federal de Pernambuco

23 de fevereiro de 2015

Resumo da Dissertação apresentada à UFPE como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

**APROXIMAÇÕES PARA A DCT VIA “PRUNING”
COM APLICAÇÕES EM CODIFICAÇÃO DE
IMAGEM E VÍDEO**

Vítor de Andrade Coutinho

fevereiro/2015

Orientador: Prof. Dr. Renato J. Cintra

Coorientador: Prof. Dr. Ricardo M. Campello de Souza

Área de Concentração: Comunicações

Palavras-chaves: Compressão de dados, DCT aproximada, aproximação DCT, DCT pruning, compressão de imagem, JPEG, MPEG, H264, HEVC.

Número de páginas: 155

O presente trabalho aborda o desenvolvimento de aproximações para a transformada discreta do cosseno (DCT) utilizando a abordagem *pruning*. Devido à propriedade da compactação de energia, a DCT é empregada em diversas aplicações de compressão de dados. Embora algoritmos rápidos permitam computar a DCT eficientemente, operações de multiplicação são inevitáveis. Devido a crescente demanda por métodos de baixo consumo energético, novos algoritmos de custo computacional reduzido são necessários. Neste contexto, aproximações para a DCT foram propostas nos últimos anos. Tais aproximações permitem algoritmos livres de multiplicação, sem a necessidade de operações de ponto flutuante, mantendo o desempenho de compressão comparável ao fornecido por métodos baseados na DCT. Uma abordagem adicional para reduzir o custo computacional da DCT é a utilização de *pruning*. Tal técnica consiste em não considerar coeficientes dos vetores de entrada e/ou saída que apresentam menor relevância em termos de energia concentrada. No caso da DCT, esses coeficientes são os termos de mais alta frequência do vetor transformado. A aplicação de *pruning* a aproximações para a DCT é uma área pouco explorada. O objetivo deste trabalho é aplicar a técnica a diferentes métodos aproximados para a DCT. As transformações resultantes foram aplicadas no contexto de compressão de imagem e vídeo e os resultados mostraram desempenho comparável ao de métodos exatos a um custo computacional bastante reduzido. Uma generalização do conceito é apresentada, assim como uma análise da complexidade aritmética.

Abstract of Dissertation presented to UFPE as a partial fulfillment of the requirements for
the degree of Master in Electrical Engineering

PRUNED DCT APPROXIMATIONS WITH APPLICATIONS IN IMAGE AND VIDEO CODING

Vítor de Andrade Coutinho

February/2015

Supervisor: Prof. Dr. Renato J. Cintra

Co-supervisor: Prof. Dr. Ricardo M. Campello de Souza

Area of Concentration: Communications

Keywords: Data compression, approximate DCT, DCT approximation, pruned DCT, image compression, JPEG, MPEG, H264, HEVC.

Number of pages: 155

This work introduces approximate discrete cosine transforms (DCT) based on the pruning approach. Due to the energy compaction property, the DCT is employed in several data compression applications. Although fast algorithms allow an efficient DCT computation, multiplication operations are inevitable. Due to the increasing demand for energy efficient methods, new algorithms with reduced computational cost are required. In this context, DCT approximations have been proposed recently. Such approximations allow multiplication free algorithms which can avoid floating point operations, while maintaining a competitive performance. A further approach to reduce the computational cost of the DCT is pruning. The technique consists of discarding input and/or output vectors coefficients which are regarded as less significant. In the case of the DCT, such coefficients are output coefficients associated to higher frequency terms. Pruned DCT approximations is a relatively unexplored field of research. The objective of this work is the combination of approximations and pruning to derive extremely low-complexity DCT approximations. The resulting methods were applied in the image and video compression scenario and results showed comparative performance with exact methods at a much lower computational complexity. A qualitative and quantitative comparison with a comprehensive list of existing methods is presented. A generalization of the pruning concept is presented.

LISTA DE FIGURAS

2.1	Extensão $y_{II}[n]$ para a DCT-II.	26
2.2	Extensão $y_I[n]$ para a DCT-I.	28
2.3	Extensão $y_{III}[n]$ para a DCT-III.	29
2.4	Extensão $y_{IV}[n]$ para a DCT-IV.	30
2.5	Estrutura borboleta.	33
2.6	Algoritmo rápido para transformada de Walsh-Hadamard	42
3.1	Funções de Walsh.	53
3.2	Algoritmo rápido para transformada de Walsh-Hadamard.	54
4.1	Algoritmo rápido para transformada de Walsh-Hadamard na ordem natural.	68
4.2	Algoritmo rápido para exemplo de transformada de Walsh-Hadamard podada.	68
5.1	Codificador de imagem por transformada de bloco.	97
5.2	Decodificador de imagem por transformada de bloco.	98
5.3	Distribuição relativa de energia $\mathcal{E}_{rel,dB}(n,m)$ para a DCT exata, WHT, SDCT, LODCT, RDCT e MRDCT.	101
5.4	Distribuição relativa de energia $\mathcal{E}_{rel,dB}(n,m)$ para os métodos da Série BAS.	102
5.5	Distribuição relativa de energia $\mathcal{E}_{rel,dB}(n,m)$ para os métodos baseados em funções inteiras.	103
5.6	Curvas de PSNR médio para cada método.	108
5.7	Curvas de SSIM médio para cada método.	111
5.8	Curvas de PSNR médio divididas pelo ganho de codificação C_g de cada método.	112
5.9	Compressão da imagem ‘Lena’ através de métodos derivados da WHT.	113
5.10	Compressão da imagem ‘Lena’ através de métodos derivados da SDCT.	114
5.11	Compressão da imagem ‘Lena’ através de métodos derivados da LODCT.	115
5.12	Compressão da imagem ‘Lena’ através de métodos derivados da RDCT.	116
5.13	Compressão da imagem ‘Lena’ através de métodos derivados da MRDCT.	117
5.14	Compressão da imagem ‘Lena’ através de métodos derivados da BAS-1.	118
5.15	Compressão da imagem ‘Lena’ através de métodos derivados da INT-6.	119
5.16	Arquitetura 1-D para a MRDCT podada.	120
5.17	Arquitetura 1-D para a LODCT podada.	120
5.18	Dispositivo BEE3 utilizado para implementar as arquiteturas.	121

A.1	Algoritmos rápidos para aproximações derivadas da SDCT.	137
A.2	Algoritmos rápidos para aproximações derivadas de BAS-1.	138
A.3	Algoritmos rápidos para aproximações derivadas de BAS-2.	139
A.4	Algoritmos rápidos para aproximações derivadas de BAS-3.	140
A.5	Algoritmos rápidos para aproximações derivadas de BAS-4, BAS-5 e BAS-6. .	141
A.6	Algoritmos rápidos para aproximações derivadas de BAS-7.	142
A.7	Algoritmos rápidos para aproximações derivadas da RDCT.	143
A.8	Algoritmos rápidos para aproximações derivadas da MRDCT.	144
A.9	Algoritmos rápidos para aproximações derivadas da LODCT e das INT- i . . .	145
A.10	Blocos $A_{(i)}$ para os algoritmos da Figura A.9.	146
B.1	Imagens aplicadas ao procedimento de compressão de imagens.	148

LISTA DE TABELAS

2.1	Complexidade aritmética para os algoritmos descritos	45
3.1	Aproximações da série BAS	58
3.2	Aproximações ortogonais baseadas em funções inteiras	61
3.3	Medidas de desempenho para os métodos abordados	64
4.1	Aproximações SDCT podadas	75
4.2	Aproximações BAS-1 podadas	76
4.3	Aproximações BAS-2 podadas	77
4.4	Aproximações BAS-3 podadas	77
4.5	Aproximações BAS-4 podadas	78
4.6	Aproximações BAS-5 podadas	78
4.7	Aproximações BAS-6 podadas	79
4.8	Aproximações BAS-7 podadas	79
4.9	Aproximações LODCT podadas	80
4.10	Aproximações RDCT podadas	81
4.11	Aproximações MRDCT podadas	82
4.12	Aproximações INT-1 podadas	83
4.13	Aproximações INT-2 podadas	83
4.14	Aproximações INT-4 podadas	84
4.15	Aproximações INT-5 podadas	84
4.16	Aproximações INT-6 podadas	85
4.17	Complexidade aritmética para as aproximações podadas apresentadas (adições/ deslocamento de bits)	89
5.1	Aplicações com compressão de vídeo	93
5.2	Energia relativa por zona para cada método	100
5.3	Valores de PSNR médio para cada método	109
5.4	Valores de SSIM médio para cada método	110
5.5	Consumo de recurso de <i>hardware</i> utilizando Xilinx Virtex-5 para a MRDCT e LODCT	110
5.6	Consumo de recurso de <i>hardware</i> utilizando Xilinx Virtex-6 para a MRDCT podada	111

5.7 Consumo de recursos para síntese CMOS 45 nm ASIC para a MRDCT podada 112

A.1 Associação entre métodos propostos, figuras e parâmetros de algoritmos rápidos 136

SUMÁRIO

Lista de Figuras	6
Lista de Tabelas	8
1 INTRODUÇÃO	13
1.1 Motivação	14
1.2 Revisão da Literatura	16
1.3 Objetivos	18
1.4 Estrutura	19
2 A TRANSFORMADA DISCRETA DO COSSENO	21
2.1 A Transformada de Karhunen-Loève	22
2.1.1 Solução Markov-1	24
2.2 Definições da Transformada Discreta do Cosseno	24
2.3 A DCT como Caso Assintótico da KLT	30
2.4 Algoritmos Rápidos para a DCT	31
2.4.1 Algoritmo de Chen	32
2.4.2 Algoritmo de Lee	36
2.4.3 Algoritmo de Wang	37
2.4.4 Algoritmo de Vetterli–Nussbaumer	38
2.4.5 Algoritmo de Suehiro–Hatori	39
2.4.6 Algoritmo de Hou	41
2.4.7 Algoritmo de Loeffler	42
2.4.8 Algoritmo de Feig–Winograd	42
2.4.9 Comparação	44
3 APROXIMAÇÕES PARA A DCT	46
3.1 Introdução	47
3.2 Classes de Aproximações	47
3.3 Ortogonalidade e Ortonormalidade	49
3.4 Aproximações no Conjunto \mathbb{P}	52
3.4.1 Transformada de Walsh-Hadamard	52
3.4.2 A DCT Sinalizada	54

3.4.3	Aproximação de Lengwehasatit e Ortega	56
3.4.4	Séries Bouguezal-Ahmad-Swamy	56
3.4.5	A DCT arredondada e a DCT arredondada modificada	57
3.4.6	Série de Aproximações Baseadas em Funções Inteiras	59
3.5	Medidas de Desempenho	61
3.5.1	Erro Médio Quadrático	61
3.5.2	Ganho de Codificação	62
3.5.3	Eficiência da Transformada	63
3.5.4	Avaliação das Transformadas	63
4	APROXIMAÇÕES PODADAS	65
4.1	Preliminares	66
4.2	Transformada Discreta de Fourier Podada	70
4.3	Transformada Discreta do Cosseno Podada	71
4.4	Relação entre Complexidade Aritmética para Métodos Podados em 1-D e 2-D	71
4.5	Aproximações Podadas	73
4.5.1	DCT Sinalizada Podada	75
4.5.2	Aproximações BAS Podadas	76
4.5.3	Aproximação Lengwehasatit-Ortega Podada	76
4.5.4	DCT Arredondada Podada	80
4.5.5	DCT Arredondada Modificada Podada	81
4.5.6	Aproximações Baseadas em Funções Inteiras Podadas	81
4.5.7	Considerações	82
4.5.8	Cálculo da Transformação Inversa	85
5	APLICAÇÃO EM COMPRESSÃO DE IMAGENS: SIMULAÇÕES E RESULTADOS	90
5.1	Introdução	91
5.2	Métricas para o Desempenho de Compressão	93
5.2.1	Similaridade Estrutural	94
5.3	Compressão de Imagem Estática	96
5.3.1	Energia Concentrada por Coeficiente e por Zona	98
5.3.2	Metodologia da Simulação JPEG	104
5.3.3	Resultados	106
5.4	Arquiteturas VLSI	107
6	CONCLUSÕES	122
7	BIBLIOGRAFIA	126
Apêndice A ALGORITMOS RÁPIDOS PARA AS APROXIMAÇÕES PODADAS		135
Apêndice B IMAGENS UTILIZADAS NAS SIMULAÇÕES		147

Apêndice C TRABALHOS REALIZADOS	149
Glossário de Notação	151
Acrônimos	154

CAPÍTULO 1

INTRODUÇÃO

1.1 MOTIVAÇÃO

TRANSFORMADAS discretas são ferramentas vastamente utilizadas em diversas aplicações relacionadas a processamento digital de sinais [1]. Em particular, a transformada discreta do cosseno (DCT) foi apresentada em 1974 como uma alternativa à transformada de Karhunen-Loève (KLT) [2] para aplicações de descorrelação de dados. A KLT é estatisticamente ótima em relação à descorrelação de componentes de um vetor arbitrário. Entretanto, ela não possui um algoritmo direto para o cálculo computacional já que o núcleo da transformação depende do vetor a ser transformado [3]. Por outro lado, a DCT possui núcleo de transformação que não depende dos dados de entrada, o que possibilita o desenvolvimento de algoritmos rápidos [4]. Verifica-se que a DCT é o caso assintótico da KLT quando o sinal de entrada é um processo markoviano de primeira ordem e o coeficiente de correlação $\rho \rightarrow 1$ [3,4]. De fato, a KLT pode ser substituída pela DCT com grande eficiência para sinais fortemente correlacionados, como imagens e áudio.

A existência de algoritmos rápidos tornou a DCT vastamente utilizada em diversas aplicações e, devido à sua capacidade de compactação de energia, é largamente empregada na compressão de dados digitais. A DCT tem sido utilizada como principal ferramenta em diversos padrões de codificação de imagem e vídeo. Em suma, é utilizada nos seguintes padrões de compactação: JPEG [5], MPEG [6, 7], H.261 [8], H.263 [9], H.264/AVC [10, 11] e HEVC [12, 13].

A DCT de comprimento 8 desempenha um papel importante no cenário de compressão de imagem e vídeo. Todos os padrões de codificação citados acima utilizam tal comprimento durante o processo de compressão. Portanto, o desenvolvimento de algoritmos rápidos para este comprimento particular da DCT se tornou um importante campo de pesquisa. Diversos algoritmos rápidos foram desenvolvidos ao longo dos anos e podem ser encontrados na literatura [14–21]. A arquitetura desenvolvida pela indústria para a implementação da DCT de duas dimensões é baseada na separação de núcleo da DCT, também chamado de decomposição linha-coluna [3, 22]. Através deste método, a transformada em 2-D é computada através de algoritmos para a DCT de uma dimensão para cada linha e subsequentemente para cada coluna do sinal de duas dimensões. Entretanto, alguns algoritmos foram desenvolvidos abordando diretamente o caso de duas dimensões [21, 23]

Inicialmente, até 1980, a implementação da DCT em *hardware* era realizada utilizando componentes discretos ou processadores de sinais digitais (DSPs) [24–26]. Posteriormente,

foram desenvolvidos circuitos integrados de aplicação específica (ASICs) para processamento em tempo real da DCT operando em até 14.32 MHz [27].¹ Nos últimos anos, arquiteturas para aplicações em TV de alta definição (HDTV) vem sendo desenvolvidos [28,29].

Embora algoritmos rápidos diminuam drasticamente o esforço computacional necessário para o cálculo da DCT, as multiplicações por números reais são inevitáveis. O limite mínimo teórico para a quantidade de multiplicações no cálculo da DCT está estabelecido na literatura [30,31]. Os esforços em pesquisa na área levaram à produção de algoritmos que atingem ou chegam próximo da cota mínima teórica [20]. Portanto, a busca por algoritmos eficientes para o cálculo da DCT é uma área já bastante explorada e obter algoritmos mais eficientes não é uma atividade trivial.

Por outro lado, aproximações para a DCT podem reduzir drasticamente a quantidade de operações aritméticas e ainda gerar resultados satisfatórios. Uma aproximação para a DCT é uma transformação linear que preserva características da DCT original, como a compactação de energia e capacidade de decorrelação de vetores [32], por exemplo. Em geral, deseja-se que a aproximação possua a menor complexidade aritmética possível. Nesse contexto, os limites teóricos para a DCT não mais se aplicam. Sendo assim, é possível encontrar aproximações com custo computacional consideravelmente inferiores aos estabelecidos para a DCT [32–46].

Outra estratégia para diminuir o custo computacional no cálculo da DCT é reduzir o número de coeficientes calculados, ou seja, desprezar componentes de menor relevância para um dado problema. Devido a propriedade de compactação de energia da DCT, em muitas aplicações a maior parte da informação útil está concentrada nos primeiros coeficientes [47]. Dessa forma, o esforço computacional em calcular as últimas componentes é dispensado. Tal abordagem é chamada de *pruning*. Doravante, algoritmos que seguem essa metodologia serão chamados *algoritmos podados*. No contexto de compressão de imagens, onde a DCT de duas dimensões é considerada, a imagem comprimida é transformada e tem suas componentes mais altas desprezadas na etapa da quantização. Entretanto, em um algoritmo comum para a DCT de duas dimensões, tais componentes desprezadas na etapa de quantização são computadas igualmente a todas as outras. Neste cenário, algoritmos que computam apenas uma parte das componentes foram desenvolvidos e também são conhecidos como algoritmos *zonais* [48,49].

Embora diversas aproximações para a DCT e algoritmos podados para a DCT tenham sido propostos, a junção das duas abordagens ainda é uma área pouco explorada. Apenas em [50]

¹neste trabalho, é utilizado o ponto ‘.’ como separador decimal

foi introduzida uma aproximação podada para a DCT baseada na aproximação dada em [43]. Além de ser um campo inexplorado em termos de aplicações, conceitualmente também é uma área pouco desenvolvida. O conceito de *pruning* carece de uma generalização matemática e de uma formalização para aplicações com DCTs aproximadas, assim como uma análise algébrica sobre complexidade aritmética e transformação inversa.

1.2 REVISÃO DA LITERATURA

Em 1971, antes mesmo do surgimento da DCT, é desenvolvido por Markel para a transformada discreta de Fourier (DFT) o primeiro algoritmo utilizando a abordagem *pruning* [51]. O algoritmo se aplica a situações em que o número de amostras nulas do vetor de entrada é grande. Dessa forma, os cálculos envolvendo tais componentes são desconsiderados e a complexidade é reduzida. O algoritmo desenvolvido se mostrou útil em aplicações como interpolação, análise e síntese de voz e aproximação por mínimos quadrados com polinômios trigonométricos [51]. Apesar de aplicar *pruning* no domínio do tempo, isto é, eliminar componentes do vetor de entrada, Markel sugere que a mesma ideia pode ser aplicada para o domínio da frequência, eliminando componentes do vetor de saída. O algoritmo de Markel foi melhorado por Skinner em 1976 [52]. Ao apenas excluir algumas operações, Markel derivou um algoritmo com multiplicações repetidas em diferentes estágios do algoritmo. Skinner propôs uma maneira diferente de derivar o mesmo algoritmo baseado na eliminação de estágios inteiros, proporcionando um maior *time-saving*.

A DCT foi introduzida em 1974 por Ahmed–Natarajan–Rao [2]. Ahmed *et al.* estavam interessados em encontrar uma boa aproximação para a KLT que pudesse ser computada eficientemente. O algoritmo rápido sugerido por Ahmed *et al.* utilizava o fato de a DCT poder ser escrita em termos da parte real da DFT com o dobro do comprimento. Então, o problema de calcular eficientemente a DCT se resumia em aplicar um algoritmo rápido para a DFT (FFT).

Em 1977, Chen introduziu um algoritmo rápido para a DCT mais eficiente do que o que utiliza a FFT de comprimento dobrado [14]. O algoritmo se aplica a comprimentos de transformada da forma 2^m , $m \geq 2$ e é seis vezes mais econômico do que o algoritmo anterior. Em 1984, Lee derivou um algoritmo baseado na dizimação no tempo (DIT) para a DCT inversa (IDCT) de comprimento N , podendo ser aplicado para a DCT de comprimento N . Para a DCT de comprimento 8, o algoritmo apresenta 12 multiplicações e 29 adições.

Também em 1984, Wang apresentou novos algoritmos através de novo método de fatoração das matrizes de cada um dos quatro tipos da DCT [16]. Além disso, Wang propôs algoritmos também para quatro tipos da transformada discreta do seno (DST) e da transformada W discreta (DWT). O algoritmo proposto por Wang para a DCT de comprimento 8 necessita de 13 multiplicações e 29 adições. Neste mesmo ano, Lee propôs um algoritmo rápido que, para a DCT de comprimento 8, necessita de 12 multiplicações e 29 adições [15]. Entretanto, a necessidade de utilizar multiplicações pelo inverso do cosseno pode levar a problemas de *overflow* e erros de arredondamento. Ainda neste ano, Vetterli propôs um algoritmo que computa a DCT de comprimento N através de duas DFTs reais de comprimento N [17], atingindo também a cota de 12 multiplicações e 29 adições. Em 1986, Suehiro propôs um algoritmo que computa a DCT-II baseado na fatoração da matriz da DCT-IV [18]. O algoritmo apresenta também 12 multiplicações e 29 adições. Em 1987, Hou apresenta um algoritmo com a mesma cota [19].

A teoria da complexidade multiplicativa aplicada a processamento de sinais foi desenvolvida por S. Winograd [30] e organizada por M. Heideman em 1988 [31]. Heideman se inspirou nos trabalhos de Winograd sobre complexidade multiplicativa [30]. Em seu livro, oriundo de sua tese de doutorado, Heideman aborda a questão da complexidade multiplicativa mínima para os problemas do cálculo de convolução e multiplicação polinomial e do cálculo da DFT. A partir da teoria desenvolvida para a DFT, são derivadas as complexidades multiplicativas para a transformada discreta de Hartley (DHT) [31, p. 115] e para a DCT [31, p. 116]. O limite mínimo de 11 multiplicações para a DCT de comprimento 8 foi atingido por Loeffler, em 1989 [20]. O algoritmo obtido requer 11 multiplicações e 29 adições.

O primeiro algoritmo aplicando *pruning* para a DCT foi proposto em 1991, por Wang [47]. Wang argumentou que, em diversas aplicações, a maior parte da informação útil de um sinal está contido nos primeiros termos do vetor da DCT. Sendo assim, apenas esses primeiros elementos do vetor necessitariam ser calculados para essa aplicação.

Feig e Winograd introduziram, em 1992, um conjunto de algoritmos para a DCT com vetores de comprimentos da forma 2^m [21]. Casos particulares para a DCT de uma dimensão com comprimento 8 e de duas dimensões com tamanho 8×8 foram estudados em detalhe. O algoritmo proposto para 2-D apresenta um número menor de multiplicações do que o algoritmo de Loeffler quando aplicado para 2-D segundo a decomposição linha-coluna.

Em 1997, Servais propõe utilizar DCT de três dimensões para codificação de vídeo, ex-

pandindo a DCT para a dimensão temporal [53]. O método é realizado considerando grupos consecutivos de oito *frames* de dimensão 8×8 . Neste mesmo ano, Chan and Lee propuseram um novo método para gerar valores de quantização para coeficientes da DCT em 3-D visando esquemas de codificação de vídeo [54]. Em 2004, Boussakta propõe um algoritmo para a DCT computado diretamente em 3-D ao invés de utilizar o método usual de decomposição linha-coluna-*frame* [55]. Em 2010, o grupo de pesquisadores liderados por Dina Katabi propõe uma nova abordagem para transmissão *wireless* de vídeo denominada *SoftCast* [56]. A técnica utiliza DCT em 3-D na etapa de compressão e visa permitir o receptor a decodificar vídeo a uma taxa de *bits* que depende da qualidade do canal observada após a recepção.

Em 2001, Haweel apresenta a DCT sinalizada (*signed* DCT, SDCT) [33], que consiste em uma aproximação para a DCT baseada na função *senal*. A aproximação apresentava algoritmos rápidos para a transformação direta e inversa de baixa complexidade, sem a necessidade de operações de multiplicação. Em 2004, Lengwehasatit e Ortega apresentam um conjunto de métodos para redução de complexidade para a DCT direta [34]. Em tal trabalho, *pruning* é definido como *método de seleção de frequência*, enquanto aproximações são definidas como *método de seleção de precisão*.

Entre 2008 e 2013, Bouguezal *et al.* propõem uma série de aproximações para a DCT apresentando bom desempenho de codificação e baixas complexidades computacionais [35–40]. Em 2011, Cintra e Bayer apresentam uma aproximação para a DCT baseada em na função de arredondamento [41]. Em 2012, Bayer e Cintra propõem a aproximação para a DCT com menor custo computacional na literatura. Tal método requer apenas 14 adições [42]. Em 2013, Kouadria *et al.* propõem a abordagem *pruning* à aproximação BAS apresentada em [43] com aplicação no contexto de redes de sensores visuais *wireless* (WWSNs) [50], representando o único trabalho na literatura que aplica *pruning* a uma aproximação para a DCT.

1.3 OBJETIVOS

Métodos alternativos para diminuir o custo computacional da DCT em aplicações diversas e especialmente em compressão de imagem vem sendo explorados. Conforme mencionado anteriormente, tais métodos podem ser classificados como métodos de seleção de frequência ou de seleção de precisão [34]. Os métodos de seleção de frequência são baseados em computar apenas um subconjunto dos coeficientes espectrais, sendo também denominada DCT podada (*pruned* DCT) [47] ou, para o caso 2-D, DCT *zonal* [22, 48, 49]. Métodos de seleção

de precisão são representados pelas aproximações para a DCT. Métodos híbridos também são possíveis, sendo bastante inexplorados na literatura.

O objetivo principal deste trabalho é formalizar matematicamente o conceito de *pruning* (seleção de frequência) e aplicar a técnica a diversas aproximações para a DCT (seleção de precisão). É objetivada a proposição de novos métodos híbridos ou aproximações para a DCT via *pruning* com aplicação no contexto de compressão de imagem. Buscaremos realizar uma análise algébrica sobre a complexidade aritmética de tais métodos, assim como do procedimento para computar a transformação inversa. Objetiva-se realizar simulações em codificação de imagem com os métodos derivados e avaliar o desempenho de compressão através de métricas difundidas na literatura. Além disso, buscaremos propor algoritmos rápidos para os novos métodos apresentados.

Os objetivos deste trabalho são sumarizados a seguir:

- Realizar uma revisão sobre a DCT, apresentando definições, relação com a KLT, descrição e comparação entre os principais algoritmos rápidos disponíveis na literatura;
- Apresentar uma revisão sobre aproximações para a DCT, fornecer descrições e comparações entre diversos métodos aproximados disponíveis na literatura;
- Formalizar matematicamente e generalizar o conceito de *pruning*;
- Propor novos métodos aproximados e podados;
- Propor algoritmos rápidos podados para os métodos propostos;
- Analisar matematicamente a questão da transformada inversa e da complexidade aritmética para aproximações podadas;
- Realizar simulação de compressão de imagem para avaliar o desempenho dos novos métodos propostos.

1.4 ESTRUTURA

Este documento está organizado como se segue: No Capítulo 2, são discutidos e apresentados diversos aspectos relacionados a DCT. Na Seção 2.1, é abordada a KLT, transformação ótima em relação a descorrelação de dados, em que a DCT é tida como uma aproximação. Na Seção 2.2, são apresentadas definições para os diversos tipos de DCT. O desenvolvimento matemático da DCT através da DFT da expansão simétrica do sinal de entrada é abordado

e ilustrado para o caso da DCT-II. A Seção 2.3 apresenta a DCT como aproximação assintótica para a KLT para sinais com alta correlação, justificando o uso da DCT no contexto de compressão de dados. A Seção 2.4 apresenta uma coleção de algoritmos rápidos para a DCT disponíveis na literatura, descrevendo-os sucintamente e comparando as complexidades aritméticas de cada um.

O Capítulo 3 aborda aproximações para a DCT, em que as aproximações utilizadas para derivar novos métodos são apresentadas. A Seção 3.1 introduz o conceito de aproximação para a DCT, justificando sua importância. Na Seção 3.2, diferentes classes de aproximações são descritas de acordo com o tipo de entrada da matriz de transformação. A Seção 3.3 apresenta os conceitos de ortogonalidade e ortonormalidade no contexto de aproximações para a DCT. A Seção 3.4 apresenta as aproximações abordadas neste trabalho. Na Seção 3.5, são apresentadas e descritas medidas para avaliar o desempenho das aproximações no contexto de codificação.

No Capítulo 4, são apresentadas as aproximações derivadas através da aplicação de *pruning* e este conceito é formalizado matematicamente. Na Seção 4.1, o conceito de *pruning* é formalizado e generalizado. Alguns exemplos para ilustrar o método são apresentados. A Seção 4.2 apresenta a DFT podada, originalmente proposta em [51]. A Seção 4.3 descreve a DCT podada, apresentada em [47]. Na Seção 4.4, é realizada uma análise sobre a complexidade aritmética de transformações podadas em 1-D e 2-D. Na Seção 4.5, é definido o conceito de *pruning* para aproximações para a DCT e são apresentados os novos métodos propostos, baseados na aplicação da definição para as aproximações descritas no Capítulo 3.

No Capítulo 5, são abordadas, de forma geral, técnicas de compressão de imagem. Na Seção 5.1, uma abordagem introdutória é apresentada, em que conceitos de imagens digitais são abordados, assim como a importância da compressão. Na Seção 5.2, são apresentadas métricas usuais de avaliação de degradação de imagem, como o erro médio quadrático (MSE), relação sinal-ruído de pico (PSNR) e a similaridade estrutural (SSIM). A Seção 5.3 descreve o procedimento de compressão de imagem estática. Um estudo visando analisar a capacidade de compactação de energia dos métodos abordados é realizado. A metodologia para simulação tipo JPEG é descrita. Os resultados obtidos para cada um dos novos métodos gerados são apresentados e discutidos. A Seção 5.4 apresenta arquiteturas em *very-large-scale integration* (VLSI) para alguns dos métodos podados propostos. Esta Seção foi desenvolvida em parceria com Dr. Arjuna Madanayake e Sunera Kulasekera, pesquisadores da Universidade de Akron, EUA.

CAPÍTULO 2

A TRANSFORMADA DISCRETA DO COSSENO

2.1 A TRANSFORMADA DE KARHUNEN-LOÈVE

A KLT é derivada através da busca de uma transformação que descorrelacione completamente um dado vetor. Diferentes nomenclaturas equivalentes à KLT são encontrados na literatura, tais como análise de componente principal e transformada de Hotelling [57], termos propostos respectivamente nos trabalhos de K. Pearson [58] e H. Hotelling [59]. Um vetor correlacionado possui seus elementos com certo grau de redundância e grande parte dos problemas físicos reais lidam com sinais dessa forma. Em particular, imagens naturais que possuem significado ao cérebro humano são fortemente correlacionadas. Os vetores de base da KLT são os autovetores da matriz de covariância e, conseqüentemente, dependem do sinal de entrada a ser transformado. O efeito causado pela KLT é a diagonalização da matriz de covariância, removendo a correlação entre elementos distintos do vetor. A matriz de covariância de um vetor $\mathbf{y} = [y_0 \ y_1 \ \cdots \ y_{N-1}]^\top$ pode ser definida como [60]

$$\mathbf{R}_y = \text{E} [(\mathbf{y} - \mathbf{m})(\mathbf{y} - \mathbf{m})^\top], \quad (2.1)$$

em que $\text{E}[\cdot]$ é o operador valor esperado [61] e $\mathbf{m} = \text{E}[\mathbf{y}]$. O valor esperado é um operador linear e, se aplicado a um vetor ou matriz, a operação é realizada elemento a elemento. O operador \top indica transposição matricial. Em geral, um processo estocástico associado ao vetor \mathbf{y} é dito ser descorrelacionado se a matriz de covariância é da forma

$$\mathbf{R}_y = \begin{bmatrix} \lambda_0 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{N-1} \end{bmatrix}, \quad (2.2)$$

em que, da expressão (2.1),

$$\lambda_i = \text{E} \left[(y_i - \text{E}[y_i])^2 \right], \quad i = 0, 1, \dots, N-1. \quad (2.3)$$

Considerando um vetor $\mathbf{x} = [x_0 \ x_1 \ \cdots \ x_{N-1}]^\top$, com matriz de covariância \mathbf{R}_x arbitrária, objetiva-se encontrar uma transformação linear da forma

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} \quad (2.4)$$

que resulte em uma matriz de covariância da forma (2.2). A matriz \mathbf{W} deve apresentar dimensão $N \times N$.

Assumindo que a matriz \mathbf{W} seja ortonormal, isto é, suas linhas e colunas são compostas de vetores unitários e ortogonais entre si, é válida a seguinte relação [60]:

$$\mathbf{W} \cdot \mathbf{W}^\top = \mathbf{W}^\top \cdot \mathbf{W} = \mathbf{I}_N,$$

em que \mathbf{I}_N é a matriz identidade de ordem N . O conceito de ortonormalidade é definido adiante na Seção 3.3. Dessa forma, a expressão que inverte a equação (2.4) é

$$\mathbf{x} = \mathbf{W}^\top \cdot \mathbf{y}.$$

Se o processo associado ao vetor \mathbf{x} tem média zero, então o processo associado ao vetor transformado \mathbf{y} também tem média nula. Esse fato pode ser mostrado utilizando-se a expressão (2.4), impondo a condição $E[\mathbf{x}] = 0$. Segue então que:

$$\begin{aligned} E[\mathbf{y}] &= E[\mathbf{W} \cdot \mathbf{x}] \\ &= \mathbf{W} \cdot E[\mathbf{x}] \\ &= 0. \end{aligned}$$

Assumir tal característica é conveniente para a notação e não acarreta em perda de generalidade. Nesse caso, a expressão (2.1) se reduz a

$$\mathbf{R}_y = E[\mathbf{y} \cdot \mathbf{y}^\top]. \quad (2.5)$$

A matriz diagonal de covariância \mathbf{R}_y pode ser calculada da matriz de covariância \mathbf{R}_x do vetor \mathbf{x} original. Considerando que $E[\mathbf{x}] = 0$ e conseqüentemente sendo válida a expressão (2.5), segue que

$$\begin{aligned} \mathbf{R}_y &= E[\mathbf{y} \cdot \mathbf{y}^\top] \\ &= E[(\mathbf{W} \cdot \mathbf{x})(\mathbf{W} \cdot \mathbf{x})^\top] \\ &= E[\mathbf{W} \cdot (\mathbf{x} \cdot \mathbf{x}^\top) \cdot \mathbf{W}^\top] \\ &= \mathbf{W} \cdot E[\mathbf{x} \cdot \mathbf{x}^\top] \cdot \mathbf{W}^\top \\ &= \mathbf{W} \cdot \mathbf{R}_x \cdot \mathbf{W}^\top, \end{aligned}$$

ou, equivalentemente,

$$\mathbf{W} \cdot \mathbf{R}_x = \mathbf{R}_y \cdot \mathbf{W}. \quad (2.6)$$

Uma vez que a matriz \mathbf{R}_y é arbitrariamente diagonal, a expressão (2.6) pode ser escrita para cada linha \mathbf{w}_i de \mathbf{W} . Então, vem que

$$\mathbf{w}_i \mathbf{R}_x = \lambda_i \mathbf{w}_i. \quad (2.7)$$

Resolver a equação (2.7) corresponde a encontrar os autovetores e autovalores da matriz \mathbf{R}_x [62, p. 436]. Portanto, encontrar a matriz de transformação \mathbf{W} que diagonaliza a matriz de covariância do vetor transformado se resume a encontrar os autovetores e autovalores da matriz de covariância do vetor de entrada.

2.1.1 SOLUÇÃO MARKOV-1

Devido à dependência da matriz de transformação com os dados de entrada, não existe uma fórmula única para o cálculo dos vetores de base da KLT. Entretanto, quando se considera determinada classe de sinais, é possível derivar expressões matemáticas únicas para a KLT. Sinais que podem ser modelados como processo markoviano estacionário de primeira ordem (Markov-1) [4] apresentam a matriz de covariância de acordo com

$$[\mathbf{R}_x]_{m,n} = \rho^{|m-n|}, \quad (2.8)$$

em que ρ é o coeficiente de correlação [63]. Para sinais desta forma, a n -ésima componente do k -ésimo autovetor é dada por [4]

$$\mathbf{w}_k(n) = \sqrt{\frac{2}{N + \lambda_k}} \sin \left\{ \mu_k \left[n + 1 - \frac{N + 1}{2} \right] + \frac{\pi(k + 1)}{2} \right\}, \quad (2.9)$$

em que

$$\lambda_k = \frac{1 - \rho^2}{1 - 2\rho \cos \mu_k + \rho^2} \quad (2.10)$$

são os autovalores. Os valores μ_k são as raízes positivas reais da equação transcendental

$$\tan(N\mu) = -\frac{(1 - \rho^2) \sin \mu}{(1 + \rho^2) \cos \mu - 2\rho}. \quad (2.11)$$

2.2 DEFINIÇÕES DA TRANSFORMADA DISCRETA DO COSSENO

Ao introduzir a DCT, Ahmed *et al.* tinham como propósito propor uma nova ferramenta para substituir a KLT que possuísse núcleo de transformação fixo e pudesse, dessa forma, ser computada eficientemente [2]. Entretanto, a nova transformada apresentada se tratava apenas de um tipo de DCT.

Uma DCT de um vetor $x[n]$ de comprimento N é uma transformada do tipo

$$X[k] = \sum_{n=0}^{N-1} x[n] \phi_k^*[n],$$

em que as funções ortogonais $\phi_k^*[n]$ que compõem o núcleo da transformação são funções cosseno. A DCT também pode ser expressa na forma matricial

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

em que \mathbf{C} é a matriz de transformação de dimensão $N \times N$, cujas entradas $c_{k,n}$ são definidas pelos vetores de base $\phi_k^*[n]$. Neste ponto, convém indexar a matriz \mathbf{C} com $k, n = 0, 1, \dots, N - 1$. Os vetores $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_{N-1}]^T$ e $\mathbf{X} = [X_0 \ X_1 \ \dots \ X_{N-1}]^T$ são os vetores de entrada e saída, respectivamente.

A DCT é desenvolvida matematicamente tomando uma sequência $x[n]$ de comprimento N e formando uma nova sequência $y[n]$ de comprimento $M > N$ com simetria par em relação a origem $n = 0$, de modo que a sequência original possa ser unicamente recuperada. A DFT $Y[k]$ de comprimento M da nova sequência $y[n]$ é então considerada [22]. A DCT de comprimento N dada por $X[k]$ é definida a partir de $Y[k]$. Tal procedimento é sumarizado a seguir:

$$\begin{array}{ccccccc} \text{Comprimento } N & & \text{Comprimento } M & M\text{-DFT} & \text{Comprimento } M & & \text{Comprimento } N \\ x[n] & \longleftrightarrow & y[n] & \longleftrightarrow & Y[k] & \longleftrightarrow & X[k] \end{array}$$

O tipo de simetria adotado vai determinar o tipo da DCT. Uma vez que são possíveis oito formas diferentes de realizar uma extensão par em torno da origem da sequência original, há oito tipos diferentes de DCT. Os tipos mais relevantes são a DCT tipo I (DCT-I), a DCT tipo II (DCT-II), a DCT tipo III (DCT-III) e a DCT tipo IV (DCT-IV) [64]. O ponto de simetria pode ser uma amostra do vetor (*whole-sample symmetry*) ou um ponto hipotético intermediário entre duas amostras (*half-sample symmetry*) [65]. As extensões podem ser interpretadas como somas de cópias deslocadas das sequências $\pm x[n]$ e $\pm x[-n]$. Também são possíveis extensões ímpares em torno da origem, dando origem a oito tipos de DST.

A DCT-II é a forma mais utilizada em aplicações de compressão de dados. Sendo assim, consideremos inicialmente este caso. A extensão do vetor $x[n]$ é feita sem a sobreposição dos elementos mais externos, em que a simetria é entre as amostras $n = N - 1$ e $n = N$ (*half-sample symmetry*). A nova sequência simétrica possui comprimento $M = 2N$ e é dada por

$$y_{\text{II}}[n] = \begin{cases} x[n], & 0 \leq n \leq N - 1, \\ x[2N - n - 1], & N \leq n \leq 2N - 1. \end{cases} \quad (2.12)$$

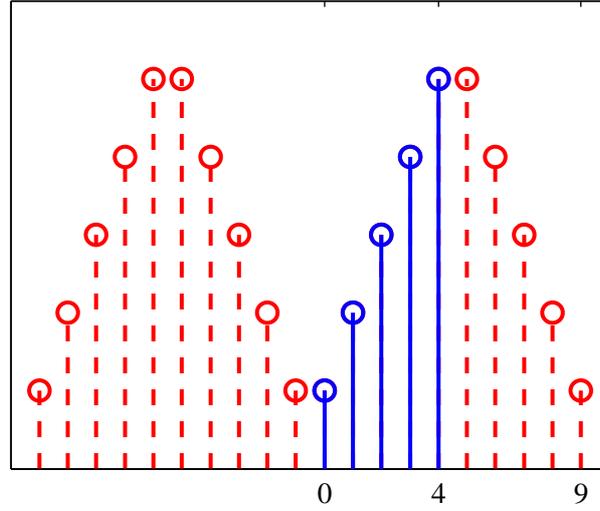


Figura 2.1: Extensão $y_{II}[n]$ para a DCT-II.

Um exemplo de extensão desse tipo para $N = 5$ é mostrado na Figura 2.1.

Tomando a DFT de comprimento M de (2.12) e dividindo o intervalo $0 \leq n \leq 2N - 1$ em dois intervalos $0 \leq n \leq N - 1$ e $N \leq n \leq 2N - 1$, obtém-se

$$\begin{aligned}
 Y[k] &= \sum_{n=0}^{2N-1} y_{II}[n] e^{-j \frac{2\pi kn}{2N}} \\
 &= \sum_{n=0}^{N-1} y_{II}[n] e^{-j \frac{2\pi kn}{2N}} + \sum_{n=N}^{2N-1} y_{II}[n] e^{-j \frac{2\pi kn}{2N}} \\
 &= \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{2N}} + \sum_{n=N}^{2N-1} x[2N - n - 1] e^{-j \frac{2\pi kn}{2N}},
 \end{aligned} \tag{2.13}$$

em que $k = 0, 1, \dots, 2N - 1$. Aplicando ao somatório de intervalo $N \leq n \leq 2N - 1$ uma mudança de variável do tipo $n' = 2N - 1 - n$, a expressão anterior resulta em

$$\begin{aligned}
 Y[k] &= \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{2N}} + \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi k(2N-1-n)}{2N}} \\
 &= \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{2N}} + e^{j \frac{\pi k}{N}} \sum_{n=0}^{N-1} x[n] e^{j \frac{2\pi kn}{2N}} \\
 &= e^{j \frac{\pi k}{2N}} \left(e^{-j \frac{\pi k}{2N}} \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi kn}{2N}} + e^{j \frac{\pi k}{2N}} \sum_{n=0}^{N-1} x[n] e^{j \frac{2\pi kn}{2N}} \right) \\
 &= e^{j \frac{\pi k}{2N}} \sum_{n=0}^{N-1} x[n] \left(e^{-j \frac{\pi k(2n+1)}{2N}} + e^{j \frac{\pi k(2n+1)}{2N}} \right) \\
 &= 2e^{j \frac{\pi k}{2N}} \sum_{n=0}^{N-1} x[n] \cos \frac{\pi k(2n+1)}{2N}, \quad k = 0, 1, \dots, 2N - 1.
 \end{aligned} \tag{2.14}$$

A DCT-II de comprimento N dada por $X_{\text{II}}[k]$ é definida a partir de $Y[k]$ da seguinte forma:

$$X_{\text{II}}[k] = e^{-j\frac{\pi k}{2N}} Y[k], \quad k = 0, 1, \dots, N-1. \quad (2.15)$$

Para obter a expressão para a transformação inversa, um procedimento similar inverso é realizado [22]. Dessa forma, o par de transformadas direta e inversa para a DCT-II pode ser obtido e é dado por

$$X_{\text{II}}[k] = 2 \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad (2.16)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} c_k X_{\text{II}}[k] \cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad (2.17)$$

em que

$$c_k = \begin{cases} \frac{1}{2}, & k = 0, \\ 1, & \text{caso contrário.} \end{cases} \quad (2.18)$$

A DCT-II também é apresentada na literatura em sua forma unitária. Na forma unitária, os vetores da base são normalizados de modo a se tornarem ortonormais (vide Seção 3.3). Os termos 2 , $\frac{2}{N}$ e c_k são distribuídos entre as equações (2.16) e (2.17), resultando nas expressões

$$\hat{X}_{\text{II}}[k] = \sqrt{\frac{2}{N}} \hat{c}_k \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad (2.19)$$

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \hat{c}_k \hat{X}_{\text{II}}[k] \cos\left(\frac{\pi k(2n+1)}{2N}\right), \quad (2.20)$$

em que

$$\hat{c}_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0, \\ 1, & \text{caso contrário.} \end{cases} \quad (2.21)$$

O desenvolvimento da DCT-I é feito de forma similar ao apresentado para a DCT-II. A extensão de $x[n]$ que leva à DCT-I é feita com sobreposição do último elemento da sequência, produzindo uma nova sequência de comprimento $M = 2N - 1$ com simetria par em torno do ponto $N - 1$. a nova sequência é expressa por [64]

$$y_i[n] = \begin{cases} x[n], & 0 \leq n < N, \\ x[2N - n - 2], & N \leq n \leq 2N - 2. \end{cases} \quad (2.22)$$

Um exemplo de extensão de tal tipo para $N = 5$ é mostrado na Figura 2.2.

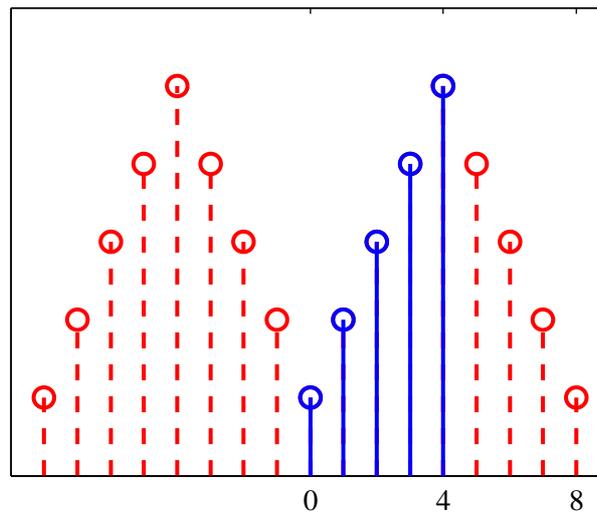


Figura 2.2: Extensão $y_I[n]$ para a DCT-I.

Tomando a DFT de $y_I[n]$ de forma análoga ao caso da DCT-II, obtém-se a definição para a DCT-I. A transformação é dada pelo par de equações de análise e síntese

$$X_I[k] = 2 \sum_{n=0}^{N-1} \alpha_n x[n] \cos\left(\frac{\pi kn}{N-1}\right), \quad (2.23)$$

$$x[n] = \frac{1}{N-1} \sum_{k=0}^{N-1} \alpha_k X_I[k] \cos\left(\frac{\pi kn}{N-1}\right). \quad (2.24)$$

A DCT-I na forma unitária é expressa pelas seguintes expressões de análise e síntese:

$$\hat{X}_I[k] = \sqrt{\frac{2}{N-1}} \sum_{n=0}^{N-1} \hat{\alpha}_n x[n] \cos\left(\frac{\pi kn}{N-1}\right),$$

$$x[n] = \sqrt{\frac{2}{N-1}} \sum_{k=0}^{N-1} \hat{\alpha}_k \hat{X}_I[k] \cos\left(\frac{\pi kn}{N-1}\right),$$

em que

$$\hat{\alpha}_n = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0 \text{ ou } n = N-1, \\ 1, & \text{caso contrário.} \end{cases}$$

Outras formas de extensão com simetria par em torno da origem geram os demais tipos de DCT. As DCT-III e DCT-IV são geradas respectivamente pelas seguintes sequências [65]:

$$y_{III}[n] = \begin{cases} x[n], & 0 \leq n \leq N-1, \\ 0, & n = N, \\ -x[2N-n], & N+1 \leq n \leq 2N. \end{cases} \quad (2.25)$$

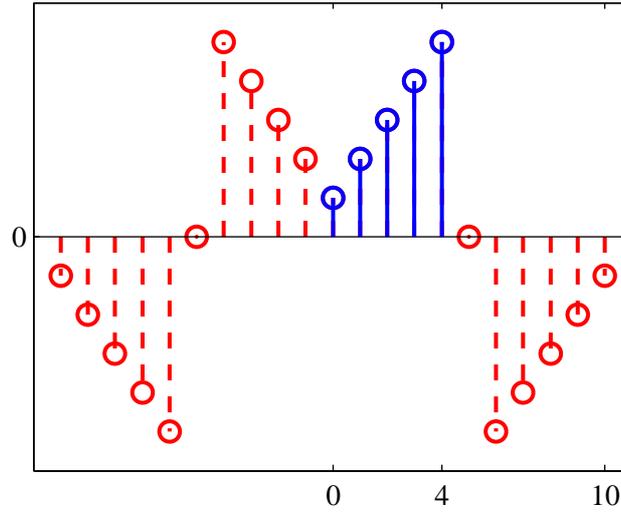


Figura 2.3: Extensão $y_{III}[n]$ para a DCT-III.

e

$$y_{IV}[n] = \begin{cases} x[n], & 0 \leq n \leq N-1, \\ -x[2N-n-1], & N+1 \leq n \leq 2N-1. \end{cases} \quad (2.26)$$

Exemplos de extensão para para a DCT-III e DCT-IV de comprimento 5 são mostrados respectivamente nas Figuras 2.3 e 2.4.

As DCTs tipo I, II, III e IV são definidas formalmente na forma unitária, ou ortonormal, através das matrizes de transformação

$$\begin{aligned} [\mathbf{C}_I^N]_{n,k} &= \sqrt{\frac{2}{N-1}} c_k c_n \cos\left(\frac{\pi k n}{N-1}\right), \\ [\mathbf{C}_{II}^N]_{n,k} &= \sqrt{\frac{2}{N}} c_k \cos\left(\frac{\pi k(2n+1)}{2N}\right), \\ [\mathbf{C}_{III}^N]_{n,k} &= \sqrt{\frac{2}{N}} c_n \cos\left(\frac{\pi(2k+1)n}{2N}\right), \\ [\mathbf{C}_{IV}^N]_{n,k} &= \sqrt{\frac{2}{N}} \cos\left(\frac{\pi(2k+1)(2n+1)}{4N}\right), \end{aligned} \quad (2.27)$$

em que $n, k = 0, 1, \dots, N-1$ e

$$c_n = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0 \text{ ou } n = N, \\ 1, & \text{caso contrário.} \end{cases} \quad (2.28)$$

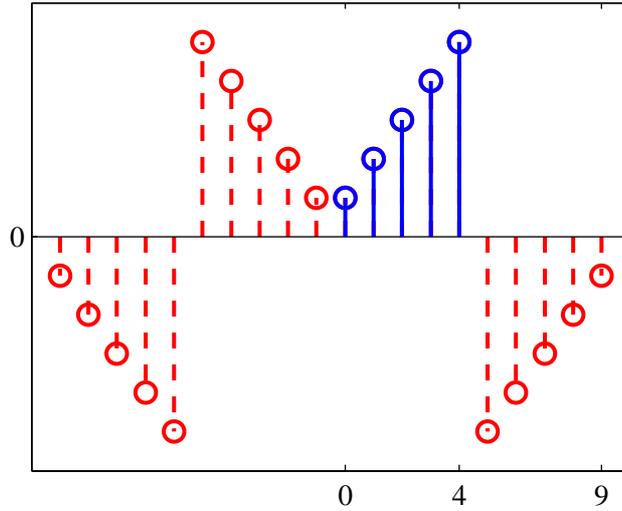


Figura 2.4: Extensão $y_{IV}[n]$ para a DCT-IV.

2.3 A DCT COMO CASO ASSINTÓTICO DA KLT

Consideremos sinais que podem ser modelados como um processo de Markov de primeira ordem com alta correlação. Nesse caso, aplica-se o limite $\rho \rightarrow 1$ às equações (2.8) – (2.11). Da equação (2.11), tem-se

$$\lim_{\rho \rightarrow 1} \tan(N\mu) = - \lim_{\rho \rightarrow 1} \frac{(1 - \rho^2) \sin \mu}{(1 + \rho^2) \cos \mu - 2\rho} = 0. \quad (2.29)$$

Dessa forma, as raízes não negativas de (2.29) são

$$\mu_k = \frac{k\pi}{N}, \quad k = 0, 1, \dots, N - 1. \quad (2.30)$$

Substituindo (2.30) em (2.10) e fazendo $\rho \rightarrow 1$, tem-se que $\lambda_k = 0$ para $k \neq 0$. Para avaliar o caso em que $k = 0$, utiliza-se o operador traço $\text{tr}(\cdot)$. O traço de uma matriz \mathbf{A} é definido por [66, p. 86]

$$\text{tr}(\mathbf{A}) = \sum_{n=1}^N a_{n,n}, \quad (2.31)$$

em que \mathbf{A} é uma matriz de dimensão $N \times N$ e $a_{n,n}$ é o n -ésimo elemento da diagonal principal. Uma vez que $\mathbf{R}_y = \mathbf{W} \cdot \mathbf{R}_x \cdot \mathbf{W}^\top$, as matrizes de covariância são ditas matrizes semelhantes [62, p. 407]. Numa transformação de similaridade deste tipo, o traço da matriz é preservado [62, p. 412]. Considerando um processo de Markov de primeira ordem, segue-se que

$$\text{tr}(\mathbf{R}_x) = \sum_{n=1}^N \rho^{|n-n|} = N = \text{tr}(\mathbf{R}_y) = \sum_{n=0}^{N-1} \lambda_n = \lambda_0.$$

Portanto, $\lambda_0 = N$. Substituindo μ_k e λ_k em (2.9) para encontrar os autovetores, segue-se que

$$\mathbf{w}_0(n) = \frac{1}{\sqrt{N}}$$

e

$$\mathbf{w}_k(n) = \sqrt{\frac{2}{N}} \cos \left\{ \frac{\pi(2n+1)i}{2N} \right\}, \quad k \neq 0.$$

Unificando as duas expressões, tem-se o núcleo da DCT-II, dado por

$$\mathbf{w}_k(n) = c_k \sqrt{\frac{2}{N}} \cos \left\{ \frac{\pi(2n+1)k}{2N} \right\}, \quad k, n = 0, 1, \dots, N-1,$$

em que

$$c_k = \begin{cases} \frac{1}{\sqrt{2}} & k = 0, \\ 1, & \text{caso contrário.} \end{cases}$$

Dessa forma, a DCT-II se comporta como a KLT para sinais perfeitamente correlacionados. Nesse caso, toda a energia do sinal é concentrada no primeiro autovetor, correspondente ao valor médio ou termo DC do sinal. Para sinais altamente correlacionados, a DCT-II é uma excelente aproximação para a KLT. Uma vez que os vetores da base não dependem do sinal de entrada, o que possibilita o desenvolvimento de algoritmos rápidos, a DCT-II é a principal ferramenta para compactação de dados e para descorrelacionar vetores.

2.4 ALGORITMOS RÁPIDOS PARA A DCT

Ao introduzir a DCT, o método proposto por Ahmed *et al.* para computar a DCT se baseava na relação da DCT com a DFT [2]. Entretanto, tal método emprega aritmética complexa em sua implementação. Posteriormente, diversos algoritmos rápidos mais eficientes para a implementação da DCT foram propostos. Tais algoritmos são denominados transformadas rápidas do cosseno (FCTs)

Nesta Seção são apresentados algoritmos rápidos relevantes para a DCT-II. Por ser um método pioneiro e por ser bastante difundido na literatura, sendo inclusive aplicado em aproximações para a DCT baseadas em *lifting scheme* [67], o método de Chen [14] será abordado em maior nível de detalhes. Outros algoritmos importantes também são abordados, sendo sucintamente descritos. Nesta Seção, por simplicidade denota-se a matriz da DCT-II de comprimento N por \mathbf{C}_N , exceto quando necessário distinguir de outro tipo de DCT. Nesse último caso, será utilizada a notação empregada na Seção 2.2. Uma comparação entre as complexidades aritméticas é apresentada ao fim desta Seção. Neste trabalho, considera-se complexidade

aritmética de uma transformada a quantidade de operações matemáticas necessárias para computar a transformação.

2.4.1 ALGORITMO DE CHEN

Apresentado em 1977, o algoritmo por Chen *et al.* apresentou uma melhora de fator seis na complexidade computacional quando comparado com o método baseado na DFT [14]. O algoritmo apresentado envolve apenas operações reais e pode ser aplicado para comprimentos $N = 2^m$, $m \geq 2$. O método é baseado na decomposição da matriz da DCT.

A DCT de comprimento N de um vetor \mathbf{x} pode ser expressa por $\mathbf{X} = \mathbf{C}_N \cdot \mathbf{x}$, onde \mathbf{X} é o vetor de saída. A matriz \mathbf{C}_N pode ser decomposta da seguinte forma recursiva [14]:

$$\mathbf{C}_N = \mathbf{P}_N \cdot \begin{bmatrix} \mathbf{C}_{N/2} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{N/2} \end{bmatrix} \cdot \mathbf{B}_N, \quad (2.32)$$

em que \mathbf{P}_N é uma matriz de permutação que permuta o vetor transformado de uma ordem *bit-reversed* para a ordem natural. A matriz $\mathbf{R}_{N/2}$ é definida de acordo com

$$[\mathbf{R}_{N/2}]_{k,n} = \left[c_k \cos \frac{(2n+1)(2k+1)\pi}{2N} \right], \quad (2.33)$$

$$n, k = 0, 1, \dots, \frac{N}{2} - 1.$$

A matriz \mathbf{B}_N é definida por

$$\mathbf{B}_N = \begin{bmatrix} \mathbf{I}_{N/2} & \bar{\mathbf{I}}_{N/2} \\ \bar{\mathbf{I}}_{N/2} & -\mathbf{I}_{N/2} \end{bmatrix}, \quad (2.34)$$

$$\mathbf{B}_N^* = \begin{bmatrix} -\mathbf{I}_{N/2} & \bar{\mathbf{I}}_{N/2} \\ \bar{\mathbf{I}}_{N/2} & \mathbf{I}_{N/2} \end{bmatrix}.$$

Denota-se por $\bar{\mathbf{I}}_N$ a matriz de ordem N cuja diagonal secundária é composta por elementos unitários e os demais elementos são nulos. A matriz \mathbf{B}_N apresenta estruturas conhecidas como “borboletas” [68]. Uma estrutura borboleta é mostrada na Figura 2.5. Linhas tracejadas representam multiplicação por -1 . De acordo com a Figura 2.5, tem-se que $X_0 = a \cdot x_0 + c \cdot x_1$ e $X_1 = b \cdot x_0 - d \cdot x_1$.

A matriz $\mathbf{R}_{N/2}$ definida em (2.33) é decomposta em $2 \log_2 N - 3$ matrizes da seguinte forma:

$$\mathbf{R}_{N/2} = \mathbf{M}_1 \cdot \mathbf{M}_2 \cdot \mathbf{M}_3 \cdot \mathbf{M}_4 \cdots \mathbf{M}_{2 \log_2 N - 3}. \quad (2.35)$$

a matriz \mathbf{B}_N , que só possui elementos ± 1 contribui com N adições. Então, deriva-se as seguintes equações recursivas:

$$A(\mathbf{C}_N) = N + A(\mathbf{C}_{N/2}) + A(\mathbf{R}_{N/2}), \quad (2.41)$$

$$M(\mathbf{C}_N) = M(\mathbf{C}_{N/2}) + M(\mathbf{R}_{N/2}). \quad (2.42)$$

A expressão para $A(\mathbf{R}_{N/2})$ pode ser inferida notando-se que (2.35) é composta de $\log_2 N - 2$ matrizes cuja diagonal secundária é ocupada pela metade e $\log_2 N - 1$ matrizes completamente ocupadas na diagonal principal e secundária e sempre que há um elemento na diagonal principal e um elemento na diagonal secundária, na mesma linha, há uma soma. Portanto,

$$\begin{aligned} A(\mathbf{R}_{N/2}) &= \frac{N}{4}(\log_2 N - 2) + \frac{N}{2}(\log_2 N - 1) \\ &= \frac{3N}{4} \log_2 N - N, \quad N \geq 4. \end{aligned} \quad (2.43)$$

Em relação ao número de multiplicações $M(\mathbf{R}_{N/2})$, apenas as matrizes de índice ímpares (tipo 1, tipo 2 e tipo 3) contribuem com termos multiplicativos. A primeira matriz, do tipo 1, contribui com N multiplicações. A última matriz, tipo 2, contribui com $N/4$ multiplicações. O restante das $\log_2 N - 3$ matrizes, tipo 3, contribuem com $N/2$. Conseqüentemente,

$$\begin{aligned} M(\mathbf{R}_{N/2}) &= N + \frac{N}{4} + \frac{N}{2}(\log_2 N - 3) \\ &= \frac{N}{2} \log_2 N - \frac{N}{4}, \quad N \geq 8. \end{aligned} \quad (2.44)$$

Uma vez que

$$\mathbf{C}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad (2.45)$$

tem-se que $A(\mathbf{C}_2) = 2$ e $M(\mathbf{C}_2) = 2$. Com esses valores e substituindo (2.44) e (2.43) em (2.41) e (2.42) e solucionando a recursividade, tem-se que [14]

$$A(\mathbf{C}_N) = \frac{3N}{2}(\log_2 N - 1) + 2, \quad (2.46)$$

$$M(\mathbf{C}_N) = N \log_2 N - \frac{3N}{2} + 4, \quad N \geq 4. \quad (2.47)$$

2.4.2 ALGORITMO DE LEE

Em 1984, Lee derivou um algoritmo DIT para a DCT-II [15]. O método foi ilustrado para a IDCT de comprimento N par, mas pode ser aplicado para a DCT.

Desconsiderando as constantes multiplicativas, a expressão (2.17) para a IDCT-II pode ser escrita separando $X_{\text{II}}[k]$ em índices pares e ímpares, da seguinte forma:

$$\begin{aligned} x[n] &= g[n] + \hat{h}[n], \\ x[N-1-n] &= g[n] - \hat{h}[n], \quad k = 0, 1, \dots, N/2-1, \end{aligned}$$

em que

$$g[n] = \sum_{k=0}^{N/2-1} \alpha_k X_{\text{II}}(2k) \cos\left(\frac{(2n+1)k}{N}\right),$$

e

$$\hat{h}[n] = \sum_{k=0}^{N/2-1} \alpha_k X_{\text{II}}(2k+1) \cos\left(\frac{(2n+1)(2k+1)}{2N}\right).$$

Após algumas manipulações algébricas, chega-se às expressões

$$\begin{aligned} x[n] &= g[n] + \frac{1}{2 \cos\left(\frac{2n+1}{2N}\right)} \cdot h[n], \\ x[N-1-n] &= g[n] - \frac{1}{2 \cos\left(\frac{2n+1}{2N}\right)} \cdot h[n], \quad k = 0, 1, \dots, N/2-1, \end{aligned} \tag{2.48}$$

em que

$$h[n] = \sum_{k=0}^{N/2-1} \alpha_k [X_{\text{II}}(2k+1) + X_{\text{II}}(2k-1)] \cos\left(\frac{(2n+1)(2k+1)}{N}\right).$$

Dessa forma, a IDCT de comprimento N é decomposta na soma de duas IDCTs de comprimento $N/2$. O procedimento pode ser aplicado de forma recursiva e pode ser realizado para a DCT. Considerando a DCT de comprimento 8, o método apresenta complexidade aritmética de 12 multiplicações e 29 adições.

2.4.3 ALGORITMO DE WANG

Em 1984, Wang propôs um método sistemático para fatorar as matrizes da DFT, da DCT, da DST e da DWT [16]. Uma forma sistemática para decompor as matrizes das DCTs tipo I, II, III e IV foi apresentada para comprimentos $N = 2^k$, com k inteiro.

Para a DTC-II, a fatoração de Wang é dada por

$$\mathbf{C}_{\text{II}}^N = \mathbf{P}_N \cdot \begin{bmatrix} \mathbf{C}_{\text{II}}^{N/2} & \\ & \bar{\mathbf{C}}_{\text{IV}}^{N/2} \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \cdot \mathbf{B}_N, \tag{2.49}$$

em que

$$\mathbf{P}_N = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 1 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 & \cdots & 0 & 0 \end{bmatrix},$$

\mathbf{B}_N é dado pela Equação (2.34) e $\bar{\mathbf{C}}_{IV}^{N/2} = \bar{\mathbf{I}}_{N/2} \cdot \mathbf{C}_{IV}^{N/2} \cdot \bar{\mathbf{I}}_{N/2}$. Considerando $N = 8$, o algoritmo de Wang apresenta complexidade aritmética de 13 multiplicações e 29 adições.

Wang também mostrou que a decomposição das matrizes de todas as versões da DCT, da DST e da DWT, assim como da DFT, dependem apenas da decomposição da matriz \mathbf{C}_{IV}^N .

2.4.4 ALGORITMO DE VETTERLI–NUSSBAUMER

O algoritmo proposto por Vetterli e Nussbaumer [17] para a DCT-II propõe computar a DCT de comprimento N através de duas DFTs reais de comprimento N . As seguintes definições são propostas:

i) N -DFT cosseno:

$$\text{cos-DFT}[k] = \sum_{n=0}^{N-1} x[n] \cos\left(\frac{2\pi nk}{N}\right), \quad k = 0, 1, \dots, N-1.$$

ii) N -DFT seno:

$$\text{sin-DFT}[k] = \sum_{n=0}^{N-1} x[n] \sin\left(\frac{2\pi nk}{N}\right), \quad k = 0, 1, \dots, N-1.$$

Através do mapeamento

$$\begin{aligned} \hat{x}[n] &= x[2n], \\ \hat{x}[N-n-1] &= x[2n+1], \end{aligned} \tag{2.50}$$

e após algumas manipulações algébricas, a DCT-II pode ser computada através da expressão

$$X_{II}[k] = \cos\left(\frac{2\pi k}{4N}\right) \cdot \text{cos-DFT}[k] - \sin\left(\frac{2\pi k}{4N}\right) \cdot \text{sin-DFT}[k]. \tag{2.51}$$

De forma análoga, expressões similares são derivadas para $\cos\text{-DFT}[k]$ e $\sin\text{-DFT}[k]$. A N -DFT cosseno pode ser computada através da soma de uma $N/2$ -DFT cosseno e uma $N/4$ -DCT, assim como a N -DFT seno pode ser computada através da soma de uma $N/2$ -DFT seno e uma $N/4$ -DCT. O algoritmo pode ser aplicado de forma recursiva. A complexidade aritmética para a DCT de comprimento 8 é dada por 12 multiplicações e 29 adições.

2.4.5 ALGORITMO DE SUEHIRO–HATORI

Suehiro e Hatori apresentaram em 1984 um algoritmo rápido para a DCT-II baseado na fatoração da matriz da DCT-IV [18]. Como mencionado anteriormente e explicitado na Equação (2.49), a decomposição de \mathbf{C}_{IV}^N depende apenas da decomposição de \mathbf{C}_{II}^N .

A DCT-IV é fatorada da seguinte forma:

$$\mathbf{C}_{\text{IV}}^N = \mathbf{Q}_N \cdot \mathbf{V}_N(J) \cdot \mathbf{Y}_N(1) \cdot \mathbf{Y}_N(2) \cdots \mathbf{Y}_N(J-2) \cdot \mathbf{Y}_N(J-1) \cdot \mathbf{H}_N, \quad (2.52)$$

em que $J = \log_2 N$, \mathbf{Q}_N é uma matriz de permutação, que inverte a ordem dos elementos de índice ímpar, dada por

$$\mathbf{Q}_N = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}.$$

$\mathbf{V}_N(J)$ é a matriz dada por $\mathbf{V}_N(J) = \text{diag}(\mathbf{T}_{1/4N} \mathbf{T}_{5/4N}, \mathbf{T}_{9/4N}, \cdots, \mathbf{T}_{(2N-3)/4N})$, em que

$$\mathbf{T}_r = \begin{bmatrix} \cos r\pi & \sin r\pi \\ \sin r\pi & -\cos r\pi \end{bmatrix}.$$

o operador $\text{diag}(\cdot)$ retorna uma matriz bloco-diagonal com os elementos da diagonal principal definidos, em ordem, pelo argumento. A matriz \mathbf{H}_N também é uma matriz de permutação

dada por

$$\mathbf{H}_N = \mathbf{P}_N \cdot \begin{bmatrix} \mathbf{P}_{N/2} & & & \\ & \bar{\mathbf{P}}_{N/2} & & \\ & & \ddots & \\ & & & \mathbf{P}_4 \\ & & & & \bar{\mathbf{P}}_4 \end{bmatrix}, \quad N \geq 4.$$

As matrizes $\mathbf{Y}_N(j)$ são definidas como apresentado a seguir:

i)

$$\mathbf{Y}_N(1) = \mathbf{R}_N(J-1) \cdot \mathbf{X}_N(J-1) \cdot \mathbf{R}_N(J-2) \cdot \mathbf{X}_N(J-2) \cdots \mathbf{R}_N(1) \cdot \mathbf{X}_N(1),$$

em que $\mathbf{R}_N(j) = \text{diag}(\mathbf{I}_{N-2^{j+1}}, \mathbf{U}_{2^{j+1}}(j))$, $j = 1, 2, \dots, J-1$, em que

$$\mathbf{U}_N(j) = \frac{1}{\sqrt{2}} \text{diag} \left(\begin{bmatrix} \mathbf{I}_{2^j} & \mathbf{I}_{2^j} \\ \mathbf{I}_{2^j} & -\mathbf{I}_{2^j} \end{bmatrix}, \begin{bmatrix} \mathbf{I}_{2^j} & \mathbf{I}_{2^j} \\ \mathbf{I}_{2^j} & -\mathbf{I}_{2^j} \end{bmatrix}, \dots, \begin{bmatrix} \mathbf{I}_{2^j} & \mathbf{I}_{2^j} \\ \mathbf{I}_{2^j} & -\mathbf{I}_{2^j} \end{bmatrix} \right),$$

$$\mathbf{X}_N(j) = \text{diag}(\mathbf{I}_{N-2^j}, \mathbf{E}(j)), \quad j = 1, 2, \dots, J-1, \quad \mathbf{E}(j) = \text{diag}(\mathbf{T}_{1/2^{j+1}}, \mathbf{T}_{5/2^{j+1}}, \dots, \mathbf{T}_{(2^{j+1}-3)/2^{j+1}}).$$

ii) As demais matrizes $\mathbf{Y}_N(j)$ são definidas de acordo com

$$\begin{aligned} \mathbf{Y}_N(2) &= \text{diag}(\mathbf{Y}_{N/2}(1), \mathbf{I}_{N/2}), \\ \mathbf{Y}_N(3) &= \text{diag}(\mathbf{Y}_{N/4}(1), \mathbf{I}_{N/4}, \mathbf{Y}_{N/4}(1), \mathbf{I}_{N/4}), \\ &\vdots \\ \mathbf{Y}_N(J-1) &= \text{diag}(\mathbf{Y}_4(1), \mathbf{I}_4, \dots, \mathbf{Y}_4(1), \mathbf{I}_4). \end{aligned}$$

O algoritmo para a DCT-II derivado a partir de tal fatoração apresenta 13 multiplicações e 29 adições.

2.4.6 ALGORITMO DE HOU

Hou propôs algoritmo rápido para a DCT-II em 1987 [70]. Após aplicar uma permutação na matriz da DCT-II da forma $\hat{\mathbf{C}}_N = \mathbf{P} \cdot \mathbf{C}_N$, com

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix},$$

a matriz $\hat{\mathbf{C}}_N$ pode ser decomposta da seguinte forma:

$$\hat{\mathbf{C}}_N = \begin{bmatrix} \hat{\mathbf{C}}_{N/2} & \hat{\mathbf{C}}_{N/2} \\ \hat{\mathbf{D}}_{N/2} & -\hat{\mathbf{D}}_{N/2} \end{bmatrix}.$$

Após uma sequência de manipulações algébricas, chega-se a expressão

$$\hat{\mathbf{C}}_N = \begin{bmatrix} \hat{\mathbf{C}}_{N/2} & \hat{\mathbf{C}}_{N/2} \\ \mathbf{K}\hat{\mathbf{C}}_{N/2}\mathbf{Q} & -\mathbf{K}\hat{\mathbf{C}}_{N/2}\mathbf{Q} \end{bmatrix},$$

em que $\mathbf{Q} = \text{diag}(\cos(n+1/4)(2\pi/N))$, $\mathbf{K} = \mathbf{R} \cdot \mathbf{L} \cdot \mathbf{R}$. A matriz \mathbf{R} é uma matriz de permutação para ordenação na sequência *bit-reversed* e

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & 0 & \cdots & 0 \\ 1 & -2 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 2 & -2 & \cdots & 2 \end{bmatrix}.$$

Aplicando-se recursivamente a decomposição, chega-se ao algoritmo dizimado na frequência.

A complexidade aritmética para $N = 8$ é dada por 12 multiplicações e 29 adições.

2.4.7 ALGORITMO DE LOEFFLER

Em 1989, Loeffler *et al.* propôs uma classe de algoritmos para a DCT-II de comprimento 8 que atingem o limite multiplicativo teórico, dado por 11 multiplicações. São também necessárias 29 adições [20]. A estrutura do algoritmo é mostrada na Figura 2.6.

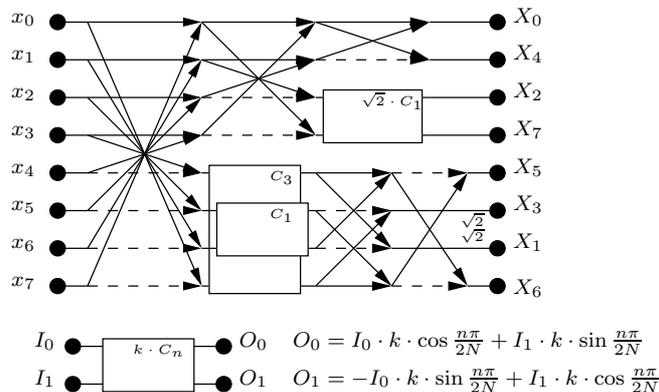


Figura 2.6: Algoritmo rápido para transformada de Walsh-Hadamard

Variações do algoritmo foram apresentadas. O algoritmo de Suehiro, descrito anteriormente, se trata de uma dentre as variações do algoritmo de Loeffler. Uma extensão do algoritmo para o comprimento $N = 16$ também foi proposta [20].

2.4.8 ALGORITMO DE FEIG–WINOGRAD

Em 1992, Feig e Winograd apresentam um algoritmo rápido para a DCT-II [21]. A matriz da DCT-II de comprimento $N = 8$ pode ser escrita de forma parametrizada da seguinte forma:

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

em que $\gamma_k = \cos(2\pi(k+1)/32)$, $k = 0, 1, \dots, 6$.

Feig e Winograd sugeriram uma fatoração para a DCT-II da seguinte forma:

$$\mathbf{C}_8 = \frac{1}{2} \cdot \mathbf{P} \cdot \mathbf{K} \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3, \quad (2.53)$$

em que

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

é uma matriz de permutação e inversão de sinal,

$$\mathbf{K} = \begin{bmatrix} \gamma_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_5 & \gamma_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\gamma_1 & -\gamma_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_4 & -\gamma_6 & \gamma_2 & \gamma_0 \\ 0 & 0 & 0 & 0 & -\gamma_0 & \gamma_4 & -\gamma_6 & \gamma_2 \\ 0 & 0 & 0 & 0 & -\gamma_2 & -\gamma_0 & \gamma_4 & -\gamma_6 \\ 0 & 0 & 0 & 0 & \gamma_6 & -\gamma_2 & -\gamma_0 & \gamma_4 \end{bmatrix},$$

é uma matriz multiplicativa e aditiva e

$$\mathbf{B}_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \text{ e}$$

$$\mathbf{B}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \end{bmatrix},$$

são matrizes aditivas. Tal fatoração proporciona um algoritmo com complexidade aritmética de 22 multiplicações (matriz \mathbf{K}) e 28 adições (matrizes \mathbf{K} , \mathbf{B}_1 , \mathbf{B}_2 e \mathbf{B}_3). Devido à propriedade da separação de núcleo, a 2-D DCT pode ser computada através de $2N$ aplicações da 1-D DCT. Apesar de ser mais custoso computacionalmente para a 1-D DCT de comprimento 8 do que outros algoritmos como o de Loeffler, por exemplo, Feig e Winograd apresentaram um algoritmo diretamente para a 2-D DCT de dimensão 8×8 que se mostrou mais eficiente do que os demais quando computados segundo a filosofia descrita anteriormente.

A 2-D DCT de dimensão 8×8 pode ser descrita como o produto da matriz $\mathbf{C}_{II}^8 \otimes \mathbf{C}_{II}^8$ pelo vetor de entrada de comprimento 64, que representa a matriz 8×8 de entrada, em que o operador \otimes indica produto tensorial [21]. Utilizando a fatoração (2.53) e aplicando o teorema apresentado em [30] para produto tensorial, deriva-se o algoritmo que apresenta 94 multiplicações e 464 adições. A título de comparação, o algoritmo de Loeffler empregando a decomposição linha-coluna [22, p. 163] apresenta 176 multiplicações e 464 adições. Uma comparação completa com todos os algoritmos será apresentada adiante.

2.4.9 COMPARAÇÃO

Na Tabela 2.1, é apresentada a complexidade aritmética de cada algoritmo descrito anteriormente considerando a DCT de comprimento $N = 8$. O número de adições e multiplicações para ambos os casos 1-D e 2-D é apresentado comparativamente.

Como introduzido anteriormente, o limite multiplicativo mínimo estabelecido em [30] é de 11 multiplicações para a DCT de comprimento 8. Este limite é atingido pelo algoritmo de Loeffler, sendo este o mais eficiente segundo o critério de número de multiplicações. Embora o algoritmo de Loeffler seja o mais eficiente para o caso 1-D, ao ser computado a DCT 8×8

Tabela 2.1: *Complexidade aritmética para os algoritmos descritos*

Método	1-D		2-D	
	Mult	Add	Mult	Add
Chen [14]	16	26	256	416
Lee [15]	12	29	192	464
Wang [16]	13	29	208	464
Vetterli [17]	12	29	192	464
Suehiro [18]	13	29	208	464
Hou [19]	12	29	192	464
Loeffler [20]	11	29	176	464
Feig-Winograd [21]	22	28	94	254

de acordo com a decomposição linha-coluna, o método se mostra mais custoso computacionalmente que o algoritmo de de Feig-Winograd, que computa a 2-D DCT diretamente sem utilizar a decomposição linha-coluna. Os menores valores para o caso 1-D e o caso 2-D são destacados em negrito na Tabela 2.1.

CAPÍTULO 3

APROXIMAÇÕES PARA A DCT

3.1 INTRODUÇÃO

O desenvolvimento de transformadas inteiras se tornou um tema relevante em processamento digital de sinais, dada a crescente demanda por tecnologias *wireless* e dispositivos móveis [71]. Tecnologias que são alimentadas por baterias e envolvem processamento de sinais de áudio, imagem estática e dinâmica (vídeo) necessitam de ferramentas com baixo custo computacional e com boa performance [44, 72]. Na prática, esses tipos de sinais são representados através de números inteiros utilizando base binária [4, p. 141]. Transformadas trigonométricas, como a DCT e a DST, mapeiam os vetores de entrada em vetores com elementos em \mathbb{R} [16, 65]. Dessa forma, os sinais precisam ser representados em aritmética de ponto flutuante [4, 73].

Embora algoritmos rápidos como os descritos na Seção 2.4 reduzam drasticamente o número de operações aritméticas, operações com pontos flutuantes permanecem necessárias. Multiplicações com pontos flutuantes são lentas e demandam maior consumo de energia [74]. Em aplicações práticas, não é possível computar a DCT com precisão infinita e erros de truncamento ou arredondamento são inevitáveis para a implementação [4, p. 141]. Além disso, a cota mínima de complexidade multiplicativa já foi atingida [20]. Sendo assim, propor melhorias para os algoritmos de DCT exata é uma tarefa não trivial.

Neste contexto, as aproximações para a DCT permitem reduções consideráveis na complexidade aritmética [32–46]. Transformadas baseadas em aritmética de ponto fixo que representem uma aproximação com números inteiros para as transformadas trigonométricas possibilitam aplicações VLSI de baixa potência [44, 71, 72]. Outra alternativa para a redução do custo computacional é considerar transformadas digitais, isto é, transformadas definidas sobre corpos finitos [75, 76]. Entretanto, este trabalho aborda apenas as transformadas aproximadas. Neste capítulo serão discutidas as aproximações inteiras para a DCT.

3.2 CLASSES DE APROXIMAÇÕES

Uma possibilidade para usar aritmética com números inteiros é substituir os coeficientes da DCT por elementos do conjunto dos números racionais \mathbb{Q} . Multiplicar um valor x por um número racional consiste em uma multiplicação e uma divisão por valores inteiros. Ainda que utilizar aritmética de ponto fixo reduza bastante o custo em termos de *hardware*, multiplicações e divisões por inteiros são operações relativamente custosas [67, 77].

É bem difundido na literatura que, em termos de implementação VLSI, multiplicar ou

dividir valores inteiros por números da forma 2^n , $n \in \mathbb{Z}$, é consideravelmente mais eficiente do que por outros valores inteiros. Isso ocorre porque, uma vez que dispositivos trabalham fisicamente na base binária, realizar tal operação consiste apenas em realizar n deslocamentos de *bits* para a esquerda (multiplicação) ou para a direita (divisão). Por exemplo, consideremos o valor $(14)_{10} = (1110)_2$, em que $(\cdot)_{10}$ e $(\cdot)_2$ representam bases decimal e binária, respectivamente. Realizar a operação $14 \times 2^2 = 56$ corresponde a fazer dois deslocamentos de *bits* para a esquerda, ou seja,

$$(56)_{10} = (111000)_2.$$

Em termos de circuitos digitais, realizar deslocamentos de *bits* consome muito menos energia e componentes de *hardware* do que realizar um algoritmo usual de multiplicação [4,67].

Considerando este fato, uma forma de diminuir o esforço computacional é considerar apenas racionais diádicos como elementos para substituir os coeficientes da DCT. Um racional diádico é um elemento do conjunto \mathbb{Q} escrito na forma $p/2^k$, em que $p, k \in \mathbb{N}$ e $p < 2^k$ [4]. Tais números podem ser representados de acordo com

$$\frac{p}{2^k} = \sum_{i=1}^k c_i \left(\frac{1}{2^i} \right), \quad c_i = 0 \text{ ou } c_i = 1. \quad (3.1)$$

Por exemplo, o racional diádico $111/128$ pode ser escrito como

$$\begin{aligned} \frac{111}{128} &= \frac{64}{128} + \frac{32}{128} + \frac{8}{128} + \frac{4}{128} + \frac{2}{128} + \frac{1}{128} \\ &= \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^7} + \frac{1}{2^8}. \end{aligned}$$

É válido ressaltar que a representação mostrada anteriormente não é única. Alternativamente, a representação para racionais diádicos pode ser realizada considerando o sistema numérico *canonical-signed-digit* (CSD) [78]. Tal representação é da forma (3.1) com $c_i = 0$ ou $c_i = \pm 1$. Em certas situações, considerar CSD pode reduzir ainda mais o número de operações. Considerando ainda o racional $111/128$, em CSD tem-se

$$\begin{aligned} \frac{111}{128} &= 1 - \frac{16}{128} - \frac{1}{128} \\ &= 1 - \frac{1}{2^3} - \frac{1}{2^8}. \end{aligned}$$

Em suma, a multiplicação de qualquer inteiro n por um racional diádico da forma (3.1) pode ser realizada com algumas operações de soma e alguns deslocamento de *bits*. Embora algumas aproximações para a DCT baseadas na representação com racionais diádicos tenham sido propostas apresentando desempenho satisfatório, como as que utilizam o *lifting scheme* [67,79,80],

a complexidade aditiva e o número de deslocamento de bits são incrementadas consideravelmente através deste método.

Uma alternativa para reduzir ainda mais a quantidade de operações é considerar apenas números da forma 2^n , $n \in \mathbb{Z}$, como aproximações para os coeficientes da DCT. Dessa forma, apenas deslocamentos de *bits* são necessários para implementar a multiplicação, evitando as adições presentes quando se considera os racionais diádicos. Um subconjunto ainda mais restrito é definido por $\mathbb{P} = \{0, \pm 1, \pm 2, \pm \frac{1}{2}\}$, onde apenas deslocamentos unitários de bits são necessários para implementação de multiplicação. Várias aproximações considerando o conjunto \mathbb{P} serão abordadas subseqüentemente.

3.3 ORTOGONALIDADE E ORTONORMALIDADE

Ortogonalidade é uma propriedade desejável em uma aproximação para a DCT. Uma vez que a inversa de uma matriz ortogonal é a própria transposta, uma matriz de baixa complexidade terá inversa de baixa complexidade inerentemente. Neste trabalho, utilizamos o conceito de ortogonalidade e ortonormalidade.

Definição 3.1

Seja uma matriz quadrada \mathbf{T} . Se o produto $\mathbf{T} \cdot \mathbf{T}^\top$ resulta em uma matriz diagonal, dizemos que \mathbf{T} é ortogonal. Particularmente, se $\mathbf{T} \cdot \mathbf{T}^\top = \mathbf{T}^\top \cdot \mathbf{T} = \mathbf{I}$, dizemos que \mathbf{T} é ortonormal. ■

De acordo com o método de ortogonalização através de decomposição polar [81,82] descrito em [83], ortonormalizar uma transformada ortogonal \mathbf{T} exige a multiplicação por uma matriz diagonal dada por

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

em que $\sqrt{\cdot}$ representa raiz quadrada matricial [84]. Se \mathbf{T} é uma matriz de transformação que representa uma aproximação ortogonal para a DCT, conforme será descrito adiante, então a aproximação ortonormal para a DCT é dada por

$$\hat{\mathbf{C}} = \mathbf{D} \cdot \mathbf{T}. \quad (3.3)$$

A transformada aproximada de um vetor $\mathbf{x} = [x_0 \ x_1 \ \cdots \ x_{N-1}]^\top$ é dada pelo vetor $\mathbf{X} = [X_0 \ X_1 \ \cdots \ X_{N-1}]^\top$, calculado pela expressão

$$\mathbf{X} = \hat{\mathbf{C}} \cdot \mathbf{x} = \mathbf{D} \cdot \mathbf{T} \cdot \mathbf{x}. \quad (3.4)$$

Em algumas aplicações em que o cálculo da transformada é apenas uma etapa de um procedimento maior, a matriz \mathbf{D} pode ser incorporada em outras etapas de tal procedimento. Neste caso, a matriz \mathbf{D} não proporciona um incremento na complexidade computacional. No contexto de compressão de imagens, a matriz \mathbf{D} pode ser embutida na etapa subsequente ao cálculo da transformada, denominada de etapa de quantização [36, 37, 41]. A etapa de quantização é um passo presente nos algoritmos de compressão de imagem [5, 85, 86] e será detalhada adiante no Capítulo 5.

Uma vez que a matriz de transformação é ortonormal, a operação inversa é dada por

$$\mathbf{x} = \hat{\mathbf{C}}^\top \cdot \mathbf{X} = \mathbf{T}^\top \cdot \mathbf{D}^\top \cdot \mathbf{X}. \quad (3.5)$$

Portanto, o cálculo da inversa possui as mesmas propriedades de baixa complexidade aritmética da transformação direta.

No caso da transformação bidimensional de uma matriz \mathbf{A} , as expressões direta e inversa são dadas respectivamente por

$$\mathbf{B} = \hat{\mathbf{C}} \cdot \mathbf{A} \cdot \hat{\mathbf{C}}^\top, \quad (3.6)$$

$$\mathbf{A} = \hat{\mathbf{C}}^\top \cdot \mathbf{B} \cdot \hat{\mathbf{C}}. \quad (3.7)$$

Transformações ortonormais são também denominadas transformações unitárias [66] e apresentam a propriedade de preservar a energia. A energia do sinal $x[n] = x_i, i = 0, 1, \dots, N - 1$ é dada por [64]

$$\mathcal{E}_x = \sum_{i=0}^{N-1} |x_i|^2. \quad (3.8)$$

Para sinais com valores reais de x_i , a Equação (3.8) pode ser escrita na seguinte forma matricial: $\mathcal{E}_x = \mathbf{x}^\top \cdot \mathbf{x}$. Avaliando a energia \mathcal{E}_X do vetor transformado \mathbf{X} e considerando que $\hat{\mathbf{C}}$ é ortonormal, tem-se que

$$\begin{aligned} \mathcal{E}_X &= \mathbf{X}^\top \cdot \mathbf{X} \\ &= (\hat{\mathbf{C}} \cdot \mathbf{x})^\top \cdot (\hat{\mathbf{C}} \cdot \mathbf{x}) \\ &= \mathbf{x}^\top \cdot \hat{\mathbf{C}}^\top \cdot \hat{\mathbf{C}} \cdot \mathbf{x} \\ &= \mathbf{x}^\top \cdot \mathbf{x} \\ &= \mathcal{E}_x. \end{aligned} \quad (3.9)$$

Portanto, a energia do sinal para transformações ortonormais é invariante.

Considerando agora aproximações não ortogonais, a Equação (3.2) não resulta em uma matriz diagonal. Dessa forma, a matriz \mathbf{D} não pode ser facilmente incorporada à etapa de quantização. Entretanto, se a matriz \mathbf{D} apresentar uma quantidade pequena de elementos não nulos fora da diagonal, uma aproximação $\tilde{\mathbf{D}}$ pode ser obtida substituindo os elementos fora da diagonal por zero. Em [32], uma métrica de desvio de diagonalidade foi proposta e utilizada como parâmetro para encontrar novas aproximações para a DCT. Generalizando (3.4), deriva-se a seguinte expressão:

$$\tilde{\mathbf{D}} = \sqrt{[\text{diag}_M(\mathbf{T} \cdot \mathbf{T}^\top)]^{-1}}, \quad (3.10)$$

em que $\text{diag}_M(\cdot)$ retorna uma matriz diagonal cujos elementos são dados pela diagonal principal da matriz passada no argumento. A aproximação não ortogonal para a DCT é dada por

$$\tilde{\mathbf{C}} = \tilde{\mathbf{D}} \cdot \mathbf{T}. \quad (3.11)$$

A transformação inversa é dada por

$$\mathbf{x} = \tilde{\mathbf{C}}^{-1} \cdot \mathbf{X} = \mathbf{T}^{-1} \cdot \tilde{\mathbf{D}}^{-1} \cdot \mathbf{X}. \quad (3.12)$$

Nota-se que não necessariamente a matriz inversa \mathbf{T}^{-1} apresenta baixa complexidade aritmética. Entretanto, é possível encontrar aproximações com tal característica. Nesse caso e também no contexto de compressão de imagens, a matriz \mathbf{D}^{-1} pode ser incorporada à etapa de dequantização, analogamente ao caso da transformação direta. Sendo assim, também não representa um incremento de custo computacional ao cômputo da transformada inversa.

A transformação direta bidimensional para o caso não ortogonal também é dada pela Equação (3.6). A expressão inversa é computada através das matrizes inversas $\hat{\mathbf{C}}^{-1}$ e $(\hat{\mathbf{C}}^\top)^{-1}$, resultando na expressão

$$\mathbf{A} = \tilde{\mathbf{C}}^{-1} \cdot \mathbf{B} \cdot (\tilde{\mathbf{C}}^\top)^{-1}. \quad (3.13)$$

Proposição 3.1

Seja uma matriz arbitrária \mathbf{W} com $\det(\mathbf{W}) \neq 0$, então $(\mathbf{W}^\top)^{-1} = (\mathbf{W}^{-1})^\top$.

Demonstração: Se $\det \mathbf{W} \neq 0$, existe uma matriz inversa \mathbf{W}^{-1} tal que

$$\mathbf{W} \cdot \mathbf{W}^{-1} = \mathbf{I}.$$

Aplicando o operador transposição \top a ambos os lados da expressão anterior e levando em conta que $\mathbf{I} = \mathbf{I}^\top$, temos

$$\begin{aligned} (\mathbf{W} \cdot \mathbf{W}^{-1})^\top &= \mathbf{I}, \\ (\mathbf{W}^{-1})^\top \cdot \mathbf{W}^\top &= \mathbf{I}. \end{aligned}$$

ou

$$(\mathbf{W}^{-1})^\top = (\mathbf{W}^\top)^{-1}. \quad \blacksquare$$

Corolário 3.1

A expressão (3.13) para o cálculo da inversa no caso não ortogonal se torna:

$$\mathbf{A} = \tilde{\mathbf{C}}^{-1} \cdot \mathbf{B} \cdot (\tilde{\mathbf{C}}^{-1})^\top, \quad (3.14)$$

o que permite computar a expressão inversa através da decomposição linha-coluna. \blacksquare

3.4 APROXIMAÇÕES NO CONJUNTO \mathbb{P}

3.4.1 TRANSFORMADA DE WALSH-HADAMARD

A transformada de Walsh-Hadamard (WHT) [87, p. 301] é construída através da expansão de um sinal utilizando as funções de Walsh $\text{wal}(k, t)$ como base. Tais funções retangulares são ortogonais no intervalo $0 \leq t < 1$ e são mostradas na Figura 3.1.

Um sinal de tempo discreto $x(n)$ de comprimento N pode ser expandido em termos das funções de Walsh uniformemente amostradas com período de amostragem $1/N$. A seguinte equação de síntese é obtida:

$$x(n) = \sum_{k=0}^{N-1} X(k) \text{wal}\left(k, \frac{n}{N}\right). \quad (3.15)$$

Os coeficientes $X(k)$ representam a sequência transformada de Walsh. Tais coeficientes são encontradas levando em conta a ortogonalidade das funções de Walsh. Para tais funções, é válida a expressão

$$\sum_{n=0}^{N-1} \text{wal}\left(k, \frac{n}{N}\right) \text{wal}\left(\kappa, \frac{n}{N}\right) = \begin{cases} N, & \text{se } k = \kappa, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.16)$$

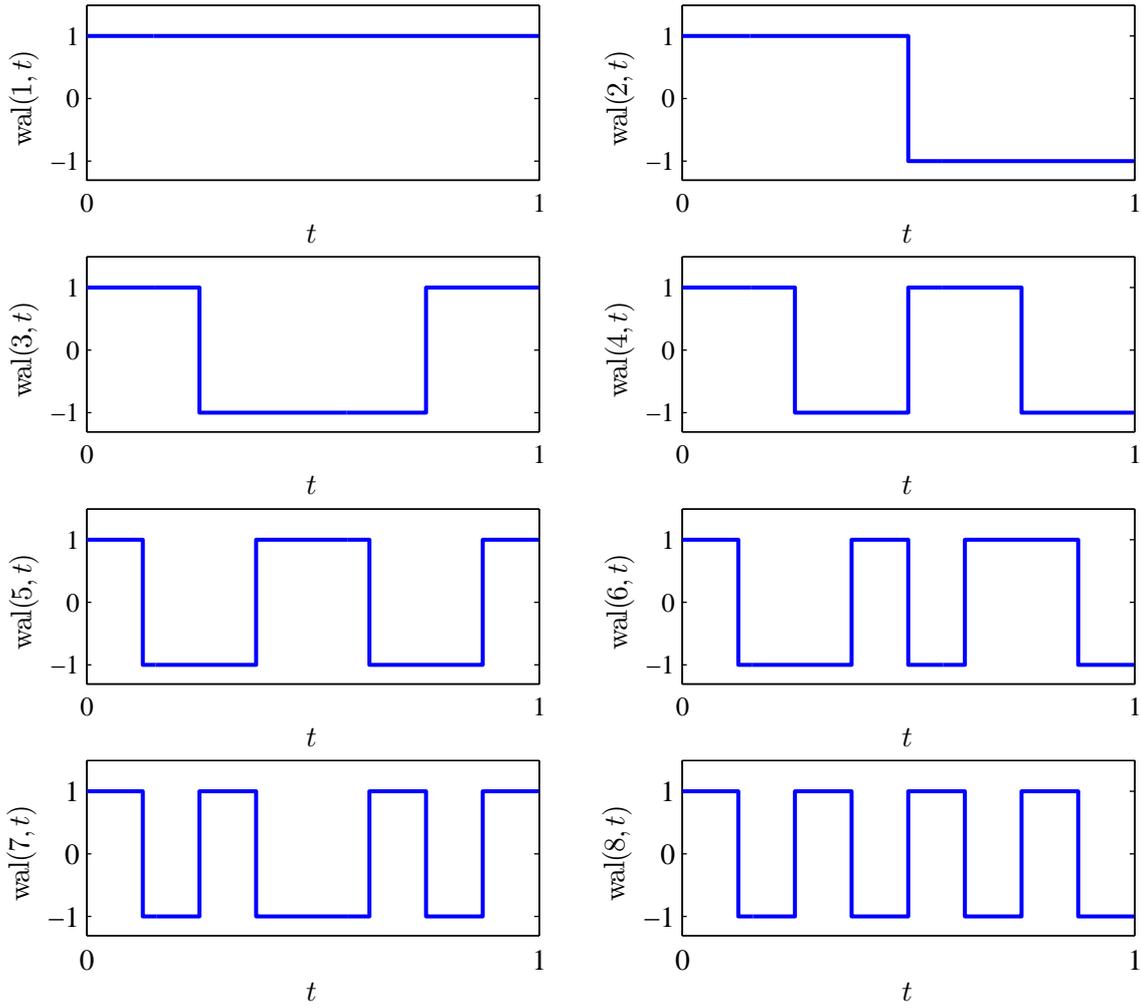


Figura 3.1: Funções de Walsh.

Multiplicando ambos os lados da Equação (3.15) por $\text{wal}(\kappa, \frac{n}{N})$ e somando entre $0 \leq n < N$ e levando em conta a Equação (3.16), deriva-se a expressão de análise para a WHT, dada por

$$X(\kappa) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \text{wal}\left(\kappa, \frac{n}{N}\right). \quad (3.17)$$

A WHT pode ser escrita na forma matricial. Seja o vetor $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^\top$, a sequência de saída $\mathbf{X} = [X_1 \ X_2 \ \cdots \ X_N]^\top$ é dada por

$$\mathbf{X} = \frac{1}{N} \cdot \mathbf{H}_N \cdot \mathbf{x}, \quad (3.18)$$

em que \mathbf{H}_N é a matriz de Walsh-Hadamard, cujas linhas são dadas por amostras igualmente espaçadas com período $1/N$ das funções de Walsh. Para a matriz de ordem N , é válida a expressão

$$\mathbf{H}_N \cdot \mathbf{H}_N^\top = \frac{1}{N} \cdot \mathbf{I}_N. \quad (3.19)$$

Portanto, a transformação inversa é dada por

$$\mathbf{x} = \mathbf{H}_N^T \cdot \mathbf{X}. \quad (3.20)$$

A matriz da WHT de ordem 8 é mostrada a seguir:

$$\mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \quad (3.21)$$

Um algoritmo rápido para computar a transformação direta dada em (3.18) é mostrado na Figura 3.2. Conforme explanado na Seção 2.4.1, as linhas tracejadas representam multiplicação por -1 .

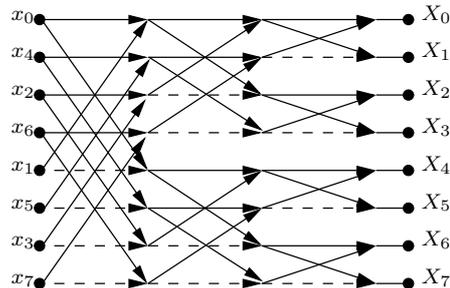


Figura 3.2: Algoritmo rápido para transformada de Walsh-Hadamard.

Através do método de ortonormalização descrito anteriormente, obtêm-se a WHT ortonormal dada por $\hat{\mathbf{C}}_{\text{WHT}} = \mathbf{D}_{\text{WHT}} \cdot \mathbf{H}_8$, com $\mathbf{D}_{\text{WHT}} = \text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}} \right)$.

As funções de Walsh possuem propriedades interessantes, como apresentar apenas elementos ± 1 , o que permite a derivação de algoritmos rápidos apenas com operações de adições. A WHT apresenta aplicações em processamento de imagem, como em codificação de vídeo [88]. Entretanto, essencialmente não foi proposta como uma aproximação para a DCT.

3.4.2 A DCT SINALIZADA

A SDCT [33] foi o método seminal das aproximações da DCT considerando entradas no conjunto \mathbb{P} . A SDCT é proposta aplicando a função $\text{sign}(\cdot)$ aos elementos da matriz da

DCT-II, definida em (2.27). A função sign é definida da seguinte forma:

$$\text{sign}(x) = \begin{cases} 1, & \text{se } x > 0, \\ 0, & \text{se } x = 0, \\ -1, & \text{se } x < 0. \end{cases} \quad (3.22)$$

Se operada em uma matriz, a função $\text{sign}(\cdot)$ é aplicada a cada elemento. Dessa forma,

$$\hat{\mathbf{C}}_{\text{SDCT}} = \frac{1}{\sqrt{N}} \text{sign}(\mathbf{C}_{\text{II}}^N). \quad (3.23)$$

Para $N = 8$, a expressão (3.23) resulta na matriz

$$\mathbf{C}_{\text{SDCT}} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

Como a matriz \mathbf{C}_{SDCT} só possui elementos ± 1 , sua implementação requer apenas operações de adição.

A transformação inversa é dada por

$$\mathbf{C}_{\text{SDCT}}^{-1} = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \\ 1 & 2 & 1 & 0 & -1 & -2 & -1 & 0 \\ 1 & 0 & -1 & -2 & -1 & 0 & 1 & 2 \\ 1 & 0 & -1 & 0 & 1 & 2 & -1 & -2 \\ 1 & 0 & -1 & 0 & 1 & -2 & -1 & 2 \\ 1 & 0 & -1 & 2 & -1 & 0 & 1 & -2 \\ 1 & -2 & 1 & 0 & -1 & 2 & -1 & 0 \\ 1 & -2 & 1 & -2 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

Uma vez que $\mathbf{C}_{\text{SDCT}}^{-1} \neq \mathbf{C}_{\text{SDCT}}^{\top}$, a SDCT não é ortogonal. Entretanto, a inversa apresenta apenas elementos do conjunto \mathbb{P} , com exceção do fator $1/\sqrt{8}$. Como este fator pode ser incluído na etapa inversa da quantização, a transformada inversa pode ser implementada apenas com adições e deslocamento de *bits*.

3.4.3 APROXIMAÇÃO DE LENGWEHASATIT E ORTEGA

Lengwehasatit e Ortega propuseram em [34] uma série de diferentes níveis de aproximações para a DCT. Em particular, a aproximação de nível 1 apresenta entradas apenas com elementos do conjunto \mathbb{P} . Nesse trabalho, esta aproximação será denominada de Lengwehasatit-Ortega DCT (LODCT). A matriz da LODCT é dada por

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

A matriz anterior fornece uma aproximação para a DCT através da expressão

$$\hat{\mathbf{C}}_{\text{LODCT}} = \mathbf{D}_{\text{LODCT}} \cdot \mathbf{T}_{\text{LODCT}},$$

em que

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

3.4.4 SÉRIES BOUGUEZEL-AHMAD-SWAMY

Uma série de aproximações para a DCT de baixa complexidade foram introduzidas por Bouguezel, Ahmad e Swamy (BAS) [35–37, 39, 40, 43]. As matrizes propostas são baseadas na SDCT, onde variações das matrizes são obtidas através de mudanças apropriadas nas entradas. Os novos elementos adicionados fazem parte do conjunto \mathbb{P} , portanto as matrizes permanecem de baixa complexidade. Tais aproximações foram denominadas no presente trabalho como $\mathbf{T}_{\text{BAS-1}}$ [36], $\mathbf{T}_{\text{BAS-2}}$ [37], $\mathbf{T}_{\text{BAS-3}}$ [39], $\mathbf{T}_{\text{BAS-4}}$ [40], $\mathbf{T}_{\text{BAS-5}}$ [40], $\mathbf{T}_{\text{BAS-6}}$ [40], $\mathbf{T}_{\text{BAS-7}}$ [43].

Em [40], foi proposta uma aproximações paramétrica, em que diferentes escolhas de um

único parâmetro geram diferentes métodos. A matriz da transformação é dada por

$$\mathbf{T}(a) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & -0 & 0 & 1 & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \end{bmatrix}.$$

Neste trabalho, definimos $\mathbf{T}_{\text{BAS-4}} = \mathbf{T}(0)$, $\mathbf{T}_{\text{BAS-5}} = \mathbf{T}(1/2)$ e $\mathbf{T}_{\text{BAS-6}} = \mathbf{T}(1)$.

As transformações ortogonais $\mathbf{T}_{\text{BAS-}m}$, $1 \leq m \leq 7$ podem ser ortonormalizadas de acordo com o formalismo descrito anteriormente. Portanto, as aproximações ortonormais para a DCT são dadas por $\hat{\mathbf{C}}_{\text{BAS-}m} = \mathbf{D}_{\text{BAS-}m} \cdot \mathbf{T}_{\text{BAS-}m}$, com $\mathbf{D}_{\text{BAS-}m} = \sqrt{(\mathbf{T}_{\text{BAS-}m} \cdot \mathbf{T}_{\text{BAS-}m}^{\top})^{-1}}$, $1 \leq m \leq 7$. As matrizes $\mathbf{D}_{\text{BAS-}m}$ e $\mathbf{T}_{\text{BAS-}m}$ são mostradas na Tabela 3.1.

3.4.5 A DCT ARREDONDADA E A DCT ARREDONDADA MODIFICADA

Em [41], Cintra e Bayer propuseram uma aproximação para a DCT baseada na função de arredondamento $\text{round}(\cdot)$, da forma como é implementada no ambiente MATLAB [89]. A aproximação é denominada DCT arredondada (RDCT). Aplica-se a função de arredondamento aos elementos da matriz da DCT de acordo com

$$\mathbf{T}_{\text{RDCT}} = \text{round}(2 \cdot \mathbf{C}_{\text{II}}^N). \quad (3.24)$$

Para $N = 8$, deriva-se a seguinte matriz:

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

Tabela 3.1: Aproximações da série BAS

m	$\mathbf{T}_{\text{BAS-}m}$	$\mathbf{D}_{\text{BAS-}m}$	Referência
1	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}} \right)$	[36]
2	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}} \right)$	[37]
3	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 1 & -1 & -2 & 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 1 & -2 & 2 & -1 & 1 & -2 & 2 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{5}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{5}} \right)$	[39]
4	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 71 & -1 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{2} \right)$	($a = 0$) em [40]
5	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{5}} \right)$	($a = \frac{1}{2}$) em [40]
6	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 71 & -1 & 1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{8}} \right)$	($a = 1$) em [40]
7	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}} \right)$	[43]

em que a aproximação ortonormal para DCT é dada por

$$\hat{\mathbf{C}}_{\text{RDCT}} = \mathbf{D}_{\text{RDCT}} \cdot \mathbf{T}_{\text{RDCT}}.$$

A matriz diagonal \mathbf{D}_{RDCT} é dada por

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

Bayer e Cintra introduzem em [42] uma nova aproximação para a DCT baseada na RDCT. Algumas modificações nos elementos da matriz da RDCT foram feitas, derivando então a RDCT modificada (MRDCT), dada pela seguinte matriz:

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

A aproximação ortonormal para a DCT é dada por

$$\hat{\mathbf{C}}_{\text{MRDCT}} = \mathbf{D}_{\text{MRDCT}} \cdot \mathbf{T}_{\text{MRDCT}}.$$

A matriz diagonal é dada por

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

3.4.6 SÉRIE DE APROXIMAÇÕES BASEADAS EM FUNÇÕES INTEIRAS

Em [32], um procedimento sistemático para derivar novas aproximações para a DCT de comprimento $N = 8$ foi proposto. O método é baseado em diferentes funções de arredondamento associados a um parâmetro utilizado para busca exaustiva. No procedimento de busca, restrições foram impostas, como baixa complexidade, ortogonalidade ou *quasi*-ortogonalidade [32] e, no caso de não ortogonalidade, a matriz inversa possuir baixa complexidade.

O mapeamento é feito da forma $\mathbf{T}_{\text{int}} = \text{int}(\alpha \cdot \mathbf{C}_{\text{II}}^N)$. Funções inteiras $\text{int}(\cdot)$ consideradas são dadas pela funções piso, teto, truncamento e arredondamentos *away-from-zero* (AFZ),

half-up (HU), *half-down*, *half-away-from-zero* (HAFZ), *half-towards-zero* (HTZ), *half-to-even* (HE), *half-to-odd* (HO), definidas respectivamente por

$$\begin{aligned}
\text{floor}(x) &= \lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}, \\
\text{ceil}(x) &= \lceil x \rceil = \min\{m \in \mathbb{Z} \mid m \geq x\}, \\
\text{trunc}(x) &= \text{sign}(x) \cdot \lfloor |x| \rfloor, \\
\text{round}_{\text{AFZ}}(x) &= \text{sign}(x) \cdot \lceil |x| \rceil, \\
\text{round}_{\text{HU}}(x) &= \lfloor x + \frac{1}{2} \rfloor, \\
\text{round}_{\text{HD}}(x) &= \lceil x - \frac{1}{2} \rceil, \\
\text{round}_{\text{HAFZ}}(x) &= \text{sign}(x) \cdot \lfloor |x| + \frac{1}{2} \rfloor, \\
\text{round}_{\text{HRZ}}(x) &= \text{sign}(x) \cdot \lceil |x| - \frac{1}{2} \rceil, \\
\text{round}_{\text{HE}}(x) &= \begin{cases} \lfloor x - \frac{1}{2} \rfloor & \text{se } \frac{2x-1}{4} \in \mathbb{Z}, \\ \lfloor x + \frac{1}{2} \rfloor & \text{caso contrário,} \end{cases} \\
\text{round}_{\text{HO}}(x) &= \begin{cases} \lfloor x + \frac{1}{2} \rfloor & \text{se } \frac{2x-1}{4} \in \mathbb{Z}, \\ \lfloor x - \frac{1}{2} \rfloor & \text{caso contrário.} \end{cases}
\end{aligned}$$

Todas as matrizes derivadas apresentam fatoração similar a do algoritmo de Feig-Winograd, descrito na Seção 2.4. A fatoração é dada pela Equação (2.53), em que as entradas γ_i da matriz multiplicativa \mathbf{K} são substituídas por entradas m_i , $0 \leq i \leq 7$, que dependem da transformação em questão. Sendo assim, uma única estrutura de algoritmo rápido pode ser derivada para todas as aproximações. Tal estrutura é mostrada na Figura A.9 (vide Tabela A.1). Foram apresentadas as novas matrizes ortogonais $\mathbf{T}_{\text{INT-}i}$, $i = 0, 1, 2, 3, 4, 5, 6, 7$. Entretanto, $\mathbf{T}_{\text{INT-}3}$ e $\mathbf{T}_{\text{INT-}7}$ não apresentam apenas elementos no conjunto \mathbb{P} , e não serão consideradas neste trabalho, enquanto $\mathbf{T}_{\text{INT-}0}$ se trata da RDCT, apresentada em [41] e descrita anteriormente. As matrizes de ortonormalização são denotadas por $\mathbf{D}_{\text{INT-}i}$. As aproximações encontradas são apresentadas respectivamente nas Tabela 3.2.

Usando (3.25), o MSE pode ser definido como [4]

$$\begin{aligned}
\epsilon &= \frac{1}{N} \mathbf{E} [\mathbf{e} \cdot \mathbf{e}^\top] \\
&= \frac{1}{N} \mathbf{E} [\mathbf{x}^\top \cdot \mathbf{\Gamma}^\top \cdot \mathbf{\Gamma} \cdot \mathbf{x}] \\
&= \frac{1}{N} \mathbf{E} [\text{tr} (\mathbf{\Gamma} \cdot \mathbf{x} \cdot \mathbf{x}^\top \cdot \mathbf{\Gamma}^\top)] \\
&= \frac{1}{N} \text{tr} (\mathbf{\Gamma} \cdot \mathbf{R}_x \cdot \mathbf{\Gamma}^\top),
\end{aligned}$$

em que \mathbf{R}_x é a matriz de covariância do vetor de entrada \mathbf{x} e $\text{tr}(\cdot)$ representa o traço de uma matriz. Uma boa aproximação tende a apresentar baixos valores para MSE (em geral, da ordem de 10^{-2}).

3.5.2 GANHO DE CODIFICAÇÃO

O ganho de codificação da transformada (*transform coding gain*) C_g mede a eficiência de codificação da transformação ortogonal para aplicações de compressão de dados e, por meio da matriz de covariância \mathbf{R}_y do vetor de saída, é definido como [4]

$$C_g = 10 \log_{10} \frac{\frac{1}{N} \sum_{i=0}^{N-1} \sigma_{x_i}^2}{\left(\prod_{i=0}^{N-1} \sigma_{x_i}^2 \|f_i\|^2 \right)^{\frac{1}{N}}}, \quad (3.26)$$

em que $\sigma_{x_i}^2$ é o i -ésimo elemento diagonal da matriz \mathbf{R}_y e $\|f_i\|$ é a norma do i -ésimo vetor de base da matriz de transformação. Valores altos de ganho de codificação correspondem a transformadas com maior capacidade de concentrar energia em uma menor quantidade de coeficientes. A matriz de covariância \mathbf{R}_y pode ser computada através da matriz de covariância do sinal de entrada \mathbf{R}_x pela Equação (2.6). Neste trabalho consideramos os sinais de entrada modelados como processos de Markov de primeira ordem, sendo válida a Equação (2.8) para o cálculo de \mathbf{R}_x .

Neste trabalho, também são abordadas transformações não ortogonais. Sendo assim, é utilizado o ganho de codificação unificado C_g^* , introduzido em [4]. Dada uma matriz de transformação $\hat{\mathbf{C}}$ de dimensão $N \times N$ e, sendo \mathbf{h}_k e \mathbf{g}_k , para $k = 0, 1, \dots, N - 1$, os vetores linha de $\hat{\mathbf{C}}$ e $\hat{\mathbf{C}}^{-1}$, respectivamente, o ganho de codificação unificado é dado por

$$C_g^* = 10 \log_{10} \left(\prod_{k=0}^{N-1} \frac{1}{(A_k \cdot B_k)^{\frac{1}{N}}} \right), \quad (3.27)$$

em que $A_k = \text{sum} [(\mathbf{h}_k^\top \cdot \mathbf{h}_k) \odot \mathbf{R}_x]$, $\text{sum}(\cdot)$ retorna a soma dos elementos da matriz de seu argumento e o operador \odot representa o produto matricial de Hadamard (produto elemento a

elemento) [90] e $B_k = \|\mathbf{g}_k\|^2$. Para transformações ortogonais, o ganho de codificação C_g e o ganho de codificação unificado C_g^* são equivalentes [4].

3.5.3 EFICIÊNCIA DA TRANSFORMADA

A eficiência da transformada (*transform efficiency*) η é uma métrica que, assim como o ganho de codificação, fornece informação de *performance* em compressão de dados [4]. É definida como

$$\eta = \frac{\sum_{i=0}^{N-1} |r_{ii}|}{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |r_{ij}|} 100\%, \quad (3.28)$$

em que r_{ij} são elementos de \mathbf{R}_y . Esta métrica mede o “poder de descorrelação” da transformada.

3.5.4 AVALIAÇÃO DAS TRANSFORMADAS

Através das figuras de mérito descritas anteriormente, é possível avaliar o desempenho das transformadas utilizadas sem a necessidade de simulações de compressão de imagem. As quantidades para as medidas de desempenho de cada método mencionado são mostradas na Tabela 3.3. A DCT exata foi incluída na tabela para fins de comparação e O MSE foi calculado tomando-a como referência. A aproximação com melhor desempenho segundo tais figuras de mérito é a LODCT. Os valores são destacados em negrito. Entretanto, veremos que o custo aritmético desta transformada é relativamente elevado, com 24 adições. Por outro lado, a MRDCT apresenta medidas de desempenho inferiores, porém apresenta o menor custo aritmético, com 14 adições.

Consideremos o coeficiente de correlação de Pearson [91]. Verifica-se um coeficiente de correlação entre o ganho de codificação unificado e a eficiência da transformada de $\rho = 0.797$. Entretanto, observando a Tabela 3.3, percebe-se que a SDCT representa um *outlier* para o valor de ganho de codificação. O ganho de codificação unificado é uma forma de representar o ganho de codificação para transformações não ortogonais, como a SDCT. Se considerarmos apenas o ganho de codificação entre as transformações ortogonais, nota-se um coeficiente de correlação de $\rho = 0.996$ em relação a eficiência da transformada, corroborando que tais métricas representam características de desempenho de codificação similares. O coeficiente de correlação entre o MSE e a eficiência da transformada é $\rho = -0.447$. O valor negativo é

esperado, uma vez que o MSE diminui para zero e a eficiência da transformada aumenta para 93.99% a medida que a transformação se aproxima da DCT.

Tabela 3.3: *Medidas de desempenho para os métodos abordados*

Método	MSE	C_g^* (dB)	η (%)
DCT	0	8.8259	93.9912
DHT	0.0251	7.9461	85.3138
SDCT [33]	0.0207	6.0261	82.6190
LODCT [34]	0.0061	8.3902	88.7023
BAS-1 [36]	0.0238	8.1194	86.8626
BAS-2 [37]	0.0275	7.9126	85.3799
BAS-3 [39]	0.0210	8.3251	88.2182
BAS-4 [40]	0.0710	7.9118	85.6419
BAS-5 [40]	0.0678	8.1194	86.8626
BAS-6 [40]	0.0710	7.9126	85.3799
BAS-7 [43]	0.0251	7.9461	85.3138
RDCT [41]	0.0098	8.1827	87.4297
MRDCT [42]	0.0594	7.3326	80.8969
INT-1 [32]	0.0375	8.1361	86.8051
INT-2 [32]	0.0100	8.1361	86.8051
INT-4 [32]	0.0098	8.1834	87.1567
INT-5 [32]	0.0100	8.1369	86.5359
INT-6 [32]	0.0062	8.3437	88.0594

CAPÍTULO 4

APROXIMAÇÕES PODADAS

4.1 PRELIMINARES

De maneira geral, podar um algoritmo rápido de uma transformada tem como objetivo diminuir a complexidade aritmética descartando elementos dos vetores da transformação. Dessa forma, as operações envolvendo o elemento descartado não são realizadas. Este procedimento pode ser representado matematicamente através do modelo matricial para uma transformação linear. Seja uma transformação linear $\mathbf{T} : \mathbf{C}^N \rightarrow \mathbf{C}^N$, representada por uma matriz \mathbf{T} genérica dada por

$$\mathbf{T} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,N} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N,1} & t_{N,2} & \cdots & t_{N,N} \end{bmatrix}. \quad (4.1)$$

A transformação linear é expressa matematicamente por

$$\mathbf{X} = \mathbf{T} \cdot \mathbf{x},$$

em que $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_N]^\top$ e $\mathbf{X} = [X_1 \ X_2 \ \cdots \ X_N]^\top$. Neste ponto, é conveniente para a notação indexar os vetores e matrizes com índices $n, k = 1, 2, \dots, N$.

Suponhamos que o termo x_i seja desconsiderado, de modo que o novo vetor de dimensão $N - 1$ é dado por $\hat{\mathbf{x}}_1 = [x_1 \ x_2 \ \cdots \ x_{i-1} \ x_{i+1} \ \cdots \ x_N]^\top$. Consideremos agora uma nova transformação $\hat{\mathbf{T}}_1 : \mathbf{C}^{N-1} \rightarrow \mathbf{C}^N$ dada pela expressão $\hat{\mathbf{X}}_1 = \hat{\mathbf{T}}_1 \cdot \hat{\mathbf{x}}_1$. Se desejarmos manter as mesmas operações dos elementos da matriz \mathbf{T} original pelos elementos restantes do vetor $\hat{\mathbf{x}}_1$, a nova matriz $\hat{\mathbf{T}}_1$ de dimensão $N \times (N - 1)$ corresponde a matriz original \mathbf{T} com eliminação da i -ésima coluna. Dessa forma,

$$\hat{\mathbf{T}}_1 = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,i-1} & t_{1,i+1} & \cdots & t_{1,N} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,i-1} & t_{2,i+1} & \cdots & t_{2,N} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ t_{N,1} & t_{N,2} & \cdots & t_{N,i-1} & t_{N,i+1} & \cdots & t_{N,N} \end{bmatrix}.$$

De forma análoga, se uma componente X_k do vetor de saída \mathbf{X} for desprezada, a nova transformação $\hat{\mathbf{T}}_2 : \mathbf{C}^N \rightarrow \mathbf{C}^{N-1}$ é dada por $\hat{\mathbf{X}}_2 = \hat{\mathbf{T}}_2 \cdot \mathbf{x}$. A matriz $\hat{\mathbf{T}}_2$ de dimensão $(N - 1) \times N$ corresponde a matriz \mathbf{T} com a k -ésima linha eliminada e $\hat{\mathbf{X}}_2 = [X_1 \ X_2 \ \cdots \ X_{k-1} \ X_{k+1} \ \cdots \ X_{N-1}]^\top$.

Dessa forma,

$$\hat{\mathbf{T}}_2 = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,N} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ t_{k-1,1} & t_{k-1,2} & \cdots & t_{k-1,N} \\ t_{k+1,1} & t_{k+1,2} & \cdots & t_{k+1,N} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N,1} & t_{N,2} & \cdots & t_{N,N} \end{bmatrix}.$$

Este procedimento pode ser generalizado como descrito adiante.

Definição 4.1

Seja uma matriz de transformação $\mathbf{T} : \mathbb{C}^N \rightarrow \mathbb{C}^N$ e os vetores \mathbf{x} e \mathbf{X} , tal que $\mathbf{X} = \mathbf{T} \cdot \mathbf{x}$. Seja um conjunto de M elementos desprezados do vetor de entrada \mathbf{x} , dado por $\{x_{i_1}, x_{i_2}, \dots, x_{i_M}\}$, com $M < N$, e um conjunto de L elementos desprezados do vetor de saída \mathbf{X} , dado por $\{X_{k_1}, X_{k_2}, \dots, X_{k_L}\}$, com $L < N$. Sejam os novos vetores $\hat{\mathbf{x}}$ e $\hat{\mathbf{X}}$ obtidos ao excluir tais elementos. Então, define-se a matriz de transformação podada $\hat{\mathbf{T}} : \mathbb{C}^{N-M} \rightarrow \mathbb{C}^{N-L}$. A matriz $\hat{\mathbf{T}}$ é uma submatriz da matriz \mathbf{T} em que as linhas i_1, i_2, \dots, i_M e colunas k_1, k_2, \dots, k_L de \mathbf{T} são excluídas. A nova transformação é dada por $\hat{\mathbf{X}} = \hat{\mathbf{T}} \cdot \hat{\mathbf{x}}$. ■

Exemplo 4.1

Para exemplificar, consideremos a transformada de Walsh-Hadamard na ordem natural [87, p. 313], cuja expressão matemática é dada por

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}. \quad (4.2)$$

Tal transformada é uma versão permutada da WHT apresentada na Seção 3.4 e será considerada apenas como exemplo.

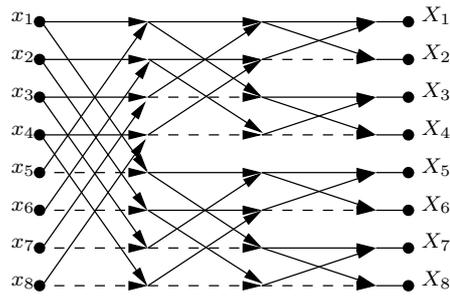


Figura 4.1: Algoritmo rápido para transformada de Walsh-Hadamard na ordem natural.

Um algoritmo rápido para o cálculo da expressão (4.2) é mostrado na Figura 4.1. O algoritmo necessita de 24 adições. Se para alguma determinada aplicação, as componentes X_3 e X_5 do vetor \mathbf{X} e x_6 do vetor \mathbf{x} podem ser desprezadas, uma nova transformação podada pode ser proposta de acordo com a teoria descrita anteriormente. Neste caso, remove-se da matriz de transformação as terceira e quinta linhas (relacionadas a X_3 e X_5) e a sexta coluna (relacionada a x_6). Sendo assim, a nova transformação é dada por

$$\begin{bmatrix} \hat{X}_1 \\ \hat{X}_2 \\ \hat{X}_4 \\ \hat{X}_6 \\ \hat{X}_7 \\ \hat{X}_8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_7 \\ x_8 \end{bmatrix}. \quad (4.3)$$

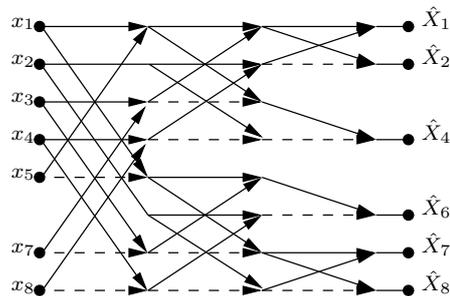


Figura 4.2: Algoritmo rápido para exemplo de transformada de Walsh-Hadamard podada.

Uma maneira de obter um algoritmo rápido para a versão podada é eliminar as linhas de fluxo que se tornam desnecessárias após a eliminação das componentes. O termo “podar” surge a partir deste procedimento. Sendo assim, um algoritmo rápido para a

versão podada da transformada de Walsh-Hadamard (4.3) é mostrado na Figura 4.2 e requer 20 adições. ■

Exemplo 4.2

Algoritmos que computam uma componente única da DFT [92–94] podem ser entendidos como casos particulares da abordagem descrita nesse capítulo. Assim como no exemplo anterior, esse exemplo não propõe um novo algoritmo para computar uma componente única da DFT, mas inclui estes tipos de algoritmo no contexto do método abordado neste trabalho. A componente $V[k]$ da DFT de um sinal $v[n]$ é calculada em sua forma direta pela expressão

$$V[k] = \sum_{n=0}^{N-1} v[n] \cdot \mathbf{W}_N^{nk}. \quad (4.4)$$

O algoritmo de componente única mais popular é o algoritmo de Goertzel [64]. Tal algoritmo é feito definindo o polinômio

$$\begin{aligned} p(x) &= (x - \mathbf{W}_N^k)(x - \mathbf{W}_N^{-k}) \\ &= x^2 - 2 \cos\left(\frac{2k\pi}{N}\right)x + 1. \end{aligned} \quad (4.5)$$

Este polinômio é o polinômio mônico de menor grau com coeficientes reais que possui \mathbf{W}_N^k como raiz, ou seja, o polinômio mínimo de \mathbf{W}_N^k sobre os reais. Fazendo

$$g(x) = \sum_{n=0}^{N-1} v[n] \cdot x^n, \quad (4.6)$$

escreve-se

$$g(x) = p(x)Q(x) + r(x), \quad (4.7)$$

em que o quociente $Q(x)$ e o resto $r(x)$ são encontrados por divisão polinomial. Considerando que \mathbf{W}_N^k é raiz de $p(x)$, a componente $V[k]$ é computada fazendo

$$V[k] = g(\mathbf{W}_N^k) = p(\mathbf{W}_N^k)Q(\mathbf{W}_N^k) + r(\mathbf{W}_N^k) = r(\mathbf{W}_N^k). \quad (4.8)$$

A expressão 4.4 pode ser escrita na forma matricial

$$V(k) = \begin{bmatrix} 1 & \mathbf{W}_N^k & \mathbf{W}_N^{2k} & \dots & \mathbf{W}_N^{(N-1)k} \end{bmatrix} \cdot \mathbf{v}. \quad (4.9)$$

A expressão anterior pode ser interpretada com a teoria proposta anteriormente, em que todas as componentes de $V(k)$, exceto pela k -ésima, são descartadas. Sendo assim, a matriz de transformação é feita eliminando todas as linhas, exceto pela k -ésima, resultando na matriz linha de transformação apresentada na expressão (4.9). ■

4.2 TRANSFORMADA DISCRETA DE FOURIER PODADA

Em 1971, Markel propôs a FFT de Cooley-Tukey podada [51], em que uma versão do algoritmo de dizimação na frequência foi introduzida. Nesse contexto, componentes do vetor de entrada que tendem a ser próximas de zero são desprezadas, eliminando assim as operações envolvendo tais componentes. A DFT de um sinal $\mathbf{x} = [x_0 \ x_1 \ \cdots \ x_{N-1}]^\top$ é definida como

$$X_k = \sum_{n=0}^{N-1} x_n \cdot W_N^{nk}, \quad k = 0, 1, \dots, N-1, \quad (4.10)$$

em que $W_N = e^{-j\frac{2\pi}{N}}$.

Como exemplo, é considerado o caso em que o comprimento da transformada é $N = 16$. Consideremos que o sinal \mathbf{x} possui a maior parte de sua energia concentrada nas componentes x_0 e x_1 , com as demais componentes próximas de zero. Então o algoritmo podado é desenvolvido removendo-se as componentes x_2, x_3, \dots, x_{15} do algoritmo original, diminuindo a quantidade de operações.

O algoritmo de Markel pode ser descrito no formalismo matricial proposto. A Equação (4.10), com pode ser escrita para $N = 16$ como

$$\mathbf{X} = \mathbf{W} \cdot \mathbf{x}, \quad (4.11)$$

em que $\mathbf{X} = [X_0 \ X_1 \ \cdots \ X_{15}]^\top$ e

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_{15}^{(1)(2)} & W_{15}^{(1)(3)} & \cdots & W_{15}^{(1)(15)} \\ 1 & W_{15}^{(2)(1)} & W_{15}^{(2)(2)} & \cdots & W_{15}^{(2)(15)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_{15}^{(15)(1)} & W_{15}^{(15)(2)} & \cdots & W_{15}^{(15)(15)} \end{bmatrix}. \quad (4.12)$$

A transformada podada é representada pela equação

$$\hat{\mathbf{X}} = \hat{\mathbf{W}} \cdot \hat{\mathbf{x}},$$

em que $\hat{\mathbf{x}} = [x_0 \ x_1]^\top$ e a nova matriz de transformação é dada por

$$\hat{\mathbf{W}} = \begin{bmatrix} 1 & 1 \\ 1 & W_{15}^{(1)(2)} \\ 1 & W_{15}^{(2)(1)} \\ \vdots & \vdots \\ 1 & W_{15}^{(15)(1)} \end{bmatrix}. \quad (4.13)$$

O vetor $\hat{\mathbf{X}}$ tem comprimento $N = 16$ e nesse contexto representa uma aproximação para o vetor \mathbf{X} .

4.3 TRANSFORMADA DISCRETA DO COSSENO PODADA

Em diversas aplicações, a maior parte da energia de um sinal é concentrada nos coeficientes de menores frequências da DCT. Esta propriedade de compactação de energia sugere podar um algoritmo para a DCT computando apenas tais coeficientes. Apesar de ser uma ideia intuitiva, o primeiro algoritmo podado para a DCT foi proposto por Wang [47] apenas em 1991, vinte anos depois do algoritmo proposto por Markel [51]. O algoritmo rápido podado proposto por Wang consiste em podar o algoritmo rápido para a DCT proposto por ele mesmo [16, 47].

Matematicamente, a DCT podada pode ser descrita segundo o formalismo proposto anteriormente. Consideramos um vetor $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_{N-1}]^\top$. Para computar apenas os K primeiros coeficientes da DCT, a transformação é dada por

$$\mathbf{X} = \mathbf{C}_{\langle K \rangle} \cdot \mathbf{x}, \quad (4.14)$$

em que $\mathbf{C}_{\langle K \rangle}$ é obtida removendo as $N - K$ últimas linhas da matriz \mathbf{C} da equação (2.2). Então, temos que

$$\mathbf{C}_{\langle K \rangle} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{K,1} & c_{K,2} & \cdots & c_{K,N} \end{bmatrix}.$$

Em [48], foi proposto um algoritmo podado para a DCT baseado no tradicional algoritmo de Loeffler. Neste cenário e em outras aplicações envolvendo compressão de imagem, são consideradas DCTs de duas dimensões. A abordagem consiste em podar um algoritmo de modo a obter apenas um subconjunto dos N^2 coeficientes da 2-D DCT. Por computar apenas uma região ou zona da matriz da DCT, o método é frequentemente denominado *zonal* [48, 49].

4.4 RELAÇÃO ENTRE COMPLEXIDADE ARITMÉTICA PARA MÉTODOS PODADOS EM 1-D E 2-D

Definição 4.2

Sejam respectivamente $M(\cdot)$, $A(\cdot)$ e $B(\cdot)$ as complexidade multiplicativa, complexidade aditiva e a complexidade de deslocamentos de bits. Define-se a complexidade aritmética $\mathcal{A}(\cdot)$

como: $\mathcal{A}(\cdot) = M(\cdot) + A(\cdot) + B(\cdot)$. ■

Os desenvolvimentos matemáticos subsequentes relacionados a $\mathcal{A}(\cdot)$ também são válidos para $M(\cdot)$, $A(\cdot)$ e $B(\cdot)$. A complexidade aritmética $\mathcal{A}_{1D}(\mathbf{C}_{\langle K \rangle})$ de uma transformação podada $\mathbf{C}_{\langle K \rangle}$ será denotada simplesmente por $\mathcal{A}_{1D}(K)$.

Uma transformação podada $\mathbf{C}_{\langle K \rangle}$ de duas dimensões aplicada a uma matriz \mathbf{A} de dimensão $N \times N$ é representada pela matriz \mathbf{X} de dimensão $K \times K$ dada por

$$\mathbf{X} = \mathbf{C}_{\langle K \rangle} \cdot \mathbf{A} \cdot \mathbf{C}_{\langle K \rangle}^{\top}. \quad (4.15)$$

Com o propósito de analisar a decomposição linha-coluna para o caso de uma transformação podada, a Equação (4.15) pode ser decomposta da seguinte forma:

$$\begin{aligned} \mathbf{X} &= \mathbf{C}_{\langle K \rangle} \cdot \mathbf{A} \cdot \mathbf{C}_{\langle K \rangle}^{\top} \\ &= \mathbf{C}_{\langle K \rangle} \cdot (\mathbf{C}_{\langle K \rangle} \cdot \mathbf{A}^{\top})^{\top} \\ &= \mathbf{C}_{\langle K \rangle} \cdot \mathbf{B}. \end{aligned} \quad (4.16)$$

Observando a Equação (4.16), nota-se que é possível computar a transformação de duas dimensões computando a transformação de uma dimensão (4.14) de cada coluna da matriz \mathbf{A}^{\top} ou, equivalentemente, de cada linha da matriz \mathbf{A} . Nesse ponto, são necessários N cálculos da transformação de uma dimensão. Em seguida, é computada a DCT de cada coluna da matriz intermediária \mathbf{B} de dimensão $N \times K$. Nesse ponto, são necessários K cálculos da transformação de uma dimensão. Portanto, são computadas N transformações das linhas da matriz \mathbf{A} , em seguida K transformações das colunas da matriz intermediária \mathbf{B} . Fazendo $K = N$ é obtida a decomposição linha-coluna usual não-podada de duas dimensões.

Seja $\mathcal{A}_{1D}(K)$, a complexidade aritmética em função de K de um algoritmo 1-D para a transformação podada $\mathbf{C}_{\langle K \rangle}$. A complexidade aritmética $\mathcal{A}_{2D}(K)$ do algoritmo podado de duas dimensões computado conforme descrito anteriormente é dada, então, por

$$\mathcal{A}_{2D}(K) = (N + K)\mathcal{A}_{1D}(K). \quad (4.17)$$

Definição 4.3

Sejam a complexidade aritmética $\mathcal{A}(K)$ do algoritmo podado e a complexidade aritmética $\mathcal{A}(N) = \mathcal{A}(K)|_{K=N}$ do algoritmo original não podado; define-se a redução de complexidade como

$$\mathcal{R}(K) = \frac{\mathcal{A}(N)}{\mathcal{A}(K)}. \quad \blacksquare$$

Uma vez que o valor $\mathcal{A}(N)$ é constante para um determinado algoritmo para uma transformação de comprimento N , quanto maior o valor $\mathcal{R}(K)$, menor o valor de $\mathcal{A}(K)$ e maior a redução da complexidade ao podar o algoritmo com parâmetro K em relação ao algoritmo não podado. Observa-se também que, como $\mathcal{A}(K) \leq \mathcal{A}(N)$, então $\mathcal{R}(K) \geq 1$. Quanto maior a redução $\mathcal{R}(K)$, mais vantajosa será a transformada podada.

Proposição 4.1

Sejam $\mathcal{R}_{1D}(K)$ a redução de complexidade para uma transformação podada de uma dimensão e $\mathcal{R}_{2D}(K)$ a redução de complexidade para a mesma transformação podada, computada em 2-D de acordo com a decomposição linha-coluna; então $\mathcal{R}_{2D}(K) > \mathcal{R}_{1D}(K)$.

Demonstração: Considerando a decomposição linha-coluna como método de computar a aproximação em 2-D, então é válida a Equação (4.17). Tomando a redução de complexidade em 2-D através da Definição 4.3, tem-se que

$$\begin{aligned} \mathcal{R}_{2D}(K) &= \frac{\mathcal{A}_{2D}(N)}{\mathcal{A}_{2D}(K)} \\ &= \frac{(2N)\mathcal{A}_{1D}(N)}{(K+N)\mathcal{A}_{1D}(K)} \\ &= \frac{2N}{K+N}\mathcal{R}_{1D}(K). \end{aligned}$$

Como $K + N < 2N$, então temos que $\mathcal{R}_{2D}(K) > \mathcal{R}_{1D}(K)$. ■

Em outras palavras, a Proposição 4.1 mostra que a redução da complexidade aritmética quando um algoritmo é podado é ainda maior para o caso de duas dimensões. Este fato representa uma grande vantagem do método para aplicações que envolvem transformadas em 2-D, como codificação de imagem e vídeo.

4.5 APROXIMAÇÕES PODADAS

O mesmo procedimento aplicado à DCT pode ser feito para uma DCT aproximada. Seja uma matriz \mathbf{T} da forma (4.1) de baixa complexidade computacional, que fornece uma aproximação para $\hat{\mathbf{C}} = \mathbf{D} \cdot \mathbf{T}$, em que $\mathbf{D} = \sqrt{[\text{diag}_M(\mathbf{T} \cdot \mathbf{T}^T)]^{-1}}$. Uma aproximação para a DCT em geral também possui propriedade de compactação de energia similar à DCT exata. Isto significa que as aproximações também tendem a concentrar a maior parte da energia e informação útil do sinal nas componentes de baixa frequência do vetor de saída. Sendo assim, não computar as componentes de alta frequência pode gerar resultados proveitosos.

De forma análoga ao caso da DCT, define-se a matriz da transformação podada em que as K componentes de mais baixa frequência são preservadas como

$$\mathbf{T}_{\langle K \rangle} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,N} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ t_{K,1} & t_{K,2} & \cdots & t_{K,N} \end{bmatrix}, \quad (4.18)$$

que fornece uma aproximação podada para a DCT de acordo com $\hat{\mathbf{C}}_{\langle K \rangle} = \mathbf{D}_{[K]} \cdot \mathbf{T}_{\langle K \rangle}$, em que $\mathbf{D}_{[K]}$ é a matriz $K \times K$ que preserva os primeiros K elementos da diagonal da matriz \mathbf{D} . Sendo assim, $\mathbf{D}_{[K]} = \sqrt{\left[\text{diag}_M \left(\mathbf{T}_{\langle K \rangle} \cdot \mathbf{T}_{\langle K \rangle}^\top \right)\right]^{-1}}$. As matrizes $\mathbf{T}_{\langle K \rangle}$ e $\mathbf{D}_{[K]}$ inspiram as Definição 4.4 e 4.5.

Definição 4.4

Seja uma matriz \mathbf{T} de dimensão $N \times N$. Defina-se a matriz $\mathbf{T}_{\langle K \rangle}$ de dimensão $K \times N$, com $K < N$, em que as primeiras K linhas da matriz \mathbf{T} são preservadas. ■

Definição 4.5

Seja uma matriz \mathbf{D} de dimensão $N \times N$. Defina-se a matriz $\mathbf{D}_{[K]}$ de dimensão $K \times K$, com $K < N$, em que as primeiras K linhas e primeiras K colunas da matriz \mathbf{D} são preservadas. ■

Note que tais definições são casos particulares da Definição 4.1.

Os algoritmos rápidos para todos os métodos são apresentados no Apêndice A. As medidas de desempenho apresentadas na Seção 3.5 assumem que a matriz deve ser inversível ou apresentar dimensão $N \times N$, não podendo ser aplicadas aos métodos podados. Entretanto, tais medidas representam uma indicação inicial de quais métodos apresentarão melhores desempenhos quando podados. Os métodos podados podem ser avaliados através de simulações de compressão de imagem, conforme será abordado no Capítulo 5.

Tabela 4.1: Aproximações SDCT podadas

K	$\mathbf{T}_{\langle K \rangle}$	$\mathbf{D}_{[K]}$
1	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$	$\frac{1}{\sqrt{8}}$
2	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$
3	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$
4	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$
5	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$
6	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$
7	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right)$

4.5.1 DCT SINALIZADA PODADA

A matriz original da SDCT é dada por

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix},$$

que representa através de $\hat{\mathbf{C}} = \mathbf{D} \cdot \mathbf{T}$ uma aproximação para a DCT, em que

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

Aplicando o procedimento descrito anteriormente em (4.18) para a matriz da SDCT, obtemos as suas respectivas matrizes podadas. As matrizes $\mathbf{T}_{\langle K \rangle}$ e $\mathbf{D}_{[K]}$ são mostradas na Tabela 4.1.

O caso particular, $K = 1$ para todos os métodos considerados, estão omitidos das figuras a seguir. Tal caso será comentado posteriormente.

Tabela 4.2: Aproximações BAS-1 podadas

K	$\mathbf{T}_{\langle K \rangle}$	$\mathbf{D}_{[K]}$
1	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$	$\frac{1}{\sqrt{8}}$
2	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ $[1\ 1\ 0\ 0\ 0\ 0\ 0\ -1\ -1]$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}\right)$
3	$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}\right)$
4	$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}\right)$
5	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}\right)$
6	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{2}\right)$
7	$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{\sqrt{5}}\right)$

4.5.2 APROXIMAÇÕES BAS PODADAS

As matrizes de baixa complexidade e matrizes de ortonormalização para todos os métodos ortogonais da série BAS são apresentadas na Subseção 3.4.4. As aproximações ortogonais são apresentadas na Tabela 3.1.

As novas aproximações podadas baseadas em cada uns dos métodos BAS são apresentadas nas Tabelas 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8.

4.5.3 APROXIMAÇÃO LENGWEHASATIT-ORTEGA PODADA

A aproximação LODCT é representada pela matriz

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

Tabela 4.9: Aproximações LODCT podadas

K	$\mathbf{T}_{\langle K \rangle}$	$\mathbf{D}_{[K]}$
1	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$	$\frac{1}{\sqrt{8}}$
2	$[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$ $[1\ 1\ 1\ 0\ 0\ -1\ -1\ -1]$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}\right)$
3	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}\right)$
4	$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & -1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{6}}\right)$
5	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}\right)$
6	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}\right)$
7	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & 0 & -1 & -1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 & 1 & -1 \\ \frac{1}{2} & -1 & 1 & -\frac{1}{2} & -\frac{1}{2} & 1 & -1 & \frac{1}{2} & 1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{5}}\right)$

A aproximação implicada pela matriz anterior é dada por $\hat{\mathbf{C}} = \mathbf{D} \cdot \mathbf{T}$, em que

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

As aproximações podadas derivadas da LODCT são mostradas na Tabela 4.9.

4.5.4 DCT ARREDONDADA PODADA

A RDCT é representada pela matriz de baixa complexidade

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix},$$

em que a aproximação para a DCT é dada por $\hat{\mathbf{C}} = \mathbf{D} \cdot \mathbf{T}$. A matriz diagonal \mathbf{D} é dada por

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

As transformadas podadas derivadas da RDCT são mostradas na Tabela 4.10.

Tabela 4.10: *Aproximações RDCT podadas*

K	$\mathbf{T}_{\langle K \rangle}$	$\mathbf{D}_{[K]}$
1	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{\sqrt{8}}$
2	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}\right)$
3	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 1 & 1 & -1 & -1 & 0 & 1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}\right)$
4	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}\right)$
5	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & -1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}\right)$
6	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}\right)$
7	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$	$\text{diag}\left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}\right)$

4.5.5 DCT ARREDONDADA MODIFICADA PODADA

A MRDCT está associada à seguinte matriz de baixa complexidade:

$$\mathbf{T} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

A matriz diagonal é dada por

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

As novas transformadas derivadas da MRDCT são mostradas na Tabela 4.11.

4.5.6 APROXIMAÇÕES BASEADAS EM FUNÇÕES INTEIRAS PODADAS

Em 3.4.6, foram apresentadas algumas aproximações baseadas em funções inteiras propostas em [32]. As matrizes de baixa complexidade e de ortonormalização para os métodos

Tabela 4.11: Aproximações MRDCT podadas

K	$\mathbf{T}_{\langle K \rangle}$	$\mathbf{D}_{[K]}$
1	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{\sqrt{8}}$
2	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}} \right)$
3	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2} \right)$
4	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}} \right)$
5	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}} \right)$
6	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}} \right)$
7	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2} \right)$

originais são mostradas na Tabela 3.2. As aproximações podadas derivadas destes métodos são mostradas nas Tabelas 4.12, 4.13, 4.14, 4.15 e 4.16.

4.5.7 CONSIDERAÇÕES

A Tabela 4.17 mostra a complexidade aditiva para cada método proposto em 1-D e 2-D para cada valor de K . As complexidades em 1-D foram calculadas contando-se as operações para cada algoritmo apresentado no Apêndice A e as complexidades em 2-D foram computadas utilizando-se a Equação (4.17). Em termos de custo computacional, os métodos mais eficientes são baseados na MRDCT [42]. Esse resultado é esperado, uma vez que esta transformação originalmente possui a menor complexidade.

Para o caso $K = 1$, observa-se que todos os métodos possuem a mesma complexidade aditiva. De fato, todos os métodos para este caso são degenerados para a mesma matriz $[1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$. Conseqüentemente, os algoritmos rápidos associados são os mesmos e as complexidades idênticas, com 7 adições e não são apresentados no Apêndice A. Esta matriz computa o termo médio (termo DC) do vetor de entrada. Um bloco de imagem codificado por essa matriz terá apenas o valor médio dos *pixels* armazenados. A imagem recuperada possuirá tais blocos com valores constantes.

Outras equivalências também podem ser observadas para o caso $K = 2$. A BAS-7 e SDCT para este caso possuem a mesma matriz $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \end{bmatrix}$. A RDCT e LODCT

Tabela 4.16: Aproximações INT-6 podadas

K	$\mathbf{T}_{\langle K \rangle}$	$\mathbf{D}_{[K]}$
1	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\frac{1}{\sqrt{8}}$
2	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \end{bmatrix}$	$\text{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}\right)$
3	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{5}}\right)$
4	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{3}}\right)$
5	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \end{bmatrix}$	$\text{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{2}}\right)$
6	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \end{bmatrix}$	$\text{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}\right)$
7	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \end{bmatrix}$	$\text{diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{5}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{2}}, \frac{1}{2\sqrt{3}}, \frac{1}{2\sqrt{5}}\right)$

se reduzem a matriz $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \end{bmatrix}$. Para o mesmo caso, BAS-1 e BAS-2 geram a matriz $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}$. Tais métodos apresentam complexidade iguais entre si, com 14, 12 e 10 adições respectivamente. Outras equivalências podem ser encontradas para valores maiores de K . Para $K = 3$, encontra-se que a SDCT e BAS-7 apresentam as matrizes $\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \end{bmatrix}$. Uma condição necessária para que haja uma equivalência entre matrizes para um determinado valor de $K = \kappa$ é que também haja equivalência para $K < \kappa$. Isso ocorre porque as κ primeiras linhas das matrizes precisam ser idênticas para que haja equivalência. Diversas outras equivalências podem ser encontradas.

4.5.8 CÁLCULO DA TRANSFORMAÇÃO INVERSA

As matrizes podadas $\hat{\mathbf{C}}_{\langle K \rangle}$ não são matrizes quadradas. Para tais tipos de matrizes, pode-se abordar as questões de inversibilidade à esquerda e à direita [66, p. 98] e de inversas generalizadas [66, p. 363]. Uma matriz \mathbf{H} de dimensão $K \times N$ é dita inversível à esquerda se existe uma matriz \mathbf{H}_L $N \times K$ tal que $\mathbf{H}_L \cdot \mathbf{H} = \mathbf{I}_N$. Nesse caso, é equivalente dizer que $\text{rank}(\mathbf{H}) = N$ [66, p. 98], em que $\text{rank}(\cdot)$ retorna o posto (*rank*) do argumento [62, p. 270]. De forma similar, a matriz \mathbf{H} é dita inversível à direita se existe uma matriz \mathbf{H}_D $N \times K$ tal que $\mathbf{H} \cdot \mathbf{H}_D = \mathbf{I}_K$. Nesse caso, é equivalente dizer que $\text{rank}(\mathbf{H}) = K$ [66, p. 98].

A transformação podada direta em 2-D de uma matriz \mathbf{A} é dada por

$$\mathbf{X} = \hat{\mathbf{C}}_{\langle K \rangle} \cdot \mathbf{A} \cdot \hat{\mathbf{C}}_{\langle K \rangle}^{\top}. \quad (4.19)$$

Para inverter a Equação (4.19), seria desejável inversibilidade à esquerda para a matriz $\hat{\mathbf{C}}_{\langle K \rangle}$, ou $\text{rank}(\hat{\mathbf{C}}_{\langle K \rangle}) = N$. Entretanto, como $N - K$ linhas linearmente independentes foram removidas da matriz original $\hat{\mathbf{C}}$ ($\text{rank}(\hat{\mathbf{C}}) = N$), tem-se que $\text{rank}(\hat{\mathbf{C}}_{\langle K \rangle}) = K$. Sendo assim, as matrizes podadas não são inversíveis à esquerda e a expressão (4.19) não é inversível.

Para derivar um método para computar a transformação inversa, vamos considerar inicialmente uma abordagem intuitiva. Levando em conta que a transformação direta dada em (4.19) fornece uma matriz $K \times K$ contendo os K^2 coeficientes de mais baixa frequência do canto superior esquerdo da transformação não podada, que em geral concentram a maior parte da energia, uma aproximação para a transformação inversa pode ser proposta utilizando a matriz inversa original não podada $\hat{\mathbf{C}}^\top$. Tal aproximação é realizada preenchendo a matriz transformada \mathbf{X} de dimensão $K \times K$ com zeros para formar uma matriz de dimensão $N \times N$. Tal forma de computar a inversa não fornece exatamente a matriz original \mathbf{A} , mas sim uma aproximação $\hat{\mathbf{A}}$, uma vez que apenas os K^2 coeficientes que concentram mais energia estão sendo considerados na transformação inversa.

Se a matriz original $\hat{\mathbf{C}}$ é ortogonal e, conseqüentemente, tem $\hat{\mathbf{C}}^\top$ como matriz inversa, a expressão para a transformação inversa sugerida é

$$\mathbf{A} \approx \hat{\mathbf{A}} = \hat{\mathbf{C}}^\top \cdot \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \hat{\mathbf{C}}. \quad (4.20)$$

Observando os elementos das matrizes $\hat{\mathbf{C}}^\top$ e $\hat{\mathbf{C}}$ na Equação (4.20), observa-se que as últimas $N - K$ colunas da matriz $\hat{\mathbf{C}}^\top$ assim como as últimas $N - K$ linhas da matriz $\hat{\mathbf{C}}^\top$ multiplicam elementos nulos. Este fato pode ser visualizado a seguir:

$$\begin{bmatrix} c_{1,1} & c_{2,1} & \cdots & c_{K,1} & c_{(K+1),1} & \cdots & c_{N,1} \\ c_{1,2} & c_{2,2} & \cdots & c_{K,2} & c_{(K+1),2} & \cdots & c_{N,2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ c_{1,N} & c_{2,N} & \cdots & c_{K,N} & c_{(K+1),N} & \cdots & c_{N,N} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{K,1} & c_{K,2} & \cdots & c_{K,N} \\ c_{(K+1),1} & c_{(K+1),2} & \cdots & c_{(K+1),N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N,1} & c_{N,2} & \cdots & c_{N,N} \end{bmatrix}. \quad (4.21)$$

Sendo assim, os termos nulos podem ser removidos das expressões (4.20) e (4.21). Ao remover as últimas $N - K$ linhas da matriz $\hat{\mathbf{C}}$ e últimas $N - K$ colunas da matriz $\hat{\mathbf{C}}^\top$, obtemos as matrizes $\hat{\mathbf{C}}_{\langle K \rangle}$ e $\hat{\mathbf{C}}_{\langle K \rangle}^\top$, respectivamente. Logo, a expressão (4.20) pode ser simplificada para

$$\hat{\mathbf{A}} = \hat{\mathbf{C}}_{\langle K \rangle}^\top \cdot \mathbf{X} \cdot \hat{\mathbf{C}}_{\langle K \rangle}. \quad (4.22)$$

Observa-se que os pares de transformação direta e inversa, mostradas respectivamente nas expressões (4.19) e (4.22), são similares às transformações inversas e diretas não podadas para

transformações ortogonais, como nas expressões (3.6) e (3.7). Os algoritmos rápidos para a transformação inversa nesse caso são dados pela inversão do fluxo de sinais dos algoritmos diretos. Um procedimento similar pode ser derivado para o caso não ortogonal substituindo $\hat{\mathbf{C}}^\top$ por $\hat{\mathbf{C}}^{-1}$ na Equação (4.20).

Agora vamos considerar o uso de uma inversa generalizada para inverter a Equação (4.19). Inversas generalizadas são ferramentas úteis para matrizes que não são inversíveis à esquerda ou à direita ou matrizes singulares.

Consideremos a inversa generalizada de Moore-Penrose [66, p. 363]. Seja a matriz \mathbf{H} de dimensão $N \times M$ não nula. Pelo teorema da decomposição em valores singulares [66, p. 302], existem matrizes ortogonais \mathbf{S}_1 , $N \times N$, e \mathbf{S}_2 , $M \times M$, tais que

$$\mathbf{H} = \mathbf{S}_1 \cdot \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \mathbf{S}_2,$$

em que $\mathbf{B} = \text{diag}(\sigma_1(\mathbf{H}), \dots, \sigma_R(\mathbf{H}))$, $R = \text{rank}(\mathbf{H})$ e $\sigma_1(\mathbf{H}) > \sigma_2(\mathbf{H}) > \dots > \sigma_R(\mathbf{H}) > 0$ são os valores singulares positivos de \mathbf{H} . Então, a matriz inversa generalizada de Moore-Penrose \mathbf{H}^+ é definida por

$$\mathbf{H}^+ = \mathbf{S}_2^\top \cdot \begin{bmatrix} \mathbf{B}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \mathbf{S}_1^\top.$$

A matriz \mathbf{H}^+ satisfaz as seguintes propriedades:

$$\begin{aligned} \mathbf{H} \cdot \mathbf{H}^+ \cdot \mathbf{H} &= \mathbf{H}, \\ \mathbf{H}^+ \cdot \mathbf{H} \cdot \mathbf{H}^+ &= \mathbf{H}^+, \\ (\mathbf{H} \cdot \mathbf{H}^+)^\top &= \mathbf{H} \cdot \mathbf{H}^+, \\ (\mathbf{H}^+ \cdot \mathbf{H})^\top &= \mathbf{H}^+ \cdot \mathbf{H}. \end{aligned}$$

Inspirado no conceito de ortonormalidade definido na Seção 3.3, propõe-se a Definição 4.6 utilizada neste trabalho.

Definição 4.6

Seja uma matriz \mathbf{H} e sua inversa generalizada \mathbf{H}^+ . Dizemos que \mathbf{H} é pseudo-ortonormal se $\mathbf{H}^+ = \mathbf{H}^\top$.

Se uma aproximação podada para a DCT dada por $\hat{\mathbf{C}}_{\langle K \rangle}$ apresentar pseudo-ortonormalidade, sua inversa generalizada é dada por $\hat{\mathbf{C}}_{\langle K \rangle}^\top$ e a expressão inversa utilizando tal matriz coincide com a Equação (4.22), derivada através da aproximação pela transformação inversa não

podada. Na prática, verificou-se computacionalmente que todos os métodos ortonormais abordados geraram métodos podados pseudo-ortonormais para todo K . Sendo assim, a transformação inversa utilizando a inversa generalizada de Moore-Penrose também é computada pela Equação (4.22) e representa uma aproximação para a inversa original não podada.

Tabela 4.17: Complexidade aritmética para as aproximações podadas apresentadas (adições/ deslocamento de bits)

Método 1-D	K							
	1	2	3	4	5	6	7	8
SDCT [33]	7/0	14/0	17/0	19/0	20/0	22/0	23/0	24/0
BAS-1 [36]	7/0	10/0	13/1	14/1	15/1	16/1	17/2	18/2
BAS-2 [37]	7/0	10/0	13/0	14/0	15/0	16/0	17/0	18/0
BAS-3 [39]	7/0	14/0	17/1	20/2	21/2	22/2	23/3	24/4
BAS-4 [40]	7/0	10/0	11/0	12/0	13/0	14/0	15/0	16/0
BAS-5 [40]	7/0	11/0	13/1	14/1	15/1	16/1	17/1	18/2
BAS-6 [40]	7/0	11/0	13/0	14/0	15/0	16/0	17/0	18/0
BAS-7 [43]	7/0	14/0	17/0	20/0	21/0	22/0	23/0	24/0
LODCT [34]	7/0	13/0	16/1	18/1	19/1	21/1	22/2	24/2
RDCT [41]	7/0	12/0	13/0	16/0	17/0	19/0	20/0	22/0
MRDCT [42]	7/0	8/0	9/0	10/0	11/0	12/0	13/0	14/0
INT-1 [32]	7/0	13/0	14/0	16/0	17/0	19/0	20/0	22/0
INT-2 [32]	7/0	13/1	14/1	16/2	17/2	19/3	20/3	22/4
INT-4 [32]	7/0	13/0	16/0	18/0	19/0	21/0	22/0	24/0
INT-5 [32]	7/0	13/1	16/1	18/2	19/2	21/3	22/3	24/4
INT-6 [32]	7/0	13/1	16/2	18/3	19/3	21/4	22/5	24/6
Método 2-D								
SDCT [33]	63/0	140/0	187/0	228/0	260/0	308/0	345/0	384/0
BAS-1 [36]	63/0	100/0	143/11	168/12	195/13	224/14	255/30	288/32
BAS-2 [37]	63/0	100/0	143/0	168/0	195/0	224/0	255/0	288/0
BAS-3 [39]	63/0	140/0	187/11	240/24	273/26	308/28	245/45	384/64
BAS-4 [40]	63/0	100/0	121/0	144/0	169/0	196/0	225/0	256/0
BAS-5 [40]	63/0	110/0	143/11	168/12	195/13	224/14	255/15	288/32
BAS-6 [40]	63/0	110/0	143/0	168/0	195/0	224/0	255/0	288/0
BAS-7 [43]	63/0	140/0	187/0	240/0	273/0	308/0	345/0	384/0
LODCT [34]	63/0	130/0	176/11	216/12	247/13	294/14	330/30	384/32
RDCT [41]	63/0	120/0	143/0	192/0	221/0	266/0	300/0	352/0
MRDCT [42]	63/0	80/0	99/0	120/0	143/0	168/0	195/0	224/0
INT-1 [32]	63/0	130/0	154/0	192/0	221/0	266/0	300/0	352/0
INT-2 [32]	63/0	130/10	154/11	192/24	221/26	266/42	300/45	352/64
INT-4 [32]	63/0	130/0	176/0	216/0	247/0	294/0	330/0	384/0
INT-5 [32]	63/0	130/10	176/11	216/24	247/26	294/42	330/45	384/64
INT-6 [32]	63/0	130/10	176/22	216/36	247/39	294/56	330/75	384/96

CAPÍTULO 5

APLICAÇÃO EM COMPRESSÃO DE IMAGENS: SIMULAÇÕES E RESULTADOS

5.1 INTRODUÇÃO

Os objetos são percebidos pelo olho humano através de estímulos luminosos. A luz representa um intervalo contínuo no vasto espectro eletromagnético. A percepção humana da luz é geralmente descrita em termos de *brilho*, *matiz* e *saturação*. O brilho está relacionado ao quão brilhante a luz aparenta, a matiz está relacionada às cores e a saturação está relacionada a quão viva ou opaca é a imagem. Tais grandezas são de natureza perceptual e dependem de diversos fatores, incluindo o ambiente e o histórico de estímulo luminoso no olho que observa a luz [22].

Um feixe de luz que contém apenas um comprimento de onda é chamado de *monocromático*. Quando dois feixes monocromáticos são combinados, novas cores podem ser geradas. Uma vasta quantidade de cores pode ser gerada combinando apropriadamente as componentes vermelha, verde e azul (RGB).

Uma imagem colorida pode ser vista como a combinação de três imagens monocromáticas, em que cada imagem contém as componentes RGB. Cada uma dessas imagens monocromáticas é representada como uma matriz de *pixels*, em que cada elemento da matriz carrega um valor inteiro associado à intensidade luminosa da cor associada à matriz (R, G ou B) para aquele ponto da imagem. Através da representação usual de 8 *bits*, esses valores variam entre 0 a 255.

O processamento de uma imagem digital colorida pode ser feito processando separadamente cada uma dessas matrizes (sinais discretos bidimensionais). Este procedimento é simples e frequentemente utilizado. Entretanto, uma vez que o brilho, a matiz e a saturação dependem de cada uma das três componentes, efeitos indesejados podem ser obtidos através dessa forma de processamento. É possível realizar uma mudança da base RGB para uma base denominada luminância-crominância, usada por exemplo no sistema de cor NTSC [22, p. 422]. A conversão é feita através da seguinte transformação:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}.$$

As componentes I e Q são chamadas de crominância e estão relacionadas com a matiz e a saturação da imagem. A componente Y é chamada de luminância e está essencialmente associada ao brilho da imagem.

Imagens em preto e branco (escala de cinza) são imagens monocromáticas e podem, por-

tanto, ser representadas por uma única matriz de *pixels*. A luminância Y pode ser utilizada como a imagem em preto e branco (escala de cinza). Neste trabalho, foram abordadas imagens estáticas monocromáticas em escala de cinza. Algumas imagens, originalmente coloridas, foram convertidas para escala de cinza através da eliminação da matiz e da saturação, mantendo a luminância.

Considerando uma imagem de dimensão 512×512 , monocromática e com 8 *bits* por *pixel*, equivalente a 1 *byte* por *pixel*, o espaço necessário para armazenar todos os pixels dessa imagem é dado por

$$512 \times 512 \times 1 B \approx 262 kB.$$

Para imagens coloridas, têm-se 262 *kB* para cada uma das três matrizes RGB ou YIQ, totalizando $\approx 786 kB$. Técnicas de compressão são importantes para diversas aplicações que lidam com transmissão e armazenamento de imagem e vídeo. De forma simplificada, comprimir uma imagem digitalizada significa armazená-la de forma mais compacta. Em outras palavras, ao reduzir a quantidade de *bits* empregados para armazenar ou transmitir uma imagem, estamos realizando compressão. Na literatura, são encontrados para designar compressão as expressões *compressão de dados*, *compressão de imagem*, *codificação de fonte*, *compressão de banda*, *codificação de imagem*, entre outros [85].

A compressão de imagem explora o fato de que imagens estáticas ou quadros de um vídeo possuem os *pixels* correlacionados. Este tipo de correlação é designado por correlação espacial. Entre quadros consecutivos de um vídeo, também existe correlação entre os mesmos *pixels* de ambos os quadros, sendo esta chamada de correlação temporal. A eficiência de um sistema de compressão pode ser observada pela diminuição da quantidade de *bits* que necessitam ser transitados. A taxa de *bits* (*bit rate*) é definida como o número de *bits* por tempo ou por amostra. No caso de uma imagem, cada amostra do sinal representa um *pixel*. Então o termo *bits por pixel* também é utilizado.

Diversas aplicações de transmissão de imagem e vídeo se tornam viáveis através da compressão. A Tabela 5.1 mostra alguns exemplos de taxa de *bits* para algumas dessas aplicações para o caso comprimido e não comprimido [85]. Os métodos de compressão podem ser classificados em sem perdas (*lossless*) ou com perdas (*lossy*). Em compressão sem perdas, a imagem recuperada após a compressão é idêntica a imagem original antes da compressão. O processo de compressão nesse caso é reversível. A eficiência da codificação para esta forma de compressão é limitada pela entropia da fonte [95]. Fontes com maior entropia são mais

Tabela 5.1: Aplicações com compressão de vídeo

Aplicação	Taxa de <i>bits</i>		Fator de Redução
	Descomprimido	Comprimido	
Vídeo <i>Slow-motion</i>	5.07 Mbps	8-16 kbps	> 99%
Conferência de Vídeo	30.41 Mbps	64-768 kbps	> 97%
Transferência de arquivo de vídeo	30.41 Mbps	384 kbps	≈ 98%
Vídeo em CD-ROM	60.83 Mbps	1.5 - 4 Mbps	> 93%
Transmissão (<i>Broadcast</i>) de vídeo	248.83 Mbps	3-8 Mbps	> 96%
HDTV	1.33 Gbps	20 Mbps	≈ 98%

difíceis de serem codificadas eficientemente.

Em compressão com perdas, a imagem obtida pelo decodificador, após a compressão, possui certo nível de distorção. Na maioria das aplicações, a imagem recuperada não necessita ser idêntica à imagem original, permitindo quantidade tolerável de perda de informação. Tal processo de compressão é irreversível, não sendo possível recuperar a imagem original. Por permitir descartar informação do sinal, comprimir com perdas em geral resulta em uma maior eficiência de codificação, em termos de taxa de *bits*, do que processos sem perdas. Este trabalho aborda métodos que são aplicados em técnicas de compressão com perdas.

5.2 MÉTRICAS PARA O DESEMPENHO DE COMPRESSÃO

Em compressão com perdas, onde a imagem recuperada não é igual a imagem original, é possível quantificar a degradação da imagem através de diferentes métricas. A medida mais tradicional na comparação de sinais de forma geral é o MSE [60]. De forma geral, o MSE entre o sinal x e o sinal recuperado \hat{x} é definido como

$$\text{MSE} = \text{E} [(x - \hat{x})^2]. \quad (5.1)$$

No contexto de imagens, os sinais x e \hat{x} são representados por matrizes de *pixels* de dimensão $N \times M$, dadas por \mathbf{X} e $\hat{\mathbf{X}}$, respectivamente. Sendo assim, o MSE é calculado através da expressão

$$\text{MSE} = \frac{1}{NM} \sum_{n=1}^N \sum_{m=1}^M (\mathbf{X}_{n,m} - \hat{\mathbf{X}}_{n,m})^2. \quad (5.2)$$

É válido ressaltar que o MSE definido neste capítulo é essencialmente diferente do apresentado na Seção 3.5.1. Enquanto este último representa uma medida de erro entre as matrizes de

transformação, o MSE definido pela Equação (5.2) representa uma medida de erro entre uma imagem de referência e uma imagem distorcida.

Uma outra medida frequentemente utilizada é a relação sinal-ruído (SNR) [85]. A SNR é definida em dB pela razão da energia do sinal de entrada pela energia do ruído. Logo,

$$\text{SNR} = 10 \log \frac{\mathcal{E}_s}{\mathcal{E}_r}, \quad (5.3)$$

em que \mathcal{E}_s é a energia do sinal de entrada e \mathcal{E}_r a energia do ruído presente no sinal. No contexto de processamento de imagens, é comum a utilização do PSNR [60], que consiste em considerar o sinal de entrada como sendo o sinal de máxima energia. Para imagens monocromáticas de 8 *bits*, o valor máximo de um *pixel* é 255. Considerando a energia, é utilizado o valor ao quadrado. A energia do ruído é dada pelo próprio MSE. Portanto,

$$\begin{aligned} \text{PSNR} &= 10 \log \left(\frac{255^2}{\text{MSE}} \right), \\ &= 20 \log (255) - 10 \log (\text{MSE}). \end{aligned} \quad (5.4)$$

As grandezas apresentadas anteriormente são vastamente utilizadas por serem simples de calcular e de interpretar fisicamente. Entretanto, elas não levam em conta a maneira como o sistema visual humano interpreta uma imagem. Em [96], foi proposto o índice de similaridade estrutural (SSIM), que busca representar uma medida de qualidade de imagem baseada na percepção humana.

5.2.1 SIMILARIDADE ESTRUTURAL

Em aplicações em que imagens são observadas por seres humanos, a qualidade da imagem observada é subjetiva. A avaliação qualitativa, entretanto, não é prática para contextos em que é necessária a quantificação da qualidade. Portanto, definir métricas que visam medir a qualidade de uma imagem é importante e vem sendo objeto de estudo nos últimos anos [96].

Medidas como o MSE e o SSIM visam quantificar a qualidade ou degradação da imagem baseado na quantidade de erro introduzida na imagem original. Entretanto, imagens com diferentes fontes de distorção que geram o mesmo valor para MSE e PSNR podem apresentar diferentes qualidades quando percebida pelo olho humano.

O índice de SSIM é baseado na premissa de que o sistema visual humano (HVS) é altamente adaptado a extrair informações estruturais da imagem observada. Portanto, o SSIM visa computar uma medida de variação na informação estrutural da imagem distorcida em relação à imagem original, considerada perfeita ou sem ruído.

A medida de estrutura não deve depender fortemente da iluminação da imagem. O sistema proposto para o cômputo da medida de similaridade é dividido em comparações de três medidas diferentes: luminância, contraste e estrutura. Um ponto crucial é que tais componentes sejam independentes. Logo, uma variação na luminância e/ou contraste não deve interferir na estrutura da imagem, por exemplo.

A luminância de uma imagem é dada pelo valor médio [96]. Para um sinal discreto $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_{N-1}]$, a luminância é dada por

$$\mu_x = \frac{1}{N} \sum_{n=0}^{N-1} x_n. \quad (5.5)$$

A comparação entre a luminância entre os sinais \mathbf{x} e \mathbf{y} é realizada através da função a ser definida $l(\mathbf{x}, \mathbf{y})$. Tal função depende de μ_x e μ_y .

Em seguida, a média (luminância) é removida, resultando no sinal $\mathbf{x} - \mu_x$. Através do desvio padrão, uma medida de contraste é obtida [96]. Na forma discreta, o constraste é dado por

$$\sigma_x = \sqrt{\frac{1}{N-1} \sum_{n=0}^{N-1} (x_n - \mu_x)^2}. \quad (5.6)$$

A comparação de contraste $c(\mathbf{x}, \mathbf{y})$ é feita através de σ_x e σ_y .

Em sequência, o sinal é normalizado pelo próprio desvio padrão, resultando em $(\mathbf{x} - \mu_x)/\sigma_x$. A medida de estrutura $s(\mathbf{x}, \mathbf{y})$ é baseada nos sinais $(\mathbf{x} - \mu_x)/\sigma_x$ e $(\mathbf{y} - \mu_y)/\sigma_y$. Por fim, as três medidas são combinadas através de uma função $f(\cdot)$, resultando na medida de similaridade

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = f(l(\mathbf{x}, \mathbf{y}), c(\mathbf{x}, \mathbf{y}), s(\mathbf{x}, \mathbf{y})). \quad (5.7)$$

Para estabelecer por completo a medida de comparação $S(\mathbf{x}, \mathbf{y})$, é necessário definir as funções $l(\mathbf{x}, \mathbf{y})$, $c(\mathbf{x}, \mathbf{y})$, $s(\mathbf{x}, \mathbf{y})$ e $f(\cdot)$. Para estabelecer tais relações, as seguintes condições são desejadas para a função $S(\mathbf{x}, \mathbf{y})$:

1. Simetria: $\text{SSIM}(\mathbf{x}, \mathbf{y}) = \text{SSIM}(\mathbf{y}, \mathbf{x})$;
2. Intervalo limitado: $\text{SSIM}(\mathbf{x}, \mathbf{y}) \leq 1$;
3. Máximo único: $\text{SSIM}(\mathbf{x}, \mathbf{y}) = 1$ se e somente se $\mathbf{x} = \mathbf{y}$.

A comparação de luminância é definida como:

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \quad (5.8)$$

em que a constante C_1 é um valor pequeno incluído para evitar instabilidade numérica quando $\mu_x^2 + \mu_y^2$ é próximo de zero.

De forma análoga, a comparação de contraste é definida como

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \quad (5.9)$$

em que C_2 também é uma constante pequena incluída para evitar instabilidade numérica. Ambas as definições satisfazem as três propriedades desejadas mencionadas anteriormente.

A comparação de estrutura é realizada com o vetor subtraído da luminância e normalizado pelo contraste. A função $s(\mathbf{x}, \mathbf{y})$ é definida pela covariância entre os vetores. Portanto,

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \quad (5.10)$$

em que

$$\sigma_{xy} = \frac{1}{N-1} \sum_{n=0}^{N-1} (x_n - \mu_x)(y_n - \mu_y) \quad (5.11)$$

e C_3 , de forma similar aos casos anteriores, é uma constante para evitar instabilidade numérica. Também se verifica que a Equação (5.10) satisfaz as três propriedades listadas anteriormente. A função de comparação $f(\mathbf{x}, \mathbf{y})$ é definida então para gerar o índice SSIM de acordo com

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^\alpha \cdot [c(\mathbf{x}, \mathbf{y})]^\beta \cdot [s(\mathbf{x}, \mathbf{y})]^\gamma, \quad (5.12)$$

em que $\alpha > 0$, $\beta > 0$ e $\gamma > 0$ são constantes escolhidas para ponderar a relevância de cada uma das comparações. Por padrão, são escolhido os valores $\alpha = \beta = \gamma = 1$ e $C_3 = C_2/2$ [96]. Então, a Equação (5.12) se torna

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \quad (5.13)$$

5.3 COMPRESSÃO DE IMAGEM ESTÁTICA

Em codificação de imagem baseada em transformadas, uma imagem de dimensão $M \times M$ é subdividida em imagens menores (blocos), de dimensão $N \times N$, com $N < M$. Tal método pode ser chamado de codificação por transformada de bloco. Cada bloco, representado por uma matriz \mathbf{B} , é então submetido a uma transformação de duas dimensões de acordo com a expressão (3.6). A redução em imagens menores acarreta em uma redução na complexidade computacional no cômputo da transformada. A transformação tem como objetivo provocar a decorrelação entre os *pixels* da imagem, assim como compactar a energia em um menor número de coeficientes. Sendo assim, a dimensão N da subimagem não pode ser arbitrária, uma

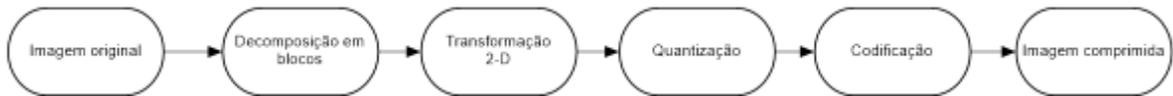


Figura 5.1: *Codificador de imagem por transformada de bloco.*

vez que valores muito pequenos não explorariam a decorrelação de quantidade considerável de *pixels*. Usualmente, a transformada utilizada se trata da DCT e é utilizado o valor $N = 8$. Para o tipo de imagens considerado, a DCT tem a capacidade de concentrar a maior parte da energia nos coeficientes de mais baixa frequência. Dessa forma, a maior parte da informação contida no bloco \mathbf{B} é concentrada no canto superior esquerdo do bloco \mathbf{X} . Neste trabalho, a etapa do cômputo da DCT foi substituída pelas diversas aproximações podadas descritas no Capítulo 4.

Em seguida, os blocos \mathbf{X} são submetidos à etapa de quantização [22,60,85,97], resultando na matriz quantizada \mathbf{Y} . Esta etapa consiste em divisão termo a termo pelos elementos de uma matriz \mathbf{Q} , denominada matriz de quantização, e subsequente arredondamento. A etapa de quantização é representada matematicamente por

$$y_{k,n} = \text{round} \left(\frac{x_{k,n}}{q_{k,n}} \right), \quad (5.14)$$

em que $y_{k,n}$, $x_{k,n}$ e $q_{k,n}$ são as entradas das matrizes \mathbf{Y} , \mathbf{X} e \mathbf{Q} , respectivamente. A operação $\text{round}(\cdot)$ representa arredondamento. Os elementos da matriz de quantização são projetados de modo a provocar uma maior atenuação nos coeficientes de alta frequência, uma vez que concentram menos energia. Dessa forma, coeficientes de maior frequência tem maiores chances de serem zerados.

Em seguida, cada bloco quantizado é submetido ao processo de codificação. Nesta etapa, os coeficientes quantizados são reorganizados de acordo com o esquema *zig-zag* [4] e subsequentemente submetidos a um codificador de entropia, usualmente o código de Huffman [4,5]. Ao fim desta etapa para todos os blocos, a imagem está em seu formato comprimido. O sistema de codificação por transformada de bloco é ilustrado na Figura 5.1.

Para recuperar a imagem, o procedimento inverso é realizado. Uma vez que a função de arredondamento $\text{round}(\cdot)$ não é injetiva e, conseqüentemente, não admite função inversa, a imagem original não pode ser recuperada de forma exata. Entretanto, a imagem recuperada representa uma aproximação para a imagem original. Como a maior parte dos coeficientes perdidos na etapa de quantização são de alta frequência e o olho humano é mais sensível às componentes de baixa frequência, a degradação da imagem aproximada é, em geral, imper-

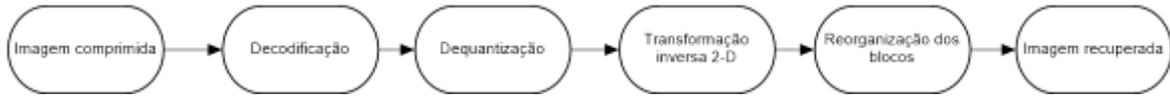


Figura 5.2: Decodificador de imagem por transformada de bloco.

ceptível. O decodificador é ilustrado na Figura 5.2.

5.3.1 ENERGIA CONCENTRADA POR COEFICIENTE E POR ZONA

Para avaliar a capacidade de compactação de energia das aproximações podadas, foi realizado um estudo sobre a distribuição de energia no domínio da transformada. A energia de uma matriz $\mathbf{X}_{n,m}$ de dimensão $N \times N$ é dada por [22, 64]

$$\mathcal{E} = \sum_{n=1}^N \sum_{m=1}^N |\mathbf{X}_{n,m}|^2. \quad (5.15)$$

Foi considerado um conjunto de 50 imagens de dimensão 512×512 obtidas de um banco público de imagens [98]. Como consideramos blocos de transformada de dimensão 8×8 , cada imagem é subdividida em blocos com esta dimensão. Cada bloco de cada imagem é então transformado de acordo com a Equação (4.19). Então, a energia contida em cada componente $\mathbf{X}_{n,m}$, dada por $|\mathbf{X}_{n,m}|^2$, é computada. Também é calculada a energia total do bloco, dada pela Equação (5.15). Assim, o percentual relativo de energia $\mathcal{E}_{rel}(n, m)$ contida em cada componente é computado tomando a razão entre a energia contida na componente $\mathbf{X}_{n,m}$ e a energia total do bloco \mathcal{E} , como definido a seguir:

$$\mathcal{E}_{rel}(n, m) = \frac{|\mathbf{X}_{n,m}|^2}{\mathcal{E}} \cdot 100\%. \quad (5.16)$$

A média entre todos os blocos de todas as figuras é considerada. Dessa forma, a distribuição média do percentual de energia armazenada em cada componente é obtida. Em geral, a maior parte da energia é concentrada no primeiro elemento da matriz $\mathbf{X}_{1,1}$ (valor médio ou valor DC). Para melhor visualização gráfica da distribuição relativa de energia, define-se a energia relativa em dB como

$$\mathcal{E}_{rel,dB}(n, m) = 10 \log \left(\frac{|\mathbf{X}_{n,m}|^2}{\mathcal{E}} \right). \quad (5.17)$$

Outra medida de compactação de energia é feita computando o percentual de energia armazenado em zonas quadradas do bloco. São consideradas as zonas quadradas no canto superior esquerdo da matriz, onde a energia deve se concentrar de acordo com a propriedade de compactação de energia da DCT. A quantidade relativa de energia contida numa zona

$K \times K$ é definida como

$$\mathcal{E}_{\text{zona}}(K) = \sum_{n=0}^{K-1} \sum_{m=0}^{K-1} \mathcal{E}_{\text{rel}}(n, m). \quad (5.18)$$

As distribuições de energia relativa em dB para as WHT, SDCT, LODCT, RDCT e MRDCT são mostrada na Figura 5.3, enquanto os métodos da série BAS e os métodos baseados em funções inteiras são mostrados nas Figura 5.4 e Figura 5.5, respectivamente. A distribuição para a DCT exata foi incluída em cada uma das figuras para fins de comparação. A Tabela 5.2 apresenta os valores da quantidade relativa de energia por zona para cada um dos métodos e para cada valor de K .

Qualitativamente, observa-se que para a DCT exata, a energia se concentra nos termos mais próximos do termo DC $(n, m) = (1, 1)$. A distribuição para a DCT exata na Figura 5.5(a) apresenta uma curva suave com decaimento de energia regular ao se aproximar do termo de mais alta frequência $(n, m) = (8, 8)$. Em contrapartida, a distribuição de energia da MRDCT, mostrada na Figura 5.3(f), apresenta uma superfície mais irregular do que a curva relacionada à DCT exata. Esse fato sugere que, de fato, a DCT exata apresenta uma melhor capacidade de compactação de energia, apesar de ser um método computacionalmente mais custoso.

Tomemos com referência para análise a quantidade de $\approx 99\%$ de energia concentrada. Na Tabela 5.2, nota-se que a DCT concentra $\approx 99.13\%$ da energia na zona 4×4 . Observa-se também que a MRDCT concentra $\approx 99.34\%$ da energia na zona de tamanho $K = 6$. Já a LODCT apresenta um desempenho mais próximo da DCT exata, concentrando $\approx 98.98\%$ em $K = 4$, sendo a aproximação com melhor desempenho segundo esta métrica para $1 < K < 8$. Este resultado é esperado, uma vez que a LODCT apresentou melhores medidas de desempenho, conforme exposto na Seção 3.5.4. O caso particular $K = 1$ apresenta valor idêntico para todos os métodos, uma vez que todos os métodos apresentam a primeira linha da matriz iguais entre si. Já o caso degenerado $K = 8$ representa a matriz inteira, concentrando 100% da energia.

A MRDCT apresenta a vantagem de ser o método com menor custo computacional na literatura até o presente momento, com 14 adições. O método MRDCT apresenta as linhas $n = 6$ e $m = 6$ com uma protuberância de energia, apresentando um máximo local de energia em $(n, m) = (6, 6)$. Este fato sugere que parte importante da informação dos sinal transformado pode ser armazenado neste coeficiente. Posteriormente, mostraremos que fazer $K < 6$ para este método acarreta em considerável perda de desempenho, apesar de diminuir o custo computacional.

Tabela 5.2: *Energia relativa por zona para cada método*

Método	K							
	1	2	3	4	5	6	7	8
Exact DCT	95.46	97.47	98.55	99.13	99.49	99.71	99.87	100.00
WHT [87]	95.46	97.18	98.08	98.76	99.10	99.44	99.77	100.00
SDCT [33]	94.02	96.39	97.30	98.16	98.52	99.26	99.61	100.00
BAS-1 [36]	95.46	97.08	98.10	98.86	99.20	99.51	99.68	100.00
BAS-2 [37]	95.46	97.08	97.96	98.71	99.04	99.35	99.68	100.00
BAS-3 [39]	95.46	97.18	98.22	98.98	99.33	99.67	99.84	100.00
BAS-4 [40]	95.46	97.08	97.95	98.70	99.03	99.32	99.65	100.00
BAS-5 [40]	95.46	97.08	98.10	98.86	99.20	99.49	99.82	100.00
BAS-6 [40]	95.46	97.08	97.96	98.71	99.04	99.33	99.66	100.00
BAS-7 [43]	95.46	97.18	98.08	98.76	99.10	99.44	99.77	100.00
LODCT [34]	95.46	97.36	98.43	98.98	99.32	99.59	99.75	100.00
RDCT [41]	95.46	97.36	98.28	98.81	99.16	99.41	99.75	100.00
MRDCT [42]	95.46	96.41	97.22	97.91	98.22	99.34	99.68	100.00
INT-1 [32]	95.46	97.29	97.52	98.04	98.35	98.63	99.75	100.00
INT-2 [32]	95.46	97.29	98.19	98.77	99.11	99.41	99.75	100.00
INT-4 [32]	95.46	97.36	98.29	98.82	99.17	99.43	99.75	100.00
INT-5 [32]	95.46	97.29	98.20	98.78	99.12	99.42	99.75	100.00
INT-6 [32]	95.46	97.29	98.34	98.93	99.27	99.58	99.75	100.00

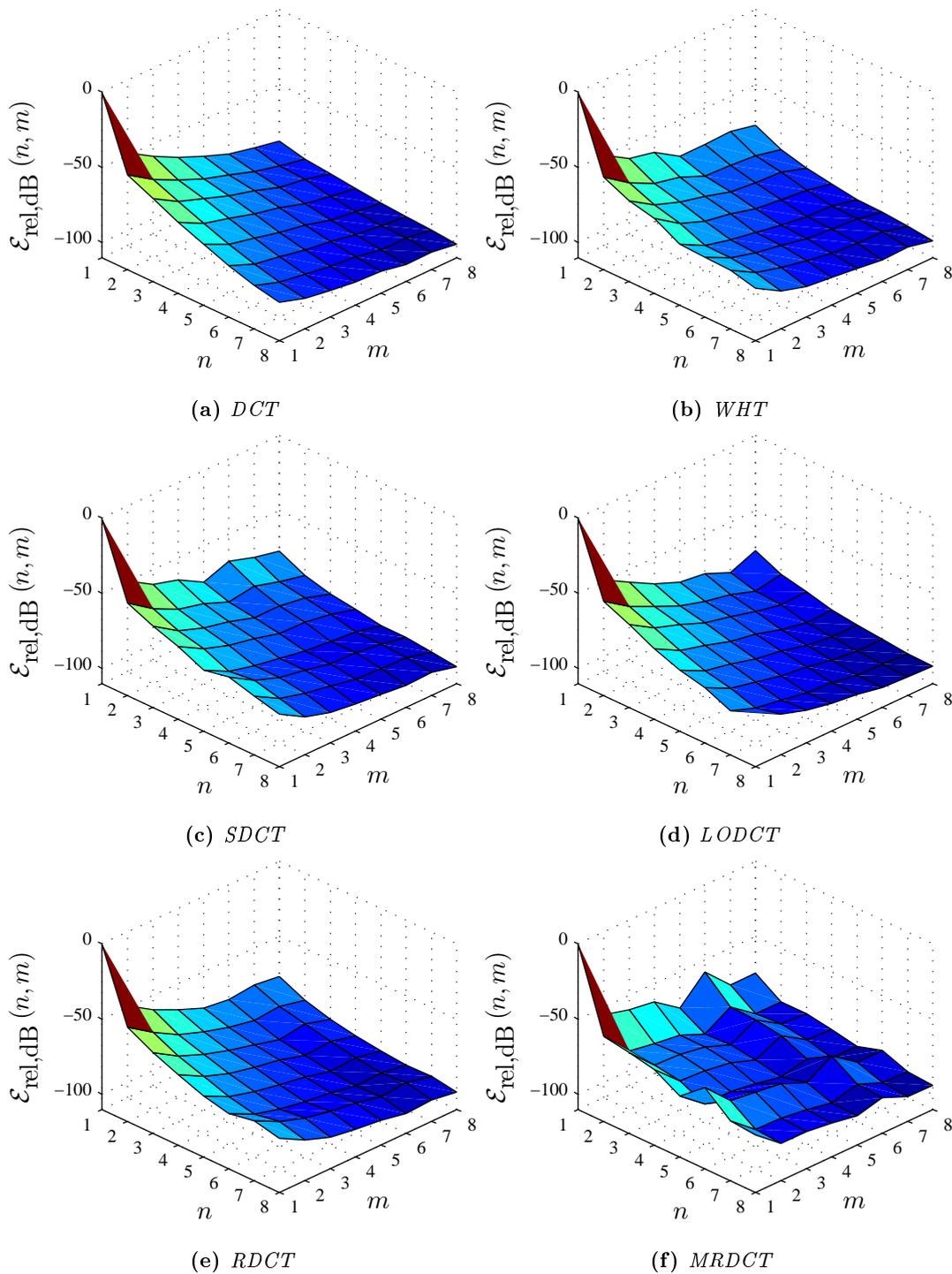


Figura 5.3: Distribuição relativa de energia $\mathcal{E}_{rel,dB}(n, m)$ para a DCT exata, WHT, SDCT, LODCT, RDCT e MRDCT.

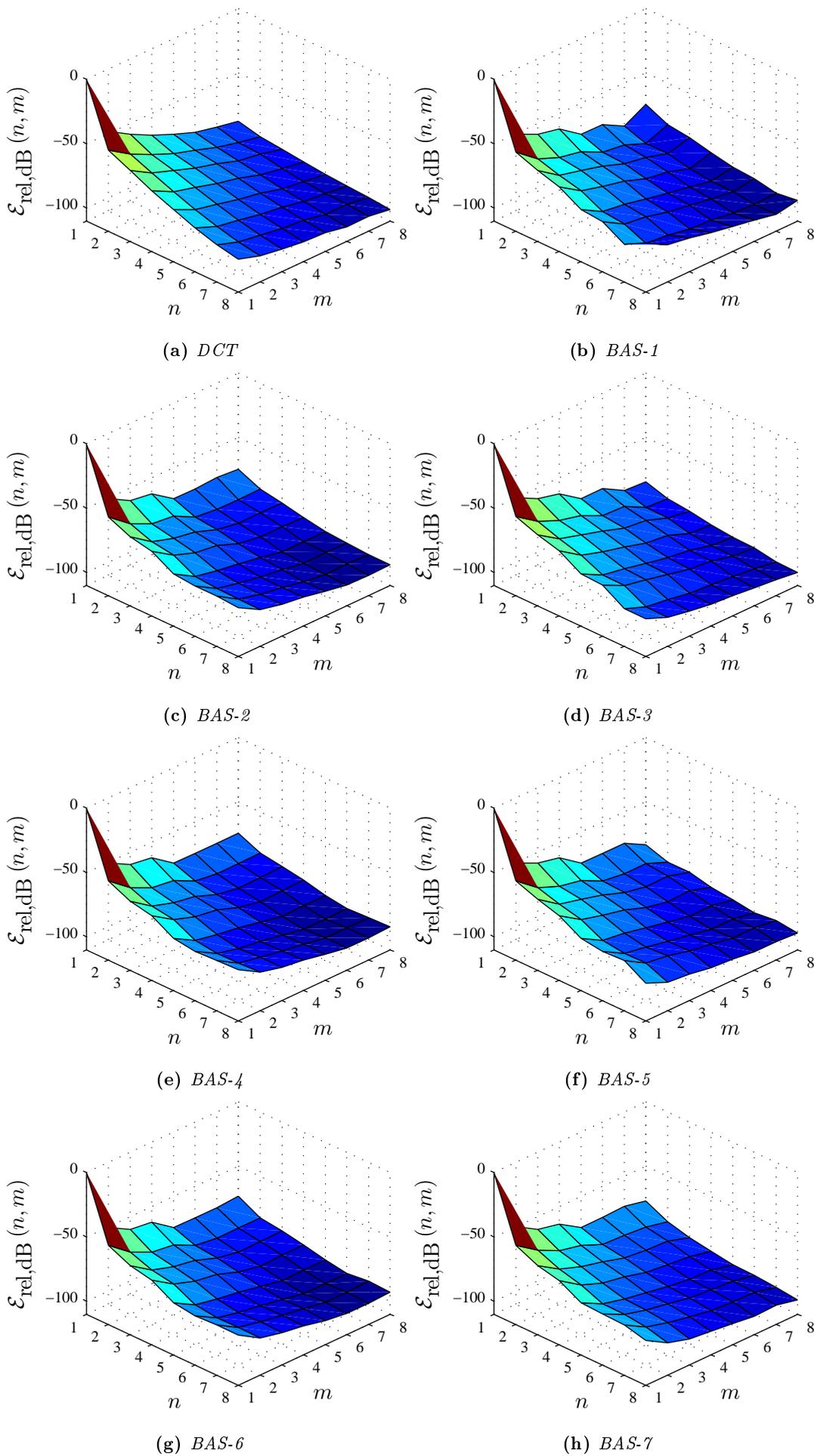


Figura 5.4: Distribuição relativa de energia $\mathcal{E}_{rel,dB}(n, m)$ para os métodos da Série BAS.

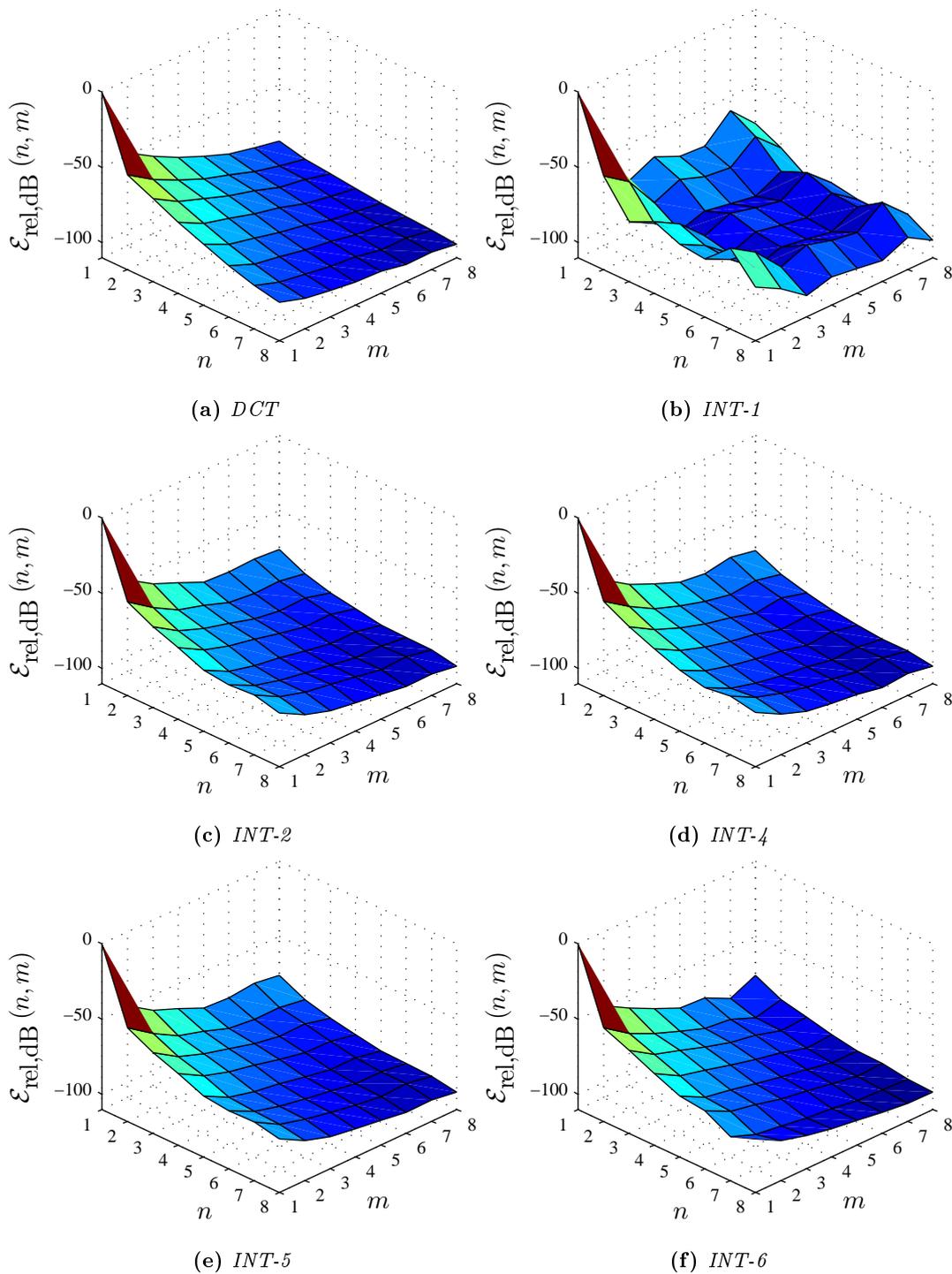


Figura 5.5: Distribuição relativa de energia $\mathcal{E}_{rel,dB}(n,m)$ para os métodos baseados em funções inteiras.

5.3.2 METODOLOGIA DA SIMULAÇÃO JPEG

Cada imagem de entrada é dividida em blocos 8×8 . Cada pixel é subtraído do valor 128 para alterar a faixa de valores entre 0 e 255 para -128 a 127. Então, a cada bloco é aplicado a transformação 2-D da aproximação $\hat{\mathbf{C}}$ em questão. Seja um bloco \mathbf{B} da imagem, então é obtida a matriz transformada $\mathbf{X} = \hat{\mathbf{C}} \cdot \mathbf{B} \cdot \hat{\mathbf{C}}^\top$. Esta etapa resulta em uma matriz $K \times K$ para cada bloco no domínio da transformada. Para uma aproximação usual não podada, temos que $K = 8$.

Para cada matriz gerada é aplicado o procedimento de quantização, conforme descrito anteriormente nesta Seção. É nesta etapa que parte da informação do sinal é descartada. Em se tratando de aproximações para a DCT, a etapa usual de quantização [85, p. 81] deve ser modificada para explorar as características de baixa complexidade das matrizes. De acordo com a teoria desenvolvida em [4], a matriz diagonal \mathbf{D} de ortonormalização, apresentada na Seção 3.3, não influencia na complexidade aritmética, uma vez que os elementos podem ser incorporados na matriz \mathbf{Q} na etapa de quantização. Dessa forma, uma nova matriz de quantização \mathbf{Q}^* é utilizada, cujos elementos $q_{k,n}^*$ são dados por

$$q_{k,n}^* = \frac{q_{k,n}}{\delta_{k,n}}, \quad (5.19)$$

em que $\delta_{k,n}$ são as entradas da matriz

$$\mathbf{\Delta} = \text{diag}_v(\mathbf{D}) \otimes \text{diag}_v(\mathbf{D}),$$

e $\text{diag}_v(\cdot)$ retorna um vetor coluna cujos elementos são dados pela diagonal principal do argumento.

Sendo assim, a nova etapa de quantização para uma aproximação não podada é caracterizada pela seguinte equação

$$Y_{k,n} = \text{round} \left(\frac{X_{k,n}}{q_{k,n}^*} \right), \quad k, n = 1, 2, \dots, 8, \quad (5.20)$$

em que $X_{k,n}$ é um elemento da matriz transformada \mathbf{X} e $Y_{k,n}$ é um elemento da matriz quantizada \mathbf{Y} . O elemento $q_{k,n}$ corresponde a uma entrada de uma matriz de quantização \mathbf{Q} , que é especificada a priori. Entretanto a definição (5.20) é aplicável para transformadas usuais de dimensão 8×8 .

A matriz transformada $\mathbf{X}_{\langle K \rangle}$ tem dimensão $K \times K$ e possui os K^2 coeficientes da parte superior e esquerda da transformação não podada. Dessa forma, há apenas tal quantidade

de coeficientes para serem submetidos à etapa de quantização. Portanto apenas os K^2 coeficientes da matriz modificada \mathbf{Q}^* são necessários para a etapa de quantização, reduzindo a memória física necessária para esta etapa. A nova etapa de quantização é feita de forma semelhante à Equação (5.14) utilizando a matriz quadrada $\mathbf{Q}_{[K]}^*$, derivada de \mathbf{Q}^* de acordo com a Definição 4.5.

Então, a Equação (5.20) pode ser generalizada como

$$Y_{k,n} = \text{round} \left(\frac{X_{k,n}}{q_{k,n}^*} \right), \quad k, n = 1, 2, \dots, K. \quad (5.21)$$

Os elementos da matriz de quantização são definidos empiricamente através de técnicas para avaliar a percepção humana [85, p. 154]. Nas simulações de compressão de imagem consideradas neste trabalho, foi utilizada a matriz de quantização baseada em luminância [85, p. 155]

$$\mathbf{Q} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

No processo de recuperação da imagem comprimida por uma transformada podada, é necessário realizar o processo inverso. Devido ao erro causado pela operação $\text{round}(\cdot)$, não é possível recuperar o valor exato de $X_{k,n}$. Um valor aproximado é obtido através da expressão

$$\hat{X}_{k,n} = Y_{k,n} \cdot q_{k,n}^*, \quad k, n = 1, 2, \dots, K, \quad (5.22)$$

em que $\hat{X}_{k,n}$ é elemento da matriz $\hat{\mathbf{X}}$. A matriz $\hat{\mathbf{X}}$ representa uma aproximação para a matriz \mathbf{X} e também possui dimensão $K \times K$. A partir desta matriz, é obtido o bloco $\hat{\mathbf{B}} = \hat{\mathbf{C}}^\top \cdot \hat{\mathbf{X}} \cdot \hat{\mathbf{C}}$ da imagem comprimida, representando o bloco original \mathbf{B} recuperado após a compressão. É fácil notar que $\hat{\mathbf{B}}$ possui dimensão 8×8 , assim como o bloco original.

No intuito de avaliar quantitativamente a qualidade da imagem recuperada após a compressão, foram utilizadas como figuras de mérito as medidas de PSNR e SSIM. O MSE não foi considerado, uma vez que apresenta uma relação direta com o PSNR.

5.3.3 RESULTADOS

As imagens mencionadas anteriormente foram submetidas ao procedimento de compressão descrito utilizando as aproximações podadas e originais. Os gráficos para o PSNR médio e o SSIM médio são mostrados nas Figura 5.6 e Figura 5.7, respectivamente. Devido ao grande número de curvas nos gráficos, os valores de PSNR e SSIM para cada método e para cada K também são mostrados nas Tabela 5.3 e Tabela 5.4, respectivamente. Uma análise qualitativa é fornecida através da imagem ‘Lena’. As imagens recuperadas após a compressão com os métodos derivados das WHT, SDCT, LODCT, RDCT, MRDCT, BAS-1 e INT-6 são mostradas nas Figura 5.9, 5.10, 5.11, 5.12, 5.13, 5.14 e B.1, respectivamente. A imagem original não comprimida é apresentada em cada uma das figuras para comparação, assim como o valor de PSNR e SSIM para esta figura específica.

De fato, o SSIM representa uma medida de qualidade visual mais fiel do que o PSNR, uma vez que leva em consideração o HVS [96]. Entretanto, quando as diferentes imagens apresentam mesma fonte de distorção (e.g., variação de contraste, valor médio alterado, imagem borrada, procedimento de compressão, etc.), o PSNR é uma medida que captura satisfatoriamente a qualidade visual percebida. Como todas as imagens apresentam distorção gerada a partir de um procedimento de compressão, as curvas de PSNR e SSIM não divergem consideravelmente na informação sobre a qualidade visual apresentada.

Como era de se esperar, a DCT exata apresenta melhor desempenho para qualquer valor de K . Entretanto, os métodos derivados apresentam performance competitiva a um custo computacional consideravelmente reduzido. Por exemplo, considerando a DCT exata podada com $K = 4$, a aproximação BAS-3 podada com $K = 5$ e a MRDCT podada com $K = 7$. Os valores de PSNR médio para cada um desses casos são dados por 30.40 dB, 30.67 dB e 30.29 dB, respectivamente. Já os valores de SSIM são 0.86, 0.87 e 0.84, respectivamente. Consideremos que o algoritmo podado para a DCT exata seja derivado do algoritmo de Loeffler [20], que possui a cota mínima multiplicativa [31]. As complexidades aritmética para computar a transformada de um bloco 8×8 para cada um de tais casos mencionados são 108 multiplicações + 288 adições [48], 273 adições + 26 deslocamentos de *bits* e 195 adições, respectivamente (vide Tabela 4.17). A redução do custo computacional é significativa e ainda há um incremento no desempenho para o caso de BAS-3 podado. Já o caso da MRDCT podada apresenta redução de desempenho tolerável.

De forma geral, os métodos que apresentam melhor desempenho estão associadas a um

custo computacional mais elevado. Mas esse fato não é uma regra geral, podendo haver um caso em que o custo é reduzido e a desempenho é incrementada, conforme mostrado no exemplo dado acima. As curvas de desempenho para cada método podado estão relacionadas com as medidas de desempenho para o método original não podado, apresentadas na Seção 3.5.4. Por exemplo, a curva de PSNR das aproximações derivadas de BAS-3 se mostra acima das demais para $K > 4$, enquanto a MRDCT se apresenta mais abaixo. Os valores de ganho de codificação para estes métodos são 8.32 dB e 7.33 dB, respectivamente. Para visualizar esta associação de maneira mais ampla, a Figura 5.6 apresenta as curvas de PSNR divididas pelo ganho de codificação C_g do método original não podado do qual a aproximação podada foi derivada. As curvas apresentam uma grande sobreposição, sugerindo que o ganho de codificação está relacionado com a desempenho dos métodos derivados.

Para o caso particular da MRDCT, observa-se um incremento considerável entre os casos $K = 4$ e $K = 5$ de 2.81 dB (PSNR) e 0.07 (SSIM). Este fato foi previsto anteriormente observando a curva de energia relativa $\mathcal{E}_{\text{rel,dB}}$, devido ao acúmulo de energia nas linhas e colunas $n = 6$ e $m = 6$ no domínio da transformada, Figura 5.3(f). Um fenômeno similar pode ser observado para o caso da aproximação INT-1 podadas com $K = 6$ e $K = 7$.

De forma geral, levando em conta as curvas de PSNR e SSIM, assim como as complexidades aritméticas, as aproximações podados representam uma alternativa de baixo consumo energético para aplicações de codificação de imagem e vídeo. A escolha de um método podado particular depende de diversos fatores relacionados à aplicação, como desempenho desejado, consumo mínimo de energia necessário, quantidade de componentes de *hardware*, etc. Na prática, a derivação de 96 novos métodos permite um maior ajuste fino no *trade-off* entre custo computacional e desempenho, proporcionando uma maior gama de ferramentas para aplicações de compressão de dados.

5.4 ARQUITETURAS VLSI

O trabalho apresentado nesta Seção foi desenvolvido em parceria com a Universidade de Akron, com Dr. Arjuna Madanayake e Sunera Kulasekera. Alguns dos novos métodos podados propostos foram mapeados em arquiteturas digitais e realizados fisicamente utilizando tecnologia CMOS. As implementações foram consideradas em FPGAs Xilinx Virtex-6 e BEE3 (Xilinx Virtex-5) [99].

A LODCT e a MRDCT apresentam, respectivamente, o melhor desempenho e a menor

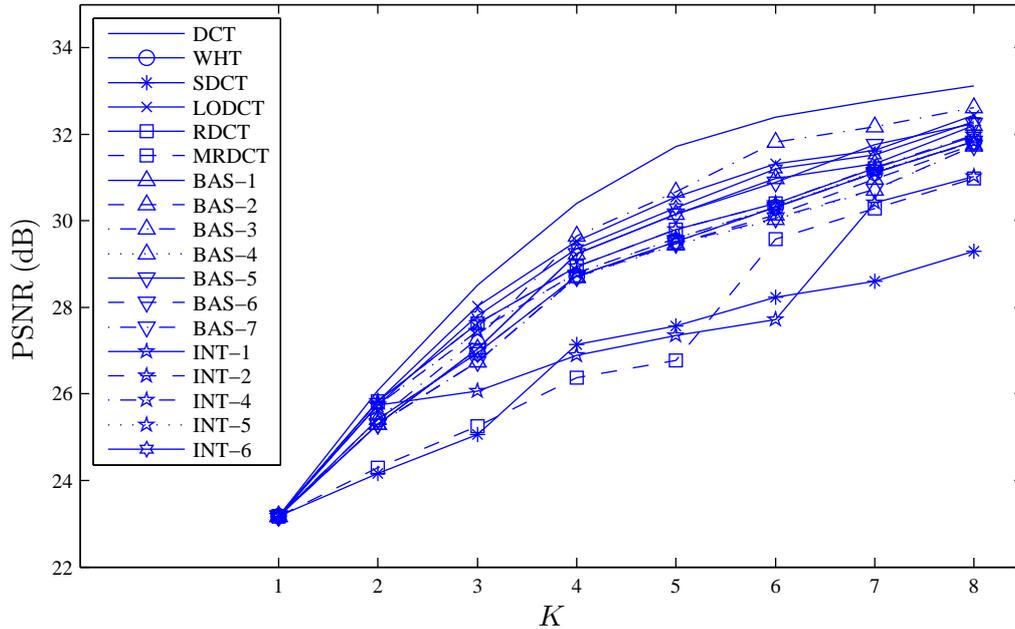


Figura 5.6: Curvas de PSNR médio para cada método.

complexidade aritmética dentre os métodos abordados. Portanto, foram considerados tais métodos podados para o desenvolvimento em VLSI. Nesta Seção, denota-se por simplicidade $\mathbf{M} = \mathbf{T}_{\text{MRDCT}}$ e $\mathbf{L} = \mathbf{T}_{\text{LODCT}}$.

Para a escolha do parâmetro K de cada método, foi utilizado a definição de energia relativa por zona $\mathcal{E}_{\text{zona}}(K)$, descrito na Seção 5.3.1. Foi considerado o critério de $\mathcal{E}_{\text{zona}}(K) \approx 99\%$, que corresponde a $K = 4$ para a LODCT e a $K = 6$ para a MRDCT, conforme apresentado na Tabela 5.2. Sendo assim, são consideradas as aproximações podadas $\mathbf{M}_{(6)}$ e $\mathbf{L}_{(4)}$, cujas arquiteturas em 1-D são apresentadas nas Figura 5.17 e Figura 5.16, respectivamente. Tais arquiteturas são baseadas nos algoritmos apresentados no Apêndice A.

As arquiteturas foram implementadas em sistema multi-FPGA BEE3. O sistema BEE3 consiste em quatro dispositivos FPGAs Xilinx Virtex-5 acopladas e é mostrado na Figura 5.18. Foram gerados, através do ambiente MATLAB, 10000 vetores randômicos de teste de comprimento 8. Tais vetores foram roteados para o BEE3 via porta USB para serem processados pelas transformações podadas implementadas.

A avaliação da complexidade de hardware foi realizada através das seguintes métricas [44]: número de blocos lógicos configuráveis (CLBs) utilizados; número de *flip-flops* (FFs); o atraso de caminho crítico (T_{cpd}), em ns; a frequência máxima de operação (F_{max}), em MHz; potência dinâmica Q_p , em mW/MHz; e potência estática D_p , em W. Os resultados são mostra-

Tabela 5.3: Valores de PSNR médio para cada método

Método	K							
	1	2	3	4	5	6	7	8
Exact DCT	23.17	26.08	28.52	30.40	31.71	32.39	32.78	33.12
WHT [87]	23.17	25.41	26.95	28.73	29.51	30.31	31.12	31.84
SDCT [33]	23.17	24.28	25.23	27.15	27.59	28.43	28.82	29.84
BAS-1 [36]	23.17	25.30	27.04	29.34	30.15	30.97	31.33	32.20
BAS-2 [37]	23.17	25.30	26.95	28.70	29.47	30.14	30.96	31.76
BAS-3 [39]	23.17	25.41	27.27	29.65	30.67	31.81	32.18	32.61
BAS-4 [40]	23.17	25.30	26.75	28.70	29.45	30.04	30.72	31.73
BAS-5 [40]	23.17	25.30	27.04	29.24	30.15	30.89	31.77	32.25
BAS-6 [40]	23.17	25.30	26.75	28.70	29.47	30.07	30.73	31.73
BAS-7 [43]	23.17	24.41	26.95	28.73	29.51	30.31	31.12	31.84
LODCT [34]	23.17	25.83	28.02	29.53	30.55	31.32	31.63	32.44
RDCT [41]	23.17	25.83	27.64	28.94	29.79	30.41	31.21	31.96
MRDCT [42]	23.17	24.29	25.26	26.37	26.77	29.58	30.29	30.98
INT-1 [32]	23.17	25.74	26.07	26.89	27.34	27.72	30.42	31.02
INT-2 [32]	23.17	25.74	27.45	28.80	29.60	30.32	31.11	31.82
INT-4 [32]	23.17	25.83	27.63	28.94	29.80	30.41	31.24	31.98
INT-5 [32]	23.17	25.74	27.44	28.79	29.60	30.32	31.14	31.84
INT-6 [32]	23.17	25.74	27.81	29.35	30.30	31.21	31.52	32.29

dos na Tabela 5.5. De forma geral, a MRDCT podada com $K = 6$ apresentou menor consumo de recursos de *hardware* do que a LODCT podada com $K = 4$.

Também foram realizadas implementações em FPGA, assim como síntese em ASIC, para a MRDCT considerando todos os possíveis valores de K . As arquiteturas foram desenvolvidas no dispositivo Xilinx Virtex-6 XC6VLX240T-1FFG1156. Um procedimento de teste análogo ao descrito anteriormente foi realizado. Os resultados são mostrados na Tabela 5.6. Os resultados apontam uma redução considerável nos recursos de *hardware* ao decrementar o valor de K . Considerando o caso $K = 6$, a redução do número de FF é de $\approx 24\%$, enquanto a do número de CLBs é de $\approx 21\%$.

Todos as aproximações MRDCT podadas foram sintetizadas em ASIC, utilizando o sistema de implementação digital *Cadence Encounter*. Foram utilizadas células ASIC 45 nm CMOS padrão do FreePDK. A tensão de alimentação para a implementação CMOS foi fixada em $V_{DD} = 1.1 V$. Para a síntese ASIC, foram adotadas as seguintes figuras de mérito [44]: complexidade tempo-área (AT), em $\text{mm}^2 \cdot \text{ns}$; complexidade tempo-quadrado-área (AT^2),

Tabela 5.4: Valores de SSIM médio para cada método

Método	K							
	1	2	3	4	5	6	7	8
Exact DCT	0.48	0.66	0.79	0.86	0.89	0.90	0.90	0.90
WHT [87]	0.48	0.64	0.74	0.82	0.85	0.87	0.87	0.88
SDCT [33]	0.48	0.59	0.67	0.77	0.80	0.81	0.82	0.84
BAS-1 [36]	0.48	0.62	0.74	0.83	0.85	0.87	0.88	0.89
BAS-2 [37]	0.48	0.62	0.73	0.82	0.84	0.85	0.87	0.88
BAS-3 [39]	0.48	0.64	0.75	0.84	0.87	0.89	0.89	0.89
BAS-4 [40]	0.48	0.62	0.72	0.81	0.83	0.85	0.86	0.87
BAS-5 [40]	0.48	0.62	0.74	0.83	0.85	0.87	0.88	0.89
BAS-6 [40]	0.48	0.62	0.73	0.82	0.84	0.86	0.87	0.88
BAS-7 [43]	0.48	0.64	0.74	0.82	0.85	0.87	0.87	0.88
LODCT [34]	0.48	0.64	0.74	0.82	0.85	0.87	0.87	0.88
RDCT [41]	0.48	0.66	0.77	0.84	0.87	0.88	0.89	0.89
MRDCT [42]	0.48	0.55	0.65	0.72	0.76	0.83	0.84	0.86
INT-1 [32]	0.48	0.64	0.67	0.74	0.76	0.78	0.85	0.86
INT-2 [32]	0.48	0.64	0.75	0.82	0.84	0.86	0.87	0.88
INT-4 [32]	0.48	0.66	0.76	0.82	0.85	0.86	0.88	0.88
INT-5 [32]	0.48	0.64	0.75	0.82	0.84	0.86	0.87	0.88
INT-6 [32]	0.48	0.64	0.76	0.83	0.86	0.88	0.88	0.89

Tabela 5.5: Consumo de recurso de hardware utilizando Xilinx Virtex-5 para a MRDCT e LODCT

Métrica	Method	
	LODCT podada $K = 4$	MRDCT podada $K = 6$
CLB	298	232
FF	1054	895
T_{cpd} (ns)	3.578	3.588
F_{max} (MHz)	279.48	278.70
D_p (mW/MHz)	3.141	3.620
Q_p (W)	1.50	1.50

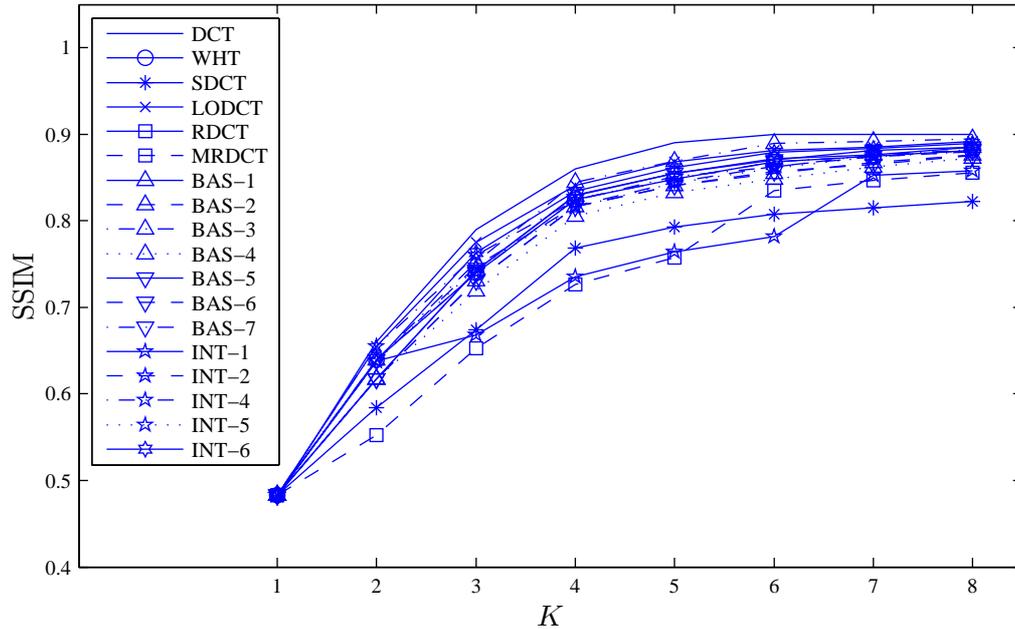


Figura 5.7: Curvas de SSIM médio para cada método.

em $\text{mm}^2 \cdot \text{ns}^2$; potência dinâmica Q_p , em mW/MHz ; potência estática D_p , em mW ; o atraso de caminho crítico (T_{cpd}), em ns ; e a frequência máxima de operação (F_{max}), em GHz . Os resultados são mostrados na Tabela 5.7. Os valores apontam uma redução considerável de recursos. A exemplo, considerando ainda o caso $K = 6$, os resultados apontam uma redução de $\approx 17\%$ em área ocupada.

Tabela 5.6: Consumo de recurso de hardware utilizando Xilinx Virtex-6 para a MRDCT podada

K	CLB	FF	T_{cpd} (ns)	F_{max} (MHz)	D_p (mW/MHz)	Q_p (W)
1	107	376	2.263	441.89	0.67	3.43
2	136	568	2.300	434.78	0.97	3.43
3	210	783	2.509	398.56	0.87	3.43
4	247	961	2.946	339.44	1.35	3.43
5	290	1123	2.877	347.58	1.70	3.43
6	350	1286	2.735	365.63	2.07	3.44
7	424	1487	3.300	303.03	2.21	3.44
8	445	1696	3.390	294.98	2.74	3.44

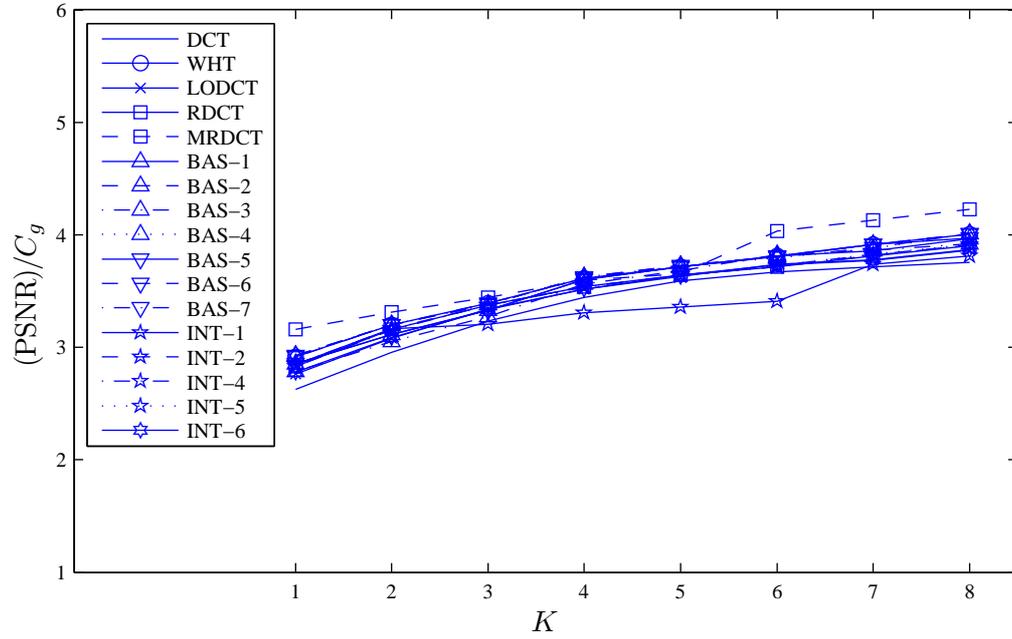


Figura 5.8: Curvas de PSNR médio divididas pelo ganho de codificação C_g de cada método.

Tabela 5.7: Consumo de recursos para síntese CMOS 45 nm ASIC para a MRDCT podada

K	Area	AT	AT^2	T_{cpd}	F_{max}	D_p	Q_p
1	0.011	0.011	0.010	0.961	1.040	0.018	0.088
2	0.017	0.016	0.015	0.962	1.039	0.028	0.125
3	0.022	0.021	0.020	0.963	1.038	0.038	0.167
4	0.027	0.027	0.026	0.970	1.030	0.047	0.208
5	0.032	0.034	0.037	1.075	0.930	0.057	0.246
6	0.038	0.038	0.037	0.995	1.005	0.067	0.288
7	0.043	0.047	0.051	1.085	0.921	0.079	0.333
8	0.046	0.051	0.057	1.103	0.906	0.084	0.357



(a) Sem compressão



(b) Método não podado ($PSNR = 33.56$ dB e $SSIM = 0.89$)



(c) $K = 7$ ($PSNR = 33.06$ dB e $SSIM = 0.89$)



(d) $K = 6$ ($PSNR = 32.44$ dB e $SSIM = 0.89$)



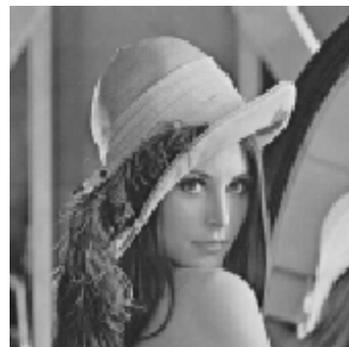
(e) $K = 5$ ($PSNR = 31.80$ dB e $SSIM = 0.88$)



(f) $K = 4$ ($PSNR = 31.20$ dB e $SSIM = 0.88$)

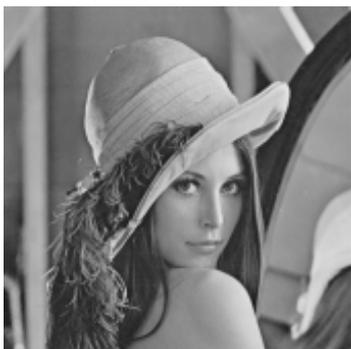


(g) $K = 3$ ($PSNR = 28.63$ dB e $SSIM = 0.82$)



(h) $K = 2$ ($PSNR = 26.90$ dB e $SSIM = 0.76$)

Figura 5.9: Compressão da imagem 'Lena' através de métodos derivados da WHT.



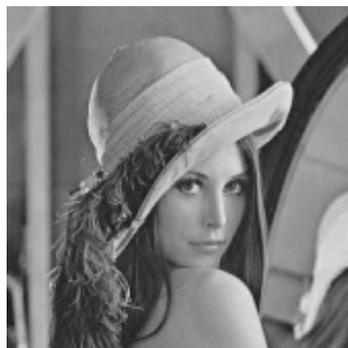
(a) Sem compressão



(b) Método não podado ($PSNR = 30.78$ dB e $SSIM = 0.84$)



(c) $K = 7$ ($PSNR = 30.29$ dB e $SSIM = 0.84$)



(d) $K = 6$ ($PSNR = 30.08$ dB e $SSIM = 0.83$)



(e) $K = 5$ ($PSNR = 29.20$ dB e $SSIM = 0.83$)



(f) $K = 4$ ($PSNR = 28.94$ dB e $SSIM = 0.82$)



(g) $K = 3$ ($PSNR = 26.34$ dB e $SSIM = 0.77$)



(h) $K = 2$ ($PSNR = 25.41$ dB e $SSIM = 0.71$)

Figura 5.10: Compressão da imagem 'Lena' através de métodos derivados da SDCT.



(a) *Sem compressão*



(b) *Método não podado (PSNR = 34.40 dB e SSIM = 0.90)*



(c) *K = 7 (PSNR = 33.78 dB e SSIM = 0.90)*



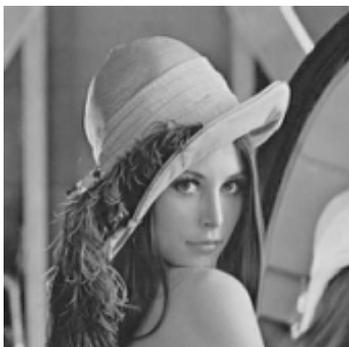
(d) *K = 6 (PSNR = 33.73 dB e SSIM = 0.90)*



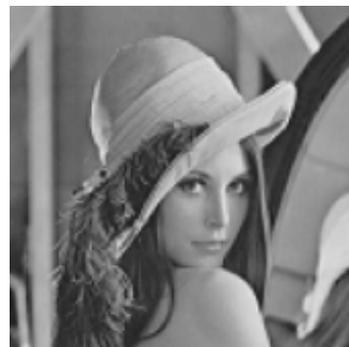
(e) *K = 5 (PSNR = 32.99 dB e SSIM = 0.89)*



(f) *K = 4 (PSNR = 32.17 dB e SSIM = 0.89)*



(g) *K = 3 (PSNR = 30.41 dB e SSIM = 0.85)*



(h) *K = 2 (PSNR = 27.64 dB e SSIM = 0.79)*

Figura 5.11: *Compressão da imagem ‘Lena’ através de métodos derivados da LODCT.*



(a) Sem compressão



(b) Método não podado ($PSNR = 33.72$ dB e $SSIM = 0.90$)



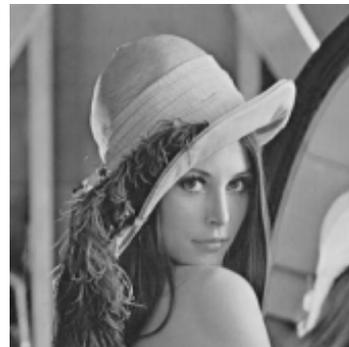
(c) $K = 7$ ($PSNR = 33.18$ dB e $SSIM = 0.89$)



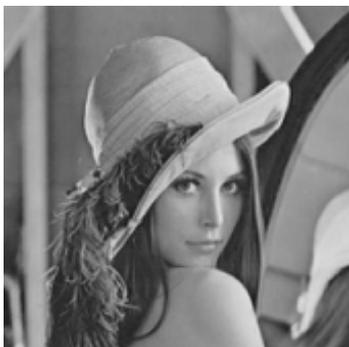
(d) $K = 6$ ($PSNR = 32.55$ dB e $SSIM = 0.89$)



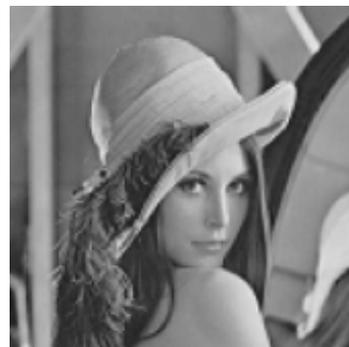
(e) $K = 5$ ($PSNR = 31.97$ dB e $SSIM = 0.88$)



(f) $K = 4$ ($PSNR = 31.34$ dB e $SSIM = 0.87$)



(g) $K = 3$ ($PSNR = 29.87$ dB e $SSIM = 0.84$)



(h) $K = 2$ ($PSNR = 27.64$ dB e $SSIM = 0.79$)

Figura 5.12: Compressão da imagem 'Lena' através de métodos derivados da RDCT.



(a) *Sem compressão*



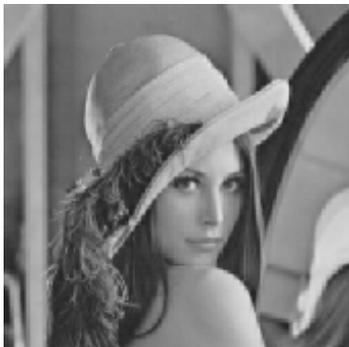
(b) *Método não podado (PSNR = 32.38 dB e SSIM = 0.87)*



(c) *K = 7 (PSNR = 32.02 dB e SSIM = 0.87)*



(d) *K = 6 (PSNR = 31.62 dB e SSIM = 0.86)*



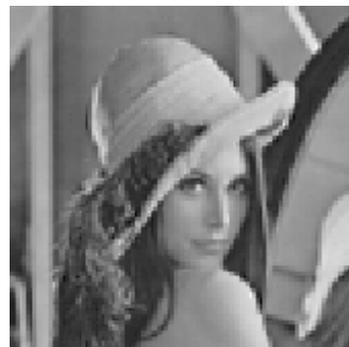
(e) *K = 5 (PSNR = 27.69 dB e SSIM = 0.80)*



(f) *K = 4 (PSNR = 27.48 dB e SSIM = 0.79)*



(g) *K = 3 (PSNR = 26.24 dB e SSIM = 0.75)*



(h) *K = 2 (PSNR = 25.29 dB e SSIM = 0.69)*

Figura 5.13: *Compressão da imagem 'Lena' através de métodos derivados da MRDCT.*



(a) Sem compressão



(b) Método não podado ($PSNR = 34.14$ dB e $SSIM = 0.90$)



(c) $K = 7$ ($PSNR = 33.53$ dB e $SSIM = 0.90$)



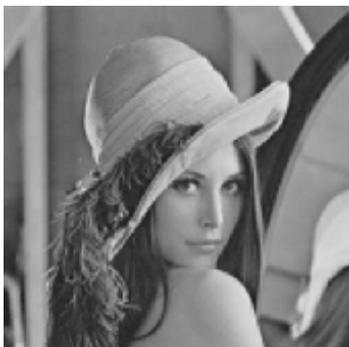
(d) $K = 6$ ($PSNR = 33.49$ dB e $SSIM = 0.90$)



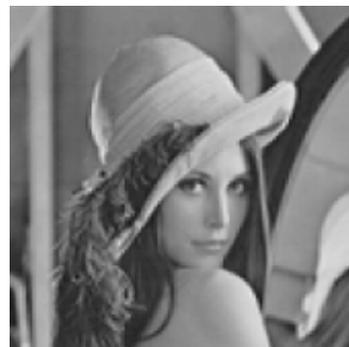
(e) $K = 5$ ($PSNR = 32.78$ dB e $SSIM = 0.89$)



(f) $K = 4$ ($PSNR = 32.04$ dB e $SSIM = 0.89$)



(g) $K = 3$ ($PSNR = 28.63$ dB e $SSIM = 0.83$)



(h) $K = 2$ ($PSNR = 26.87$ dB e $SSIM = 0.75$)

Figura 5.14: Compressão da imagem 'Lena' através de métodos derivados da BAS-1.



(a) *Sem compressão*



(b) *Método não podado (PSNR = 32.58 dB e SSIM = 0.88)*



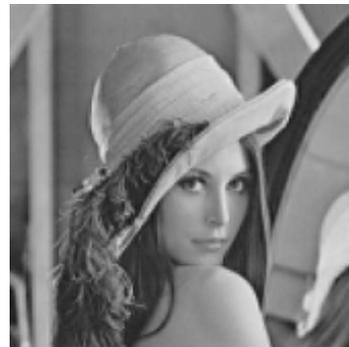
(c) *K = 7 (PSNR = 32.16 dB e SSIM = 0.87)*



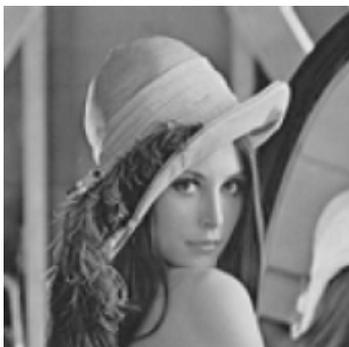
(d) *K = 6 (PSNR = 29.25 dB e SSIM = 0.83)*



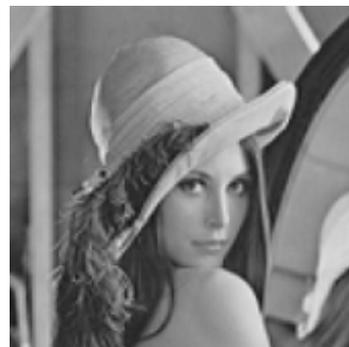
(e) *K = 5 (PSNR = 28.99 dB e SSIM = 0.82)*



(f) *K = 4 (PSNR = 28.70 dB e SSIM = 0.81)*



(g) *K = 3 (PSNR = 27.93 dB e SSIM = 0.79)*



(h) *K = 2 (PSNR = 27.63 dB e SSIM = 0.78)*

Figura 5.15: *Compressão da imagem ‘Lena’ através de métodos derivados da INT-6.*

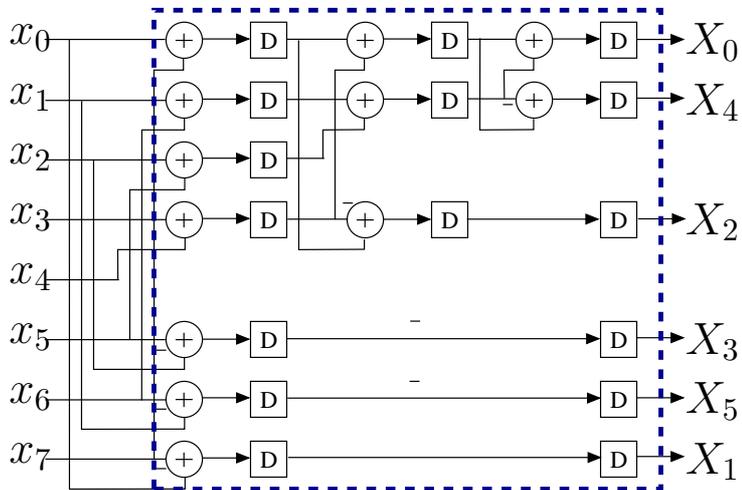


Figura 5.16: Arquitetura 1-D para a MRDCT podada.

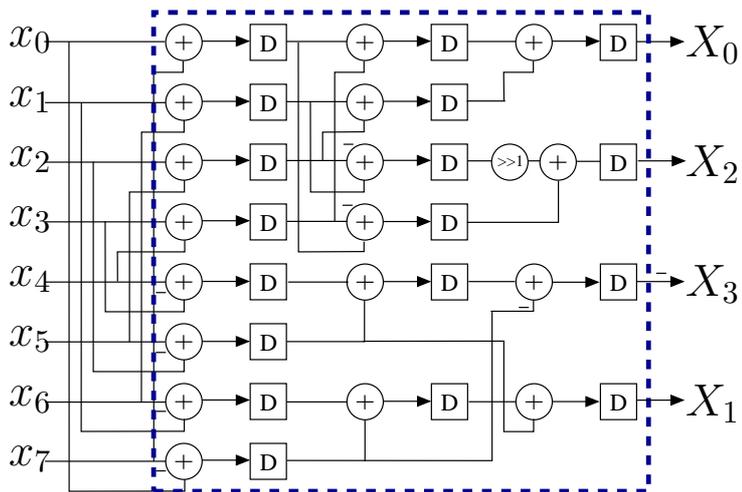


Figura 5.17: Arquitetura 1-D para a LODCT podada.

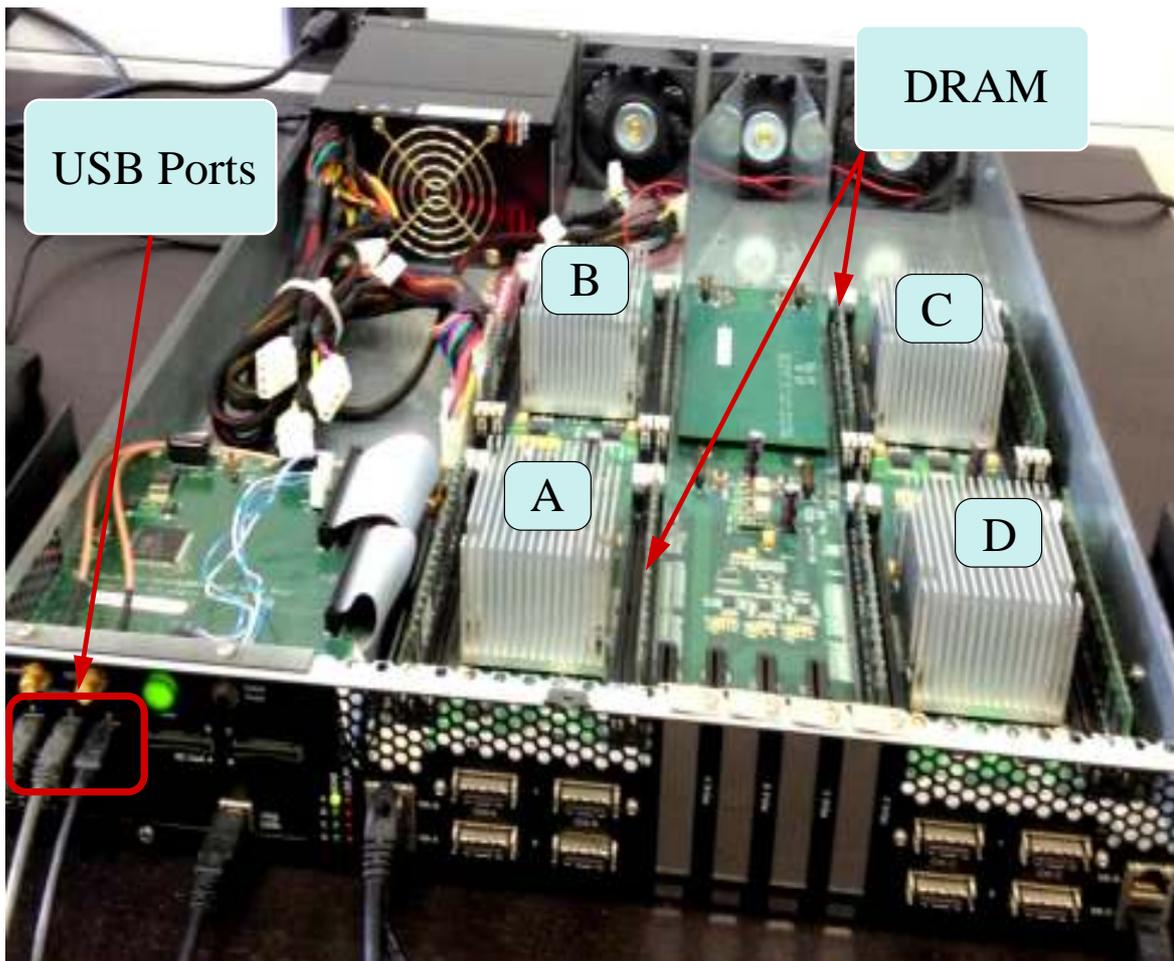


Figura 5.18: Dispositivo BEE3 utilizado para implementar as arquiteturas. Os marcadores A, B, C e D indicam as quatro FPGAs Xilinx Virtex-5 XC5VSX95T-2FF1136.

CAPÍTULO 6

CONCLUSÕES

NESTE trabalho, foram apresentadas diversas aproximações para a DCT baseadas em *pruning*, ou aproximações podadas. Tais aproximações são derivadas de aproximações já existentes na literatura. Os novos métodos podem ser considerados em diversas aplicações em *hardware* com baixo consumo energético, tais como transmissão de vídeo em tempo real, armazenamento/transmissão de imagem e vídeo através de dispositivos portáteis, compressão de imagem em rede de sensores *wireless*, compressão de imagem para aplicações médicas, como endoscopia *wireless*, entre outros. Os novos métodos derivados proporcionam um maior ajuste fino no compromisso entre custo computacional e desempenho. Os métodos apresentaram desempenho satisfatório e complexidades aritméticas bastante reduzidas.

No Capítulo 2, foram apresentadas questões teóricas relativas a DCT. As definições formais foram mostradas, assim como a relação da DCT com a KLT. Diversos algoritmos rápidos populares para a DCT foram descritos e comparados.

No Capítulo 3, foram abordadas as aproximações para a DCT. Os conceitos de ortogonalidade e ortonormalidade foram apresentados, assim como o procedimento matemático de ortonormalização. Diferentes métodos encontrados na literatura foram catalogados e comparados segundo as métricas descritas neste Capítulo.

As principais contribuições deste trabalho foram desenvolvidas nos Capítulo 4 e Capítulo 5. No Capítulo 4, foi proposta a definição generalizada de *pruning*, ou transformação podada. Foram descritas a DFT e a DCT exata podadas. Um estudo sobre a relação entre a complexidade aritmética de transformações em 1-D e 2-D considerando a decomposição linha-coluna foi apresentado. Foi verificado que as aproximações podadas apresentam uma redução de complexidade ainda maior para o caso 2-D. Um caso particular do conceito geral de transformação podada foi desenvolvido para o caso de aproximações para a DCT. Diversos métodos baseados nas aproximações descritas no Capítulo 3 foram derivados. Um procedimento para o cômputo da transformação inversa baseado na aproximação pela transformada inversa não podada foi desenvolvido. A definição de pseudo-ortonormalidade foi proposta. Foi mostrado que para métodos pseudo-ortogonais, computar a transformada inversa através da matriz generalizada de Moore-Penrose resulta na mesma expressão matemática obtida através da aproximação pela inversa não podada. Além disso, foi verificado computacionalmente que todos os métodos ortonormais abordados produzem métodos podados pseudo-ortonormais.

No Capítulo 5, foram abordadas questões introdutórias sobre processamento de imagens, com foco em compressão de imagem. As figuras de mérito MSE, PSNR e SSIM foram descritas.

O procedimento geral de compressão de imagem foi apresentado. Um estudo de distribuição de energia no domínio da transformada foi realizado utilizando cinquenta imagens de um banco público de imagens. Para tal, foram definidos o percentual relativo de energia, a energia relativa em dB e a energia relativa por zona. Através desse estudo, pode-se observar a propriedade de compactação de energia para cada uma das aproximações para a DCT abordadas. Simulações de compressão de imagem utilizando as imagens do banco público e as aproximações podadas derivadas no Capítulo 4 foram realizadas. O desempenho de compressão foi avaliada segundo as figuras de mérito. As curvas obtidas mostraram que as aproximações podadas apresentam desempenho competitivo com as aproximações originais não podadas e também com a DCT exata. Por apresentarem uma redução de complexidade ainda mais significativa em 2-D, as aproximações podadas se mostraram de relevante utilidade para a redução do consumo energético para aplicações *wireless* e de baixa potência. Alguns dos métodos podados propostos foram realizados fisicamente utilizando tecnologia CMOS, em parceria com a Universidade de Akron.

As contribuições do presente trabalho são sumarizadas a seguir:

- É apresentada uma revisão teórica sobre a DCT, catalogando, descrevendo e comparando diversos algoritmos rápidos;
- Conceitos teóricos a respeito de aproximações para a DCT são abordados. Diferentes métodos existentes na literatura são descritos, analisados e comparados. Medidas de desempenho são abordadas;
- Uma definição generalizada de *pruning* é proposta;
- Uma análise da complexidade aritmética para transformações em 2-D e 1-D é apresentada;
- Casos particulares da definição de *pruning* voltados para aproximações para a DCT são propostos;
- Um conjunto extenso de novos métodos aproximados e podados é proposto. Algoritmos rápidos para cada método são apresentados. Uma comparação entre a complexidade aritmética de cada método é realizada;
- Uma análise algébrica é realizada a respeito da transformação inversa para métodos podados. Métodos aproximados e baseados em matrizes inversas generalizadas para realizar a transformação inversa são propostos;
- Uma introdução a processamento e compressão de imagens é fornecida. Métricas para

avaliar o desempenho de compressão também são abordadas;

- Um estudo visando analisar a propriedade de compactação de energia de cada método aproximado é realizado;
- Simulações de compressão de imagem utilizando cada novo método proposto são realizadas. Os métodos podados são comparados segundo as métricas abordadas. Os resultados mostraram que, enquanto a complexidade aritmética é reduzida consideravelmente, métodos podados proporcionam desempenho competitivo. Conclui-se que considerar métodos podados é uma alternativa relevante para aplicações de baixo consumo de potência.
- Desenvolvimento de arquitetura digital de alguns dos métodos podados propostos e implementação em VLSI (parceria com a Universidade de Akron).

Como trabalhos futuros, sugere-se:

- Realizar estudo visando o desenvolvimento de algoritmos rápidos para aproximações para a DCT diretamente em duas dimensões. Assim como no caso da DCT exata, é possível que tais algoritmos apresentem complexidade aritmética ainda menores do que utilizando a decomposição linha-coluna;
- Aplicar a abordagem *pruning* para algoritmos rápidos diretamente em duas dimensões;
- Desenvolver o conceito de *pruning* para a DCT de três dimensões, visando codificação de vídeo;
- Derivar aproximações para a DCT em três dimensões, com objetivo de gerar métodos de codificação de vídeo de baixo custo computacional;
- Considerar a abordagem *pruning* para DCTs aproximadas em três dimensões, uma vez que tais aproximações sejam propostas;
- Verificar a veracidade ou não veracidade da seguinte conjectura: *Seja uma matriz de transformação \mathbf{T} . Seja a matriz podada $\mathbf{T}_{\langle K \rangle}$, obtida a partir de \mathbf{T} de acordo com a Definição 4.4. Se \mathbf{T} é ortonormal, então $\mathbf{T}_{\langle K \rangle}$ é pseudo-ortonormal.*
- Aplicar os métodos podados propostos em CODEC de vídeo x264 *open source*.

CAPÍTULO 7

BIBLIOGRAFIA

- [1] N. Ahmed and K. R. Rao. *Orthogonal Transforms for Digital Signal Processing*. Springer, 1975.
- [2] N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, C-23:90–93, January 1974.
- [3] K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, San Diego, CA, 1990.
- [4] V. Britanak, P. Yip, and K. R. Rao. *Discrete Cosine and Sine Transforms*. Academic Press, 2007.
- [5] G. K. Wallace. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38:xviii–xxxiv, 1992.
- [6] International Organisation for Standardisation. Generic coding of moving pictures and associated audio information – Part 2: Video. ISO/IEC JTC1/SC29/WG11 - coding of moving pictures and audio, ISO, 1994.
- [7] N. Roma and L. Sousa. Efficient hybrid DCT-domain algorithm for video spatial downscaling. *EURASIP Journal on Advances in Signal Processing*, 2007:30–30, 2007.
- [8] International Telecommunication Union. ITU-T recommendation H.261 version 1: Video codec for audiovisual services at $p \times 64$ kbits. Technical report, ITU-T, 1990.

- [9] International Telecommunication Union. ITU-T recommendation H.263 version 1: Video coding for low bit rate communication. Technical report, ITU-T, 1995.
- [10] Joint Video Team. Recommendation H.264 and ISO/IEC 14 496–10 AVC: Draft ITU-T recommendation and final draft international standard of joint video specification. Technical report, ITU-T, 2003.
- [11] A. Luthra, G. J. Sullivan, and T. Wiegand. Introduction to the special issue on the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:557–559, July 2003.
- [12] International Telecommunication Union. High efficiency video coding: Recommendation ITU-T H.265. Technical report, ITU-T Series H: Audiovisual and Multimedia Systems, 2013.
- [13] M. T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos. HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC? *IEEE Consumer Electronics Magazine*, 1:36–46, July 2012.
- [14] W. H. Chen, C. Smith, and S. Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on Communications*, 25:1004–1009, September 1977.
- [15] B. G. Lee. A new algorithm for computing the discrete cosine transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-32:1243–1245, December 1984.
- [16] Z. Wang. Fast algorithms for the discrete W transform and for the discrete Fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-32:803–816, August 1984.
- [17] M. Vetterli and H. Nussbaumer. Simple FFT and DCT algorithms with reduced number of operations. *Signal Processing*, 6:267–278, August 1984.
- [18] N. Suehiro and M. Hateri. Fast algorithms for the DFT and other sinusoidal transforms. *IEEE Transactions on Acoustic, Signal, and Speech Processing*, 34:642–644, 1986.
- [19] H. S. Hou. A fast recursive algorithm for computing the discrete cosine transform. *IEEE Transactions on Acoustic, Signal, and Speech Processing*, 6:1455–1461, 1987.

- [20] C. Loeffler, A. Ligtenberg, and G. S. Moschytz. Practical fast 1-D DCT algorithms with 11 multiplications. *ICASSP International Conference on Acoustics, Speech, and Signal Processing*, 2:988–991, 1989.
- [21] E. Feig and S. Winograd. Fast algorithms for the discrete cosine transform. *IEEE Transactions on Signal Processing*, 40:2174–2193, 1992.
- [22] J. S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice Hall, 1990.
- [23] N. Cho. Fast algorithm and implementation of 2-D discrete cosine transform. *IEEE Transactions on Circuits and Systems*, 38:297 – 305, 1991.
- [24] M. J. Narasimha. Design of FIR filter banks for a 24-channel transmultiplexer. *IEEE Transactions on Communications*, 30:1506 – 1510, 1982.
- [25] W. H. Chen. Scene adaptive coder. *IEEE Transactions on Communications*, 32:225–232, 1984.
- [26] M. Nomura. Implementation of video codec with programmable parallel DSP. *Global Telecommunications Conference and Exhibition Communications Technology for the 1990s and Beyond, (GLOBECOM)*, pages 908 – 912, 1989.
- [27] A. M. Chiang. A video-rate CCD two-dimensional cosine transform processor. *Visual Communications and Image Processing II*, 0845, 1987.
- [28] B. Jian, Z. Xuan, T. J. Rong, and L. Yue. An efficient VLSI architecture for 2D-DCT using direct method. *International Conference on ASIC*, pages 393 – 396, 2001.
- [29] B. Sheng, W. Gao, and D. Wu. An implemented VLSI architecture of inverse quantizer for AVS HDTV video decoder. *6th International Conference On ASIC (ASICON)*, pages 244–247, 2005.
- [30] S. Winograd. *Arithmetic Complexity of Computations*. CBMS-NSF Regional Conference Series in Applied Mathematics, 1980.
- [31] M. T. Heideman. *Multiplicative complexity, convolution, and the DFT*. Signal Processing and Digital Filtering. Springer-Verlag, 1988.
- [32] R. J. Cintra, F. M. Bayer, and C. J. Tablada. Low-complexity 8-point DCT approximations based on integer functions. *Signal Processing*, 99:201–214, 2014.

- [33] T. I. Haweel. A new square wave transform based on the DCT. *Signal Processing*, 82:2309–2319, 2001.
- [34] K. Lengwehasatit and A. Ortega. Scalable variable complexity approximate forward DCT. *IEEE Transactions on Circuits and Systems for Video Technology*, 14:1236–1248, November 2004.
- [35] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy. A multiplication-free transform for image compression. In *2nd International Conference on Signals, Circuits and Systems (SCS)*, pages 1–4, November 2008.
- [36] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy. Low-complexity 8×8 transform for image compression. *Electronics Letters*, 44:1249–1250, September 2008.
- [37] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy. A fast 8×8 transform for image compression. In *2009 International Conference on Microelectronics (ICM)*, pages 74–77, December 2009.
- [38] F. M. Bayer and R. J. Cintra. Image compression via a fast DCT approximation. *IEEE Latin America Transactions*, 8:708–713, December 2010.
- [39] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy. A novel transform for image compression. In *53rd IEEE International Midwest Symposium on Circuits and Systems (MWS-CAS)*, pages 509–512, August 2010.
- [40] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy. A low-complexity parametric transform for image compression. In *Proceedings of the 2011 IEEE International Symposium on Circuits and Systems*, 2011.
- [41] R. J. Cintra and F. M. Bayer. A DCT approximation for image compression. *IEEE Signal Processing Letters*, 18:579–582, October 2011.
- [42] F. M. Bayer and R. J. Cintra. DCT-like transform for image compression requires 14 additions only. *Electronics Letters*, 48:919–921, 2012.
- [43] S. Bouguezel, M. O. Ahmad, and M. N. S. Swamy. Binary discrete cosine and Hartley transforms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60:989–1002, 2013.

- [44] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya. Improved 8-point approximate DCT for image and video compression requiring only 14 additions. *IEEE Transactions on Circuits and Systems I*, PP:1–14, April 2014.
- [45] C. J. Tablada. Aproximações para a DCT baseadas nos algoritmos de feig-winograd e chen. Master’s thesis, Universidade Federal de Pernambuco, 2014.
- [46] C. J. Tablada, F. M. Bayer, and R. J. Cintra. A class of DCT approximations based on the Feig–Winograd algorithm. *Signal Processing*, 113:38–51, 2015.
- [47] Z. Wang. Pruning the fast discrete cosine transform. *IEEE Transactions on Communications*, 39:640–643, May 1991.
- [48] L. Makkaoui, V. Lecuire, and J. Moureaux. Fast zonal DCT-based image compression for wireless camera sensor networks. *2nd International Conference on Image Processing Theory Tools and Applications (IPTA)*, pages 126–129, 2010.
- [49] V. Lecuire, L. Makkaoui, and J.-M. Moureaux. Fast zonal DCT for energy conservation in wireless image sensor networks. *Electronics Letters*, 48:125–127, 2012.
- [50] N. Kouadria, N. Doghmane, D. Messadeg, and S. Harize. Low complexity DCT for image compression in wireless visual sensor networks. *Electronics Letters*, 49:1531–1532, 2013.
- [51] J. Markel. FFT pruning. *IEEE Transactions on Audio and Electroacoustics*, 19:305–311, 1971.
- [52] D. P. Skinner. Pruning the decimation in-time FFT algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 24:193–194, 1976.
- [53] M. Servais and G. de Jager. Video compression using the three dimensional discrete cosine transform (3D-DCT). *Proceedings of the South African Symposium on Communications and Signal Processing*, pages 27 – 32, 1997.
- [54] R. K. W. Chan and M. C. Lee. 3D-DCT quantization as a compression technique for video sequences. *Proceedings of the International Conference on Virtual Systems and MultiMedia*, pages 188 – 196, 1997.
- [55] S. Boussakta. Fast algorithm for the 3-D DCT-II. *IEEE Transactions on Signal Processing*, 52:992 – 1001, 2004.

- [56] S. Jakubczak, H. Rahul, and D. Katabi. SoftCast: one-size-fits-all wireless video. *ACM SIGCOMM Computer Communication Review*, 40:449–450, 2010.
- [57] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:103–108, 1990.
- [58] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2:559–572, 1901.
- [59] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.
- [60] K. R. Rao and P. Yip. *The Transform and Data Compression Handbook*. CRC Press LLC, 2001.
- [61] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, volume 1 of *Prentice Hall Signal Processing Series*. Prentice Hall, Upper Saddle River, NJ, 1993.
- [62] B. Kolman and D. R. Hill. *Elementary Linear Algebra with Applications*. Pearson, 9a. edição, 2008.
- [63] W. D. Davenport and W. L. Root. *An Introduction to the Theory of Random Signals and Noise*. McGraw-Hill, 1958.
- [64] A. Oppenheim and R. Schaffer. *Discrete-Time Signal Processing*. Pearson, 3a. edição, 2010.
- [65] S. A. Martucci. Symmetric convolution and the discrete sine and cosine transforms. *IEEE Transactions on Signal Processing*, 42:1038–1051, 1994.
- [66] Dennis S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas*. Princeton University Press, 2009.
- [67] J. Liang and T. D. Tran. Fast multiplierless approximation of the DCT with the lifting scheme. *IEEE Transactions on Signal Processing*, 49:3032–3044, 2001.
- [68] Richard E. Blahut. *Fast Algorithms for Signal Processing*. Cambridge University Press, 2010.

- [69] T. R. Harley and G. P. Maheshwaramurthy. Address generators for mapping arrays in bit-reversed order. *IEEE Transactions on Signal Processing*, 52:1693–1703, 2004.
- [70] H. S. Hou. A fast recursive algorithm for computing the discrete cosine transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35:1455 – 1461, 1987.
- [71] R. Xuejun and F. Dingyi. A sensor node compression algorithm based on H.264 integer DCT transform in WSN. *IET International Conference on Wireless Sensor Network - IET-WSN*, pages 214–217, 2010.
- [72] P. Turcza and M. Duplaga. Low-power image compression for wireless capsule endoscopy. *IEEE International Workshop on Imaging Systems and Techniques - IST*, pages 1–4, 2007.
- [73] M.E. Rizkalla, P. M. Ei-Sharkawy Salama, and B. Dukel. Implementation of floating point fast discrete cosine transform. *The 2002 45th Midwest Symposium on Circuits and Systems - MWSCAS.*, 2:II–17 – II–20, 2002.
- [74] S. Kim and W. Sung. A floating-point to fixed-point assembly program translator for the TMS 320C25. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 41:730 – 739, 1994.
- [75] M. M. Campello de Souza, H. M. de Oliveira, R. M. Campello de Souza, and M. M. Vasconcelos. The discrete cosine transform over prime finite fields. *Telecommunications and Networking - ICT*, 3124:482–487, 2004.
- [76] J. B. Lima and R. M. Campello de Souza. Finite field trigonometric transforms. *Applicable Algebra in Engineering, Communication and Computing*, 22:3930411, 2011.
- [77] S.-F. Hsiao, Y. H. Hu, T.-B. Juang, and C.-H. Lee. Efficient VLSI implementations of fast multiplierless approximated DCT using parameterized hardware modules for silicon intellectual property design. *IEEE Transactions on Circuits and Systems I*, 52:1568 – 1579, 2005.
- [78] M.-H. Lee. CSD filter design for VLSI implementation of GA-VSB receiver. *IEEE Transactions on Consumer Electronics*, 43:197–206, 1997.
- [79] T. Suzuki and M. Ikehara. Integer DCT based on direct-lifting of DCT-IDCT for lossless-to-lossy image coding. *IEEE Transactions on Image Processing*, 19:2958–2965, November 2010.

- [80] T. Suzuki, Y. Tanaka, M. Ikehara, and H. Aso. Multiplierless fast algorithm for DCT via fast Hartley transform. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3469 – 3472, 2012.
- [81] N. J. Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7:1160–1174, October 1986.
- [82] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur. Computing the polar decomposition and the matrix sign decomposition in matrix groups. *SIAM J. Matrix Analysis Applications*, 25:1178–1192, 2004.
- [83] R. J. Cintra. An integer approximation method for discrete sinusoidal transforms. *Journal of Circuits, Systems, and Signal Processing*, 30:1481–1501, 2011.
- [84] N. J. Higham. Computing real square roots of a real matrix. *Linear Algebra and its Applications*, 88–89:405–430, 1987.
- [85] V. Bhaskaran and K. Konstantinides. *Image and Video Compression Standards*. Kluwer Academic Publishers, 1997.
- [86] I. E. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley & Sons, 2004.
- [87] D. F. Elliot and K. R. Rao. *Fast Transforms: Algorithms, Analyses, Applications*. Academic Press, 1982.
- [88] S. C. Knauer. Real-time video compression algorithm for Hadamard transform processing. *IEEE Transactions on Electromagnetic Compatibility*, ECM-18:28–36, 1976.
- [89] MathWorks. *MATLAB*. The MathWorks, Inc., Natick, MA, 2011.
- [90] S. Cao, Z. Ye, D. Xu, and X. Xu. A Hadamard product based method for DOA estimation and gain-phase error calibration. *IEEE Transactions on Aerospace and Electronic Systems*, 49:1224–1233, 2013.
- [91] M. G. Kendall. *The Advanced Theory of Statistics*, volume 1. Charles Griffin & Company Limited, 2 edition, 1945.
- [92] Jae-Hyung Kim, Jun-Gu Kim, Young-Hyok Ji, Yong-Chae Jung, and Chung-Yuen Won. An islanding detection method for a grid-connected system based on the Goertzel algorithm. *IEEE Transactions on Power Electronics*, 26:1049–1055, April 2011.

- [93] G. J. da Silva and R. M. C. de Souza. Minimum multiplicative complexity algorithm for computing a single component of the discrete Fourier transform. *Digital Signal Processing*, 23:1040–1043, 2013.
- [94] I. Carugati, S. Maestri, P. G. Donato, D. Carrica, and M. Benedetti. Variable sampling period filter PLL for distorted three-phase systems. *IEEE Transactions on Power Electronics*, 27:321–330, January 2012.
- [95] R. G. Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, 1968.
- [96] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, April 2004.
- [97] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Springer Science and Business Media New York. 8a. edição, 2001.
- [98] Signal and Image Processing Institute. USC-SIPI Image Database. <http://sipi.use.edu/database/>.
- [99] BEEcube Inc. Berkeley Emulation Engine. <http://www.beecube.com/>.

APÊNDICE A

ALGORITMOS RÁPIDOS PARA AS APROXIMAÇÕES PODADAS

Tabela A.1: Associação entre métodos propostos, figuras e parâmetros de algoritmos rápidos

Métodos podados	Figura	Parâmetros
SDCT	A.1	—
BAS-1	A.2	—
BAS-2	A.3	—
BAS-3	A.4	—
BAS-4	A.5	$a = 0$
BAS-5	A.5	$a = 1/2$
BAS-6	A.5	$a = 1$
BAS-7	A.6	—
LODCT	A.9 e A.10	$\mathbf{m} = [1 \ 1 \ 1 \ 1 \ 1 \ \frac{1}{2} \ 0]$
RDCT	A.7	—
MRDCT	A.8	—
INT-1	A.9 e A.10	$\mathbf{m} = [1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0]$
INT-2	A.9 e A.10	$\mathbf{m} = [2 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0]$
INT-4	A.9 e A.10	$\mathbf{m} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0]$
INT-5	A.9 e A.10	$\mathbf{m} = [2 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0]$
INT-6	A.9 e A.10	$\mathbf{m} = [2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 0]$

A seguir são apresentados os algoritmos rápidos para todos os métodos podados propostos. A Tabela A.1 associa os métodos descritos com as respectivas figuras e com os respectivos parâmetros para algoritmo rápido, quando existentes na Figura. Define-se $\mathbf{m} = [m_0 \ m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6]$, parâmetros utilizados nas Figura A.9 e Figura A.10.

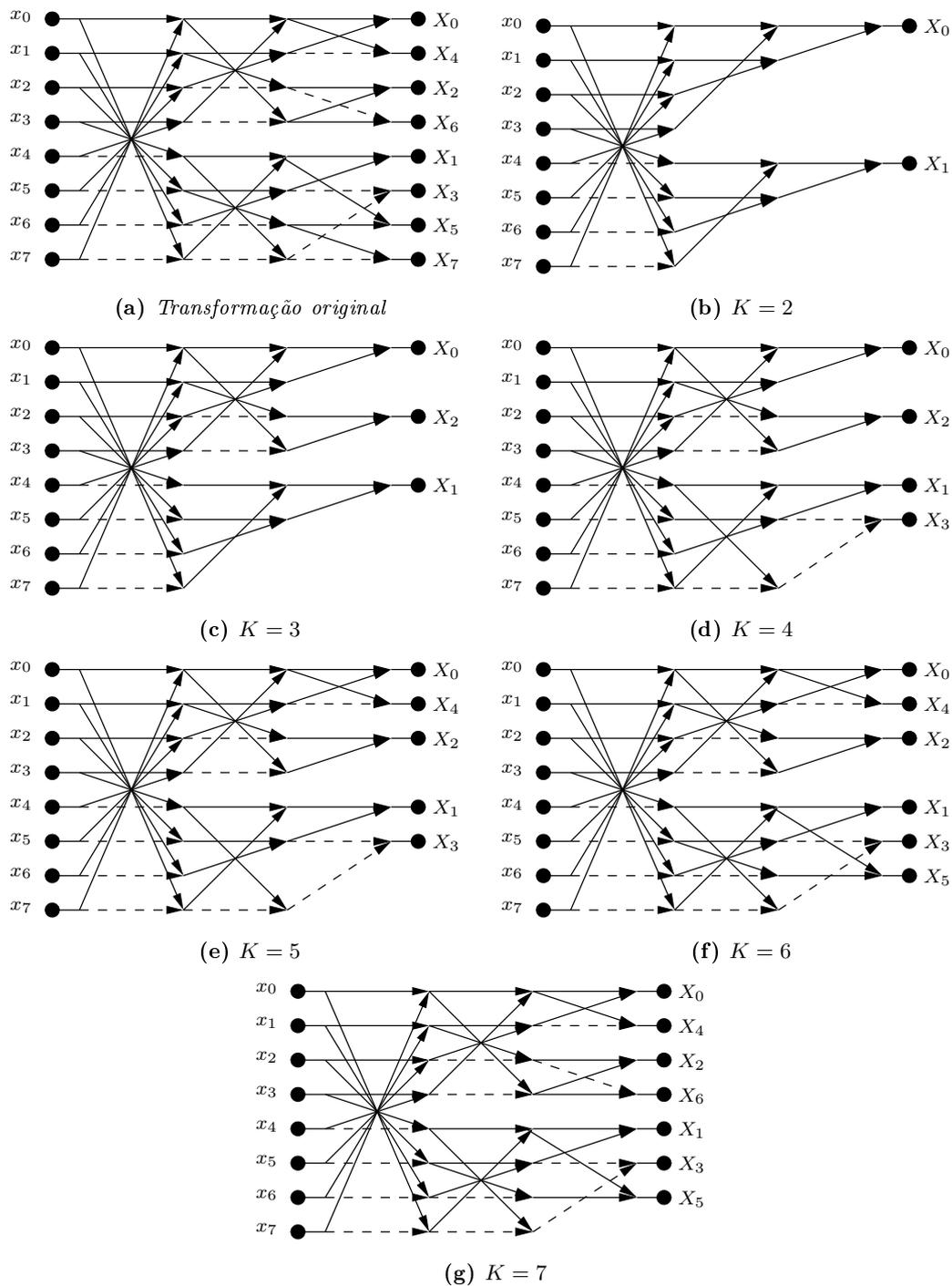


Figura A.1: Algoritmos rápidos para aproximações derivadas da SDCT.

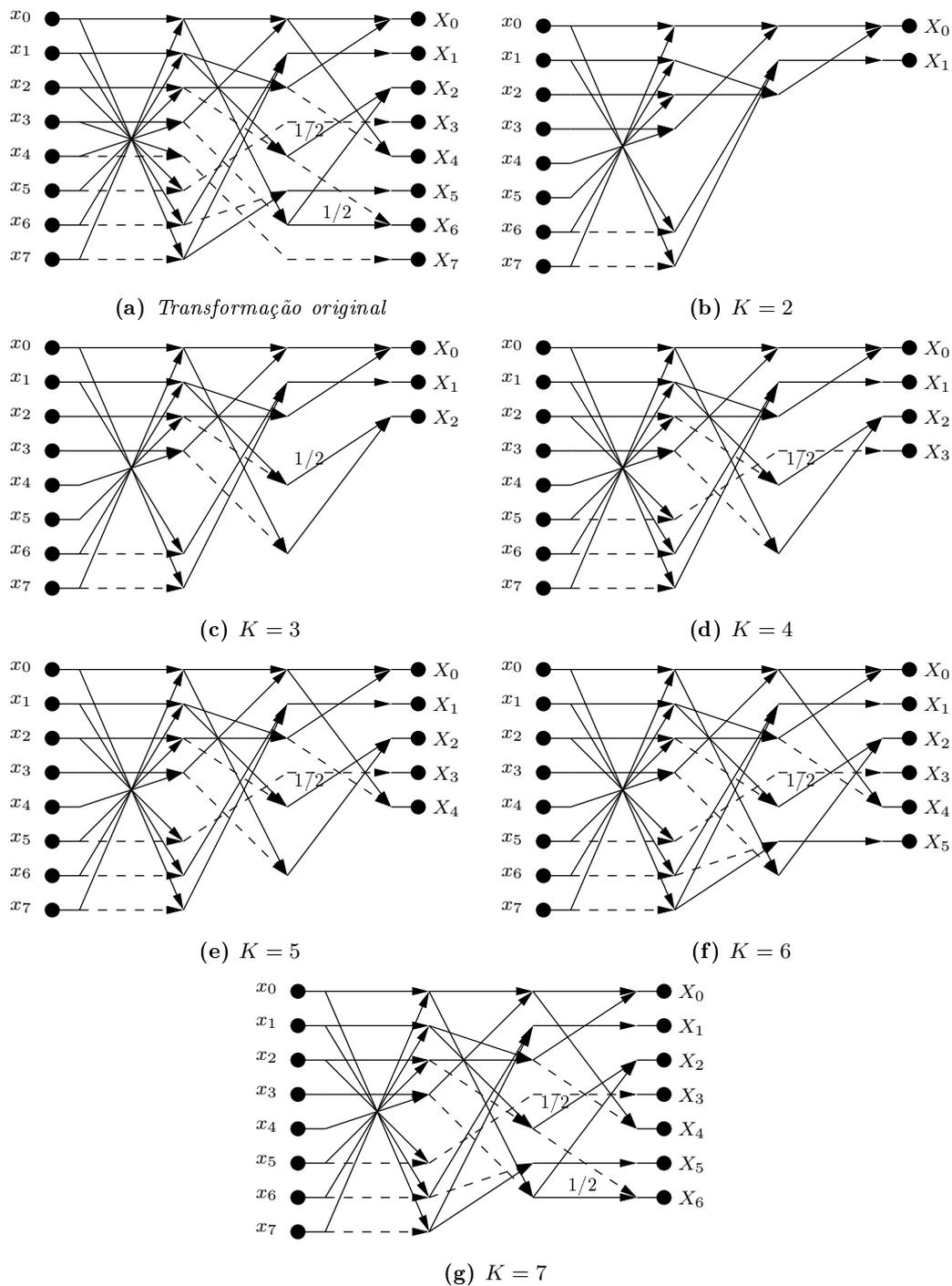


Figura A.2: Algoritmos rápidos para aproximações derivadas de BAS-1.

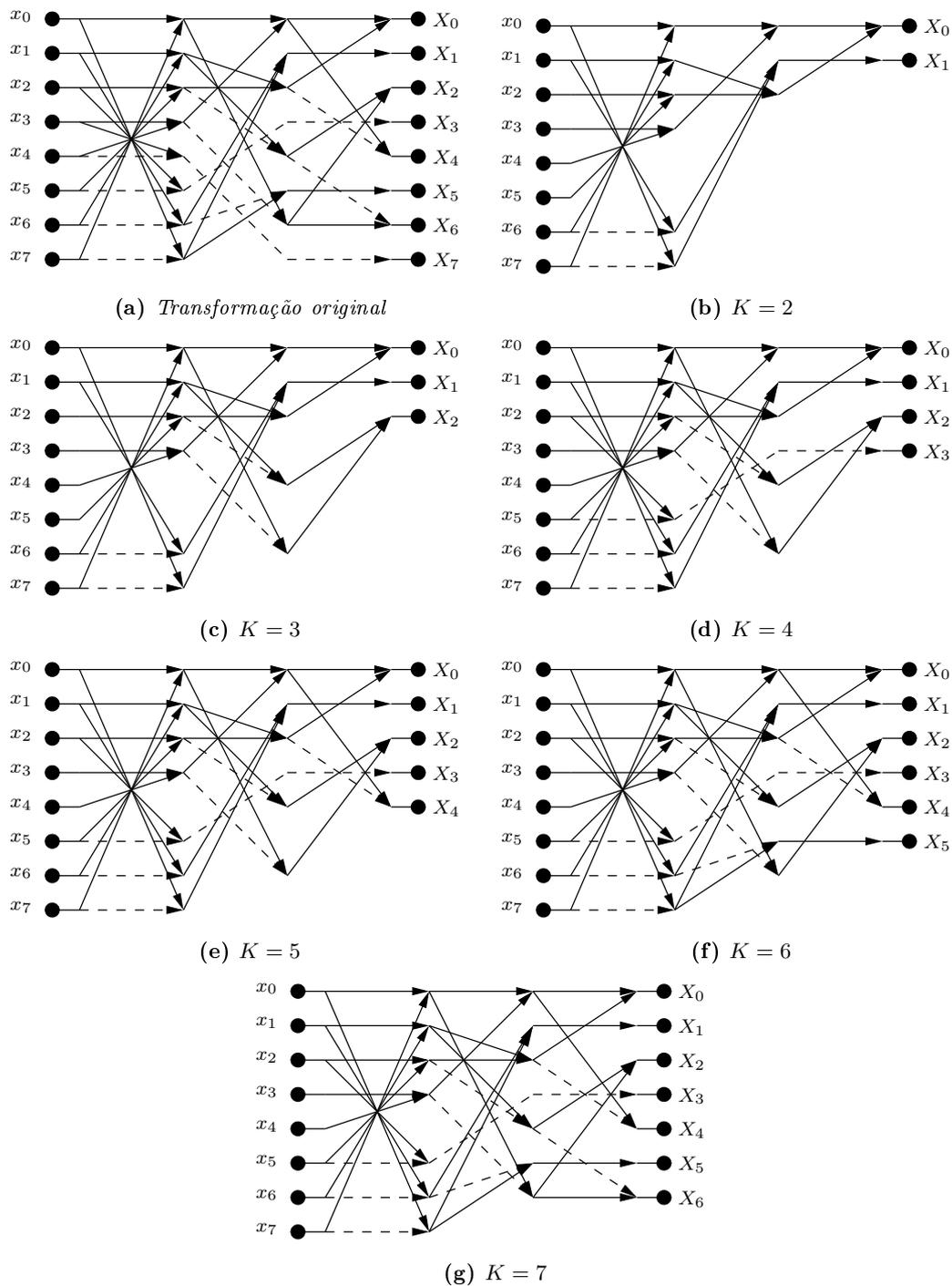


Figura A.3: Algoritmos rápidos para aproximações derivadas de BAS-2.

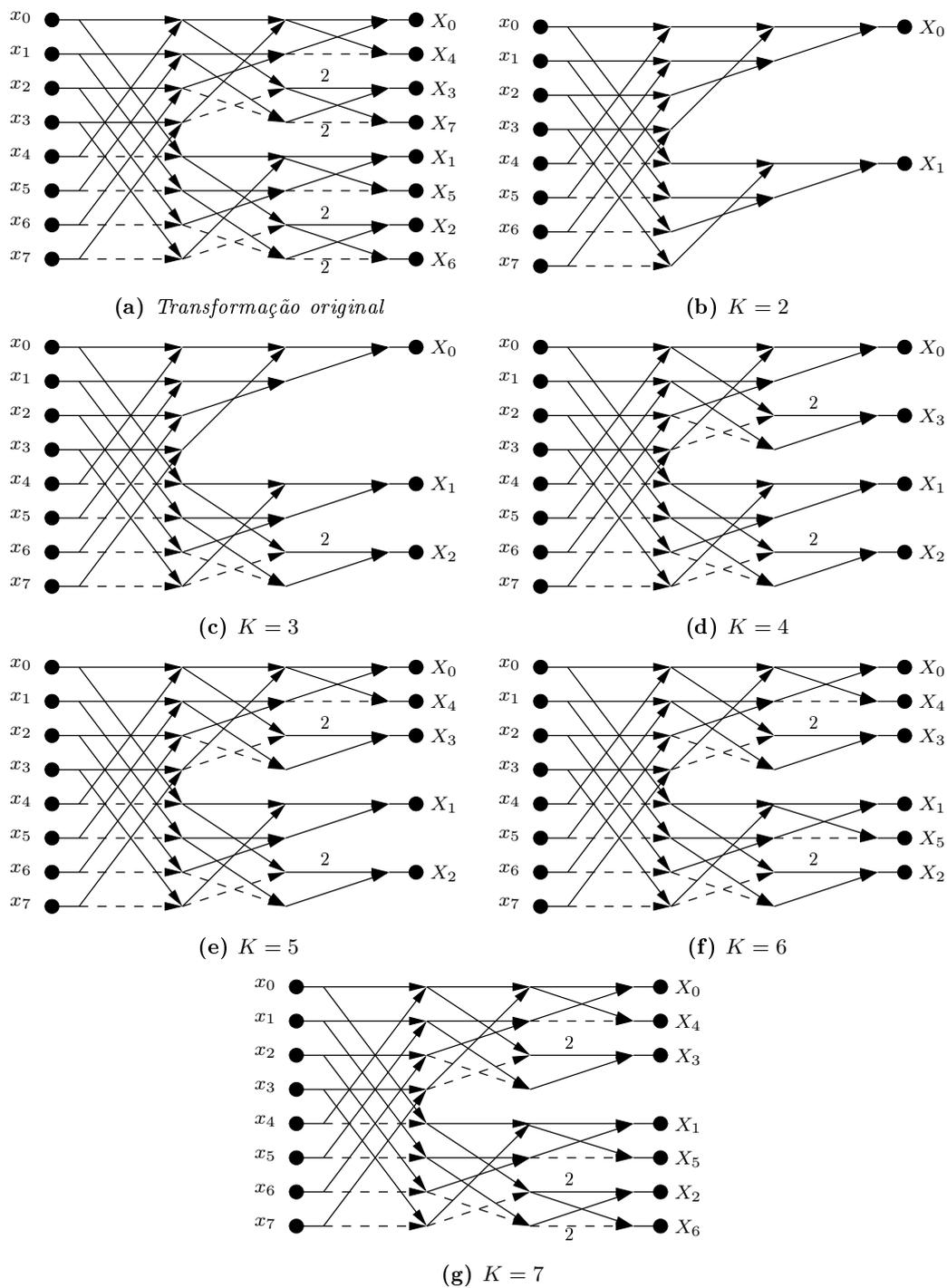


Figura A.4: Algoritmos rápidos para aproximações derivadas de BAS-3.

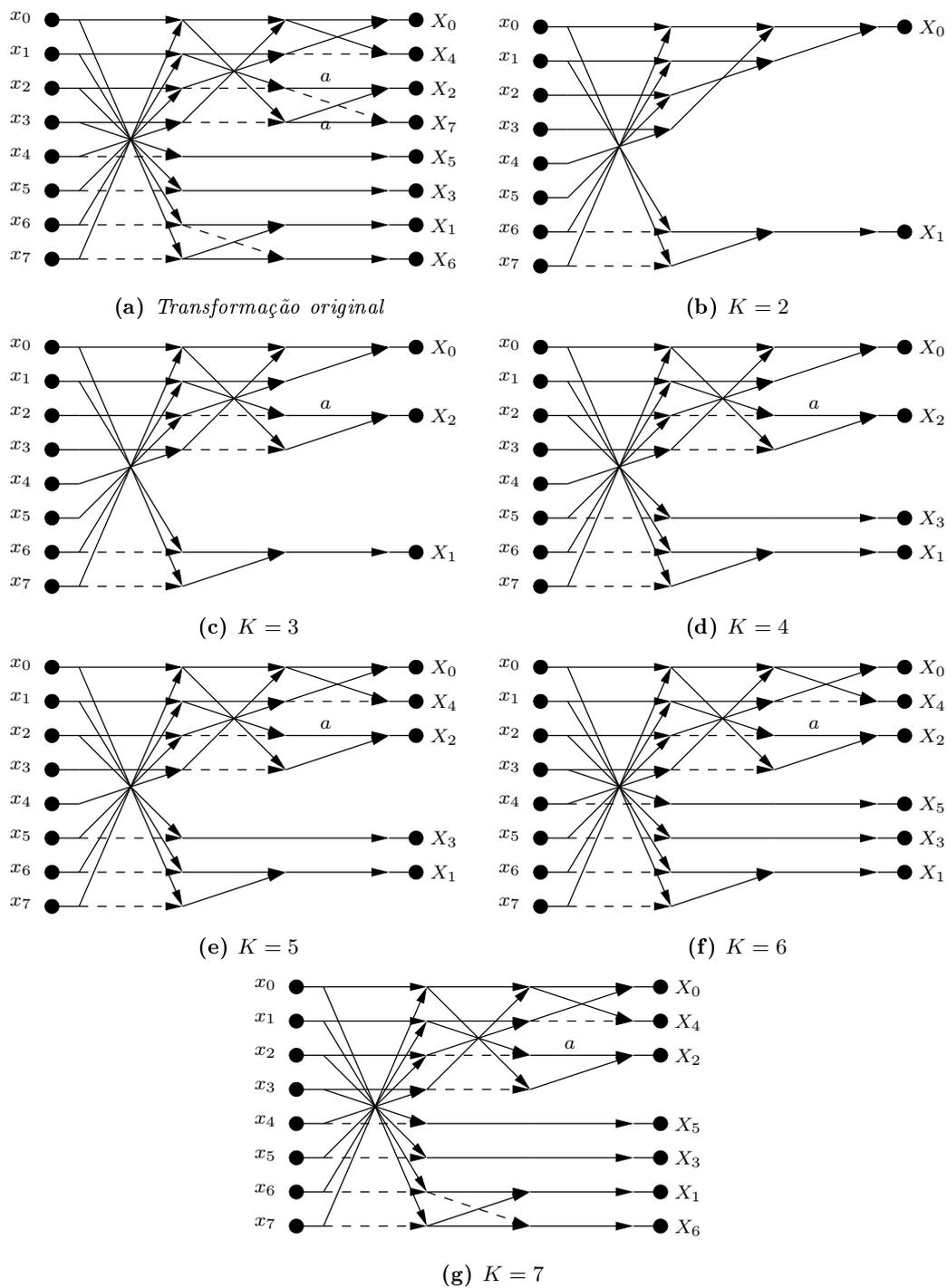


Figura A.5: Algoritmos rápidos para aproximações derivadas de BAS-4, BAS-5 e BAS-6.

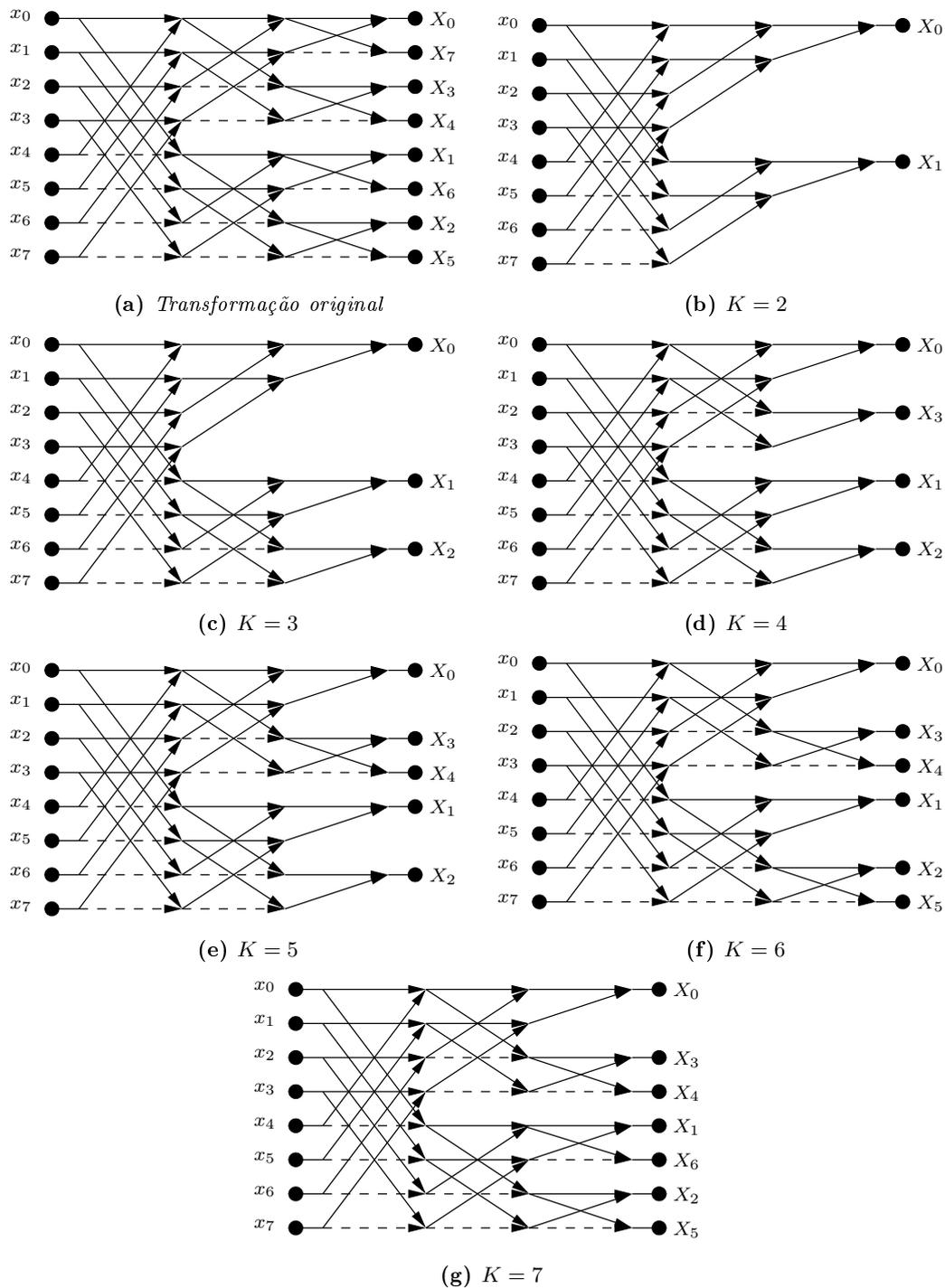


Figura A.6: Algoritmos rápidos para aproximações derivadas de BAS-7.

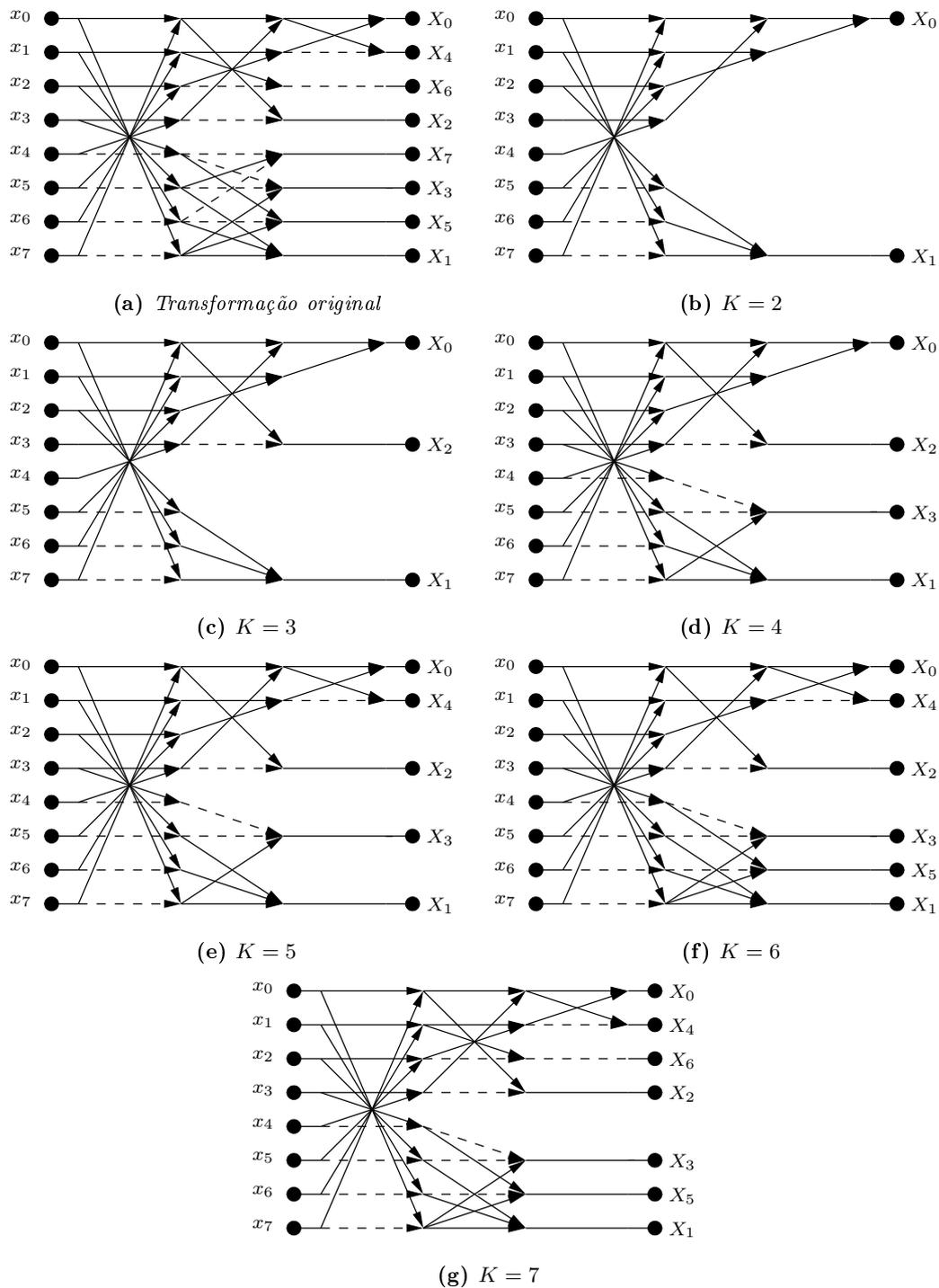


Figura A.7: Algoritmos rápidos para aproximações derivadas da RDCT.

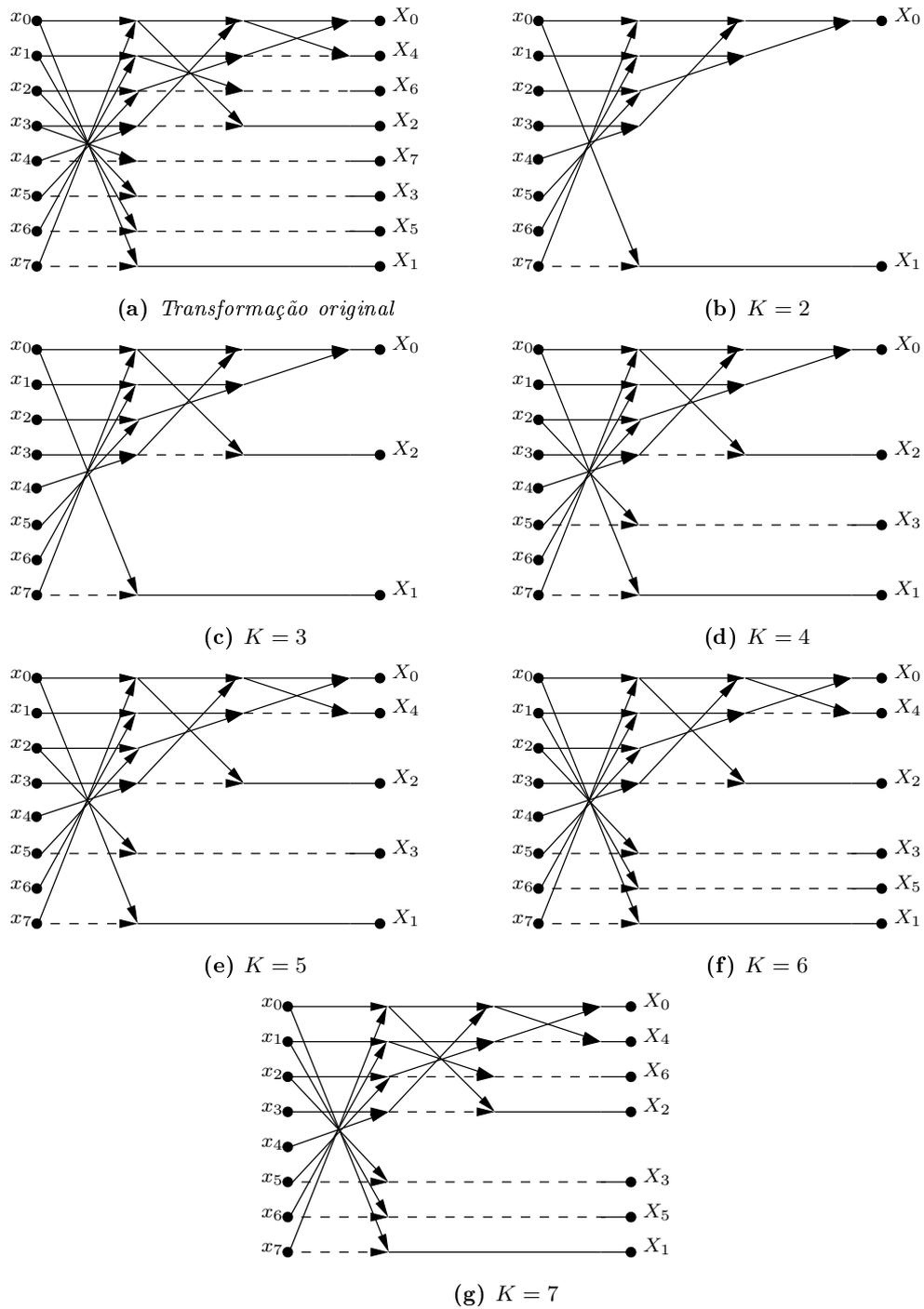


Figura A.8: Algoritmos rápidos para aproximações derivadas da MRDCT.

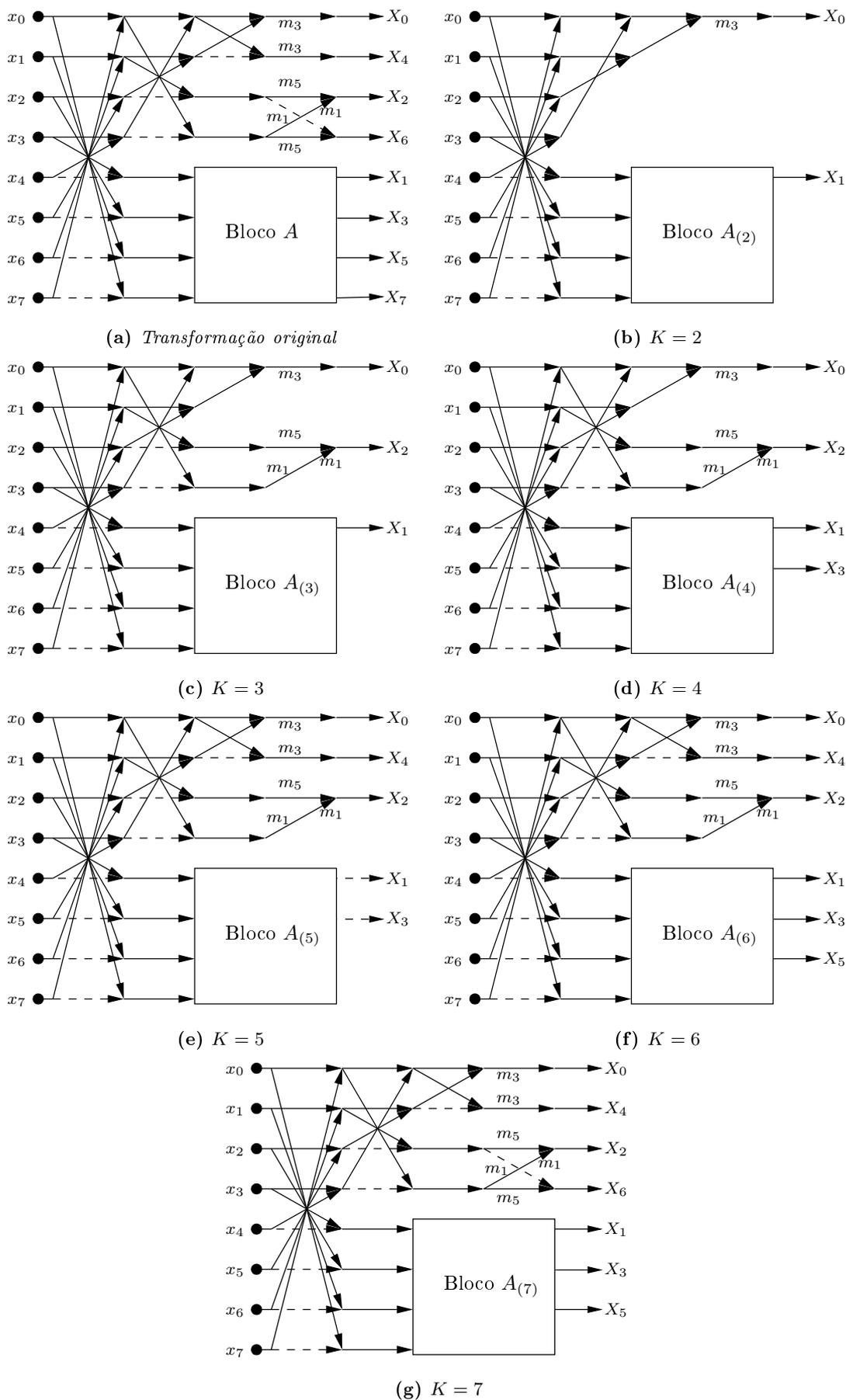


Figura A.9: Algoritmos rápidos para aproximações derivadas da LODCT e das INT-*i*.

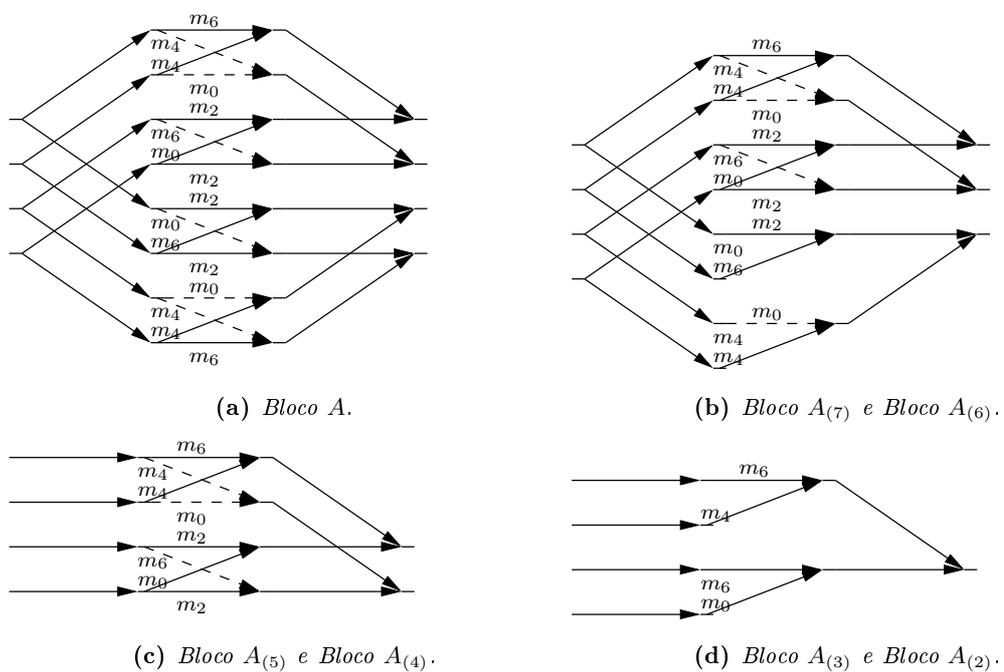


Figura A.10: Blocos $A_{(i)}$ para os algoritmos da Figura A.9.

APÊNDICE B

IMAGENS UTILIZADAS NAS SIMULAÇÕES



Figura B.1: *Imagens aplicadas ao procedimento de compressão de imagens.*

APÊNDICE C

TRABALHOS REALIZADOS

Os desenvolvimentos do presente trabalho geraram as seguintes contribuições:

- Artigos publicados:

1. V. A. Coutinho, R. J. Cintra, F. M. Bayer, S. Kulasekera and A. Madanayake. *A multiplierless pruned DCT-like transformation for image and video compression that requires ten additions only*. Journal of Real-Time Image Processing (JRTIP). Special Issue: HEVC for Real-Time Video Applications. 2015. DOI: 10.1007/s11554-015-0492-8.
2. V. A. Coutinho, R. J. Cintra, F. M. Bayer, S. Kulasekera and A. Madanayake. *Low-complexity Pruned 8-point DCT Approximations for Image Encoding*. Proceedings of IEEE International Conference on Electronics, Communications and Computers - CONIELECOMP 2015. DOI: 10.1109/CONIELECOMP.2015.7086923.

- Artigos submetidos:

1. F. M. Bayer, R. J. Cintra, V. A. Coutinho, S. Kulasekera and A. Madanayake. *Energy-efficient 8-point DCT Approximations: Theory and Hardware Architectures*. Circuits, Systems & Signal Processing (CSSP);

- Patente:

1. V. A. Coutinho, R. J. Cintra, F. M. Bayer. *Conjuntos de Métodos de Baixo Consumo Para Compressão de Imagem e Vídeo*. BR 10 2015 003660 4.

GLOSSÁRIO DE NOTAÇÃO

\mathbb{P}

conjunto $\{0, 1, 2, \frac{1}{2}\}$

\mathbb{N}

conjunto dos números naturais

\mathbb{Z}

conjunto dos números inteiros

\mathbb{Q}

conjunto dos números racionais

\mathbb{R}

conjunto dos números reais

\mathbb{C}

conjunto dos números complexos

\mathbb{C}^N

espaço vetorial de dimensão N com elementos em \mathbb{C}

\top

transposição matricial

\otimes

produto tensorial

\odot

produto matricial de Hadamard

$M(\cdot)$

complexidade multiplicativa

$A(\cdot)$

complexidade aditiva

$B(\cdot)$

complexidade de deslocamento de *bits*

$\mathcal{A}(\cdot)$

complexidade aritmética

$E[\cdot]$

retorna valor esperado ou esperança

$\text{tr}(\cdot)$

retorna traço de uma matriz quadrada

$\text{diag}(\cdot)$

retorna uma matriz bloco-diagonal cujos elementos da diagonal principal são definidos, em ordem, pelo argumento

$\text{diag}_{\mathbf{M}}(\cdot)$

retorna uma matriz diagonal cujos elementos são dados pela diagonal principal do argumento

$\text{diag}_{\mathbf{v}}(\cdot)$

retorna um vetor coluna cujos elementos são dados pela diagonal principal do argumento

$\text{sum}(\cdot)$

retorna a soma dos elementos da matriz de seu argumento

$\text{rank}(\cdot)$

retorna o *rank* de uma matriz passada no argumento

$\text{round}(\cdot)$

função de arredondamento usual

$(\cdot)_{10}$

representação em base binária

$(\cdot)_2$

representação em base decimal

C_g

ganho de codificação

 C_g^*

ganho de codificação unificado

 W_N

$$e^{-j\frac{2\pi}{N}}$$

 η

eficiência da transformada

 \mathbf{I}_N matriz identidade de dimensão $N \times N$ $\bar{\mathbf{I}}_N$ matriz de dimensão $N \times N$ cuja diagonal secundária é composta por elementos unitários e os demais elementos são nulos $\mathbf{T}_{(K)}$ matriz de dimensão $K \times N$ derivada de uma matriz arbitrária \mathbf{T} de dimensão $N \times N$, $K < N$, em que as primeiras K linhas são preservadas $\mathbf{D}_{[K]}$ matriz de dimensão $K \times K$ derivada de uma matriz arbitrária \mathbf{D} de dimensão $N \times N$, $K < N$, em que as primeiras K linhas e primeiras K colunas são preservadas \mathbf{H}^+ inversa generalizada de Moore-Penrose de uma matriz \mathbf{H}

ACRÔNIMOS

KLT	Transformada de Karhunen-Loève
DCT	Transformada discreta do cosseno
DCT-I	DCT tipo I
DCT-II	DCT tipo II
DCT-III	DCT tipo III
DCT-IV	DCT tipo IV
IDCT	DCT inversa
DST	Transformada discreta do seno
DFT	Transformada discreta de Fourier
DHT	Transformada discreta de Hartley
DWT	Transformada W discreta
WHT	Transformada de Walsh-Hadamard
FFT	Transformada rápida de Fourier
FCT	Transformada rápida do cosseno
DIT	Dizimação no tempo
SDCT	DCT sinalizada
LODCT	Lengwehasatit-Ortega DCT
BAS	Série de aproximações Bouguezal–Ahmad–Swamy
RDCT	DCT arredondada
MRDCT	RDCT modificada
INT	Referência às aproximações baseadas em funções inteiras

HDTV TV de alta definição

MSE Erro médio quadrático

SNR Relação sinal-ruído

PSNR Relação sinal-ruído de pico

SSIM Similaridade estrutural

VLSI *Very-large-scale integration*

CSD *Canonical-signed-digit*

RGB Sistema aditivo de cores formado por vermelho, verde e azul

HVS Sistema visual humano

FPGA *Field Programmable Gate Array*

CMOS Semicondutor metal-óxido complementar

BEE3 *Berkeley emulation engine*

PDK *Process design kit*

DSP Processador de sinais digitais

ASIC Circuito integrado de aplicação específica

WWSN Rede de sensores visuais *wireless*

CLB Bloco lógico configurável

FF *Flip-flop*