

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



DIEGO RAMOS CANTERLE

**NOVAS CLASSES
MULTIPARAMÉTRICAS DE
TRANSFORMADAS DISCRETAS COM
ALTO GANHO DE CODIFICAÇÃO E
EFICIÊNCIA**

RECIFE
2017

DIEGO RAMOS CANTERLE

**NOVAS CLASSES
MULTIPARAMÉTRICAS DE
TRANSFORMADAS DISCRETAS COM
ALTO GANHO DE CODIFICAÇÃO E
EFICIÊNCIA**

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

Área de Concentração: Comunicações
Linha de Pesquisa: Processamento Digital de Sinais

Orientador: Prof. Dr. Renato J. Cintra.
Coorientador: Prof. Dr. Fábio M. Bayer.

RECIFE
2017

Catálogo na fonte
Bibliotecária: Neide Mesquita Gonçalves Luz / CRB4-1361 (BCTG)

- C229n Canterle, Diego Ramos.
Novas classes multiparamétricas de transformadas discretas com alto ganho de codificação e eficiência. / Diego Ramos Canterle. – 2017.
108 folhas, il., gráfs., tabs.
- Orientador: Prof. Dr. Renato José de Sobral Cintra.
Coorientador: Prof. Dr. Fábio Mariano Bayer.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2017.
Inclui Referências e Apêndices.
1. Engenharia Elétrica. 2. Transformada discreta do cosseno. 3. Série de aproximações BAS. 4. Transformadas aproximadas. 5. Complexidade aritmética. 6. Compressão de imagens. I. Cintra, Renato José de Sobral (Orientador). Bayer, Fábio Mariano (Coorientador). III. Título.

621.3 CDD (22. Ed.)

UFPE/BCTG/2017 - 348



Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

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

DIEGO RAMOS CANTERLE

TÍTULO

**“NOVAS CLASSES MULTIPARAMÉTRICAS DE TRANSFORMADAS
DISCRETAS COM ALTO GANHO DE CODIFICAÇÃO E EFICIÊNCIA”**

A comissão examinadora composta pelos professores: RENATO JOSÉ DE SOBRAL CINTRA, DE/UFPE; JULIANO BANDEIRA LIMA, DES/UFPE e HÉLIO MAGALHÃES DE OLIVEIRA, DE/UFPE, sob a presidência do primeiro, consideram o candidato **DIEGO RAMOS CANTERLE APROVADO.**

Recife, 27 de julho de 2017.

MARCELO CABRAL CAVALCANTI
Coordenador do PPGE

RENATO JOSÉ DE SOBRAL CINTRA
Orientador e Membro Titular Interno

HÉLIO MAGALHÃES DE OLIVEIRA
Membro Titular Externo

JULIANO BANDEIRA LIMA
Membro Titular Interno

*A todos que me apoiaram
incondicionalmente em minhas escolhas.*

AGRADECIMENTOS

Aos meus pais, Jaime Renato Bertazzo Canterle e Tânia Regina Ramos Canterle, pelo suporte e por me apoiar em minhas decisões.

A Renata e a Elizandra por me acolherem em sua casa quando eu cheguei em Recife.

Aos meus amigos do LACRI, Ravi, Bruno, Carlos, José e Rodrigo, que dividiram comigo os dias de trabalho nos últimos dois anos.

Aos meus amigos, Fernando, Renata, Luciano, Luciana, Juan e Bruna, pela companhia e pelos momentos de descontração.

A Andressa Alves Ferreira que foi um pilar muito forte que me sustentou incondicionalmente, sempre que possível, durante minha estada em Recife.

Ao meu orientador, Prof. Dr. Renato J. Cintra, pela orientação e ensinamentos.

Ao meu coorientado, Prof. Dr. Fábio Bayer, pelos ensinamentos, dedicação, motivação, atenção, paciência, pelas cobranças oportunas e por acreditar na minha capacidade. Estando presente na minha vida acadêmica desde a graduação.

A todos os professores da pós-graduação que contribuíram em meus conhecimentos.

A todos professores de todas etapas da minha vida que contribuíram em me tornar uma pessoa melhor.

A FACEPE pelo apoio financeiro.

Aos participantes da banca examinadora, pelas colaborações para melhoria deste trabalho.

Não compita com ninguém, você não tem que demonstrar nada a ninguém. Não tem que chegar aonde outra pessoa chegou, só precisa superar os seus próprios limites.

— Autor desconhecido

RESUMO

Devido à crescente demanda por elevadas taxas em processamento de dados, a comunidade científica vem dando atenção a metodologias que envolvam baixo custo computacional. Uma maneira de reduzir o custo computacional em processamento de sinais é a utilização de transformadas que tenham custo multiplicativo nulo. Neste trabalho, são propostas novas classes de transformadas que possuem alto ganho de codificação. Essas transformadas são aproximações da transformada discreta do cosseno (DCT) e possuem grande utilidade em compressão de imagem e vídeo, como nos padrões JPEG, MPEG-1, MPEG-2, H.261, H.263, H.264 e, mais recentemente, o HEVC. Neste trabalho, são propostas duas novas classes paramétricas de aproximações da DCT de comprimentos 8 e 16. São também introduzidos algoritmos rápidos para estas novas aproximações. Essas duas novas classes foram obtidas como generalizações da série de aproximações da DCT propostas por Bouguezel, Ahmed e Swamy (BAS). As transformadas BAS são coletivamente caracterizadas por elevados ganhos de codificação e de eficiência de transformação. Para as novas classes multiparamétricas de aproximações, foram considerados problemas de otimização multicritério para selecionar as transformadas ótimas. O problema de otimização foi baseado em métricas como: erro quadrático médio, erro de energia total, ganho de codificação unificado e eficiência. A fim de encontrar transformadas de comprimentos maiores, úteis em padrões de vídeo atuais, essas novas classes foram escalonadas para comprimentos 16 e 32. Todas as novas transformadas ótimas obtidas foram comparadas com transformadas arquivadas na literatura. Também foram considerados experimentos de compressão de imagem, considerando-se métricas como a relação sinal-ruído de pico e o índice de similaridade estrutural. De acordo com os resultados obtidos, conclui-se que as novas transformadas possuem bom desempenho em aplicações de compressão de imagem e requerem baixo custo de implementação. Em especial, destacam-se uma transformada de comprimento 8 que exige 20 adições para o seu cômputo, transformadas de comprimento 16 que exigem 48, 52 e 56 adições para seu cômputo e transformadas de comprimento 32 que exigem 128, 136 e 144 adições para seu cômputo.

Palavras-chave: Transformada discreta do cosseno. Série de aproximações BAS. Transformadas aproximadas. Complexidade aritmética. Compressão de imagens.

ABSTRACT

Due to the increasing demand for high rates of data processing, the scientific community has been looking for new methodologies that require low computational cost. One way to reduce the computational cost in signal processing is to use transforms that have null multiplicative cost. In this work, new classes of transforms that have high coding gain are proposed. These transforms are approximations of the discrete cosine transform (DCT) and have great utility in image and video compression, as in the JPEG, MPEG-1, MPEG-2, H.261, H.263, H.264 standards and more recently, the HEVC. Two new parametric classes of DCT approximations of lengths 8 and 16 are proposed. Fast algorithms are also introduced for these new approximations. These two new classes were obtained as generalizations of the series of DCT approximations proposed by Bouguezel, Ahmed and Swamy (BAS). BAS transforms are collectively characterized by high coding gains and efficiency. For the new multiparametric classes of approximations, multicriteria optimization problems were considered to select the optimal transforms. The optimization problem was based on metrics such as: mean square error, total energy error, unified coding gain, and efficiency. In order to find larger-length transforms useful in current video standards, these new classes were scaled to lengths 16 and 32. All new optimal transforms were compared with transforms archived in the literature. We also considered image compression experiments, considering metrics such as the peak-to-noise ratio and the structural similarity index. According to the results, we concluded that the new transforms perform well in image compression applications and require a low implementation cost. In particular we highlight a length 8 transform requiring 20 additions for its computation, length 16 transforms requiring 48, 52 and 56 additions for their computation and length 32 transforms requiring 128, 136 and 144 additions for their computation.

Keywords: Discrete cosine transform. BAS Approximation Series. Approximate transforms. Arithmetic complexity. Image Compression.

LISTA DE ILUSTRAÇÕES

3.1	Caso geral do algoritmo rápido de comprimento 8, relacionando os dados de entrada \mathbf{x}_n , $n = 0, 1, 2, \dots, 7$, aos coeficientes da saída \mathbf{X}_k , $k = 0, 1, 2, \dots, 7$. Setas tracejadas representam multiplicação por -1	49
3.2	Algoritmo rápido da matriz parametrizada BAS de comprimento 16, relacionando os dados de entrada \mathbf{x}_n , $n = 0, 1, 2, \dots, 15$, aos coeficientes da saída \mathbf{X}_k , $k = 0, 1, 2, \dots, 15$. Setas tracejadas representando multiplicação por -1	57
3.3	Gráficos de avaliação das transformadas ótimas propostas de comprimento 8. A linha pontilhada é a fronteira de Pareto para a matriz parametrizada de comprimento 8 proposta.	63
3.4	Análise de <i>trade-off</i> para transformadas de diferentes comprimentos.	74
4.1	Medidas de avaliação das transformadas ótimas propostas com comprimento 8 com relação a compressão de imagem variando o valor de r	78
4.2	Medidas de avaliação das transformadas ótimas propostas com comprimento 16 com relação a compressão de imagem variando o valor de r	80
4.3	Medidas de avaliação das transformadas ótimas propostas com comprimento 32 com relação a compressão de imagem variando o valor de r	82
4.4	Medidas de avaliação das transformadas ótimas propostas com comprimento de 20 adições com relação a compressão de imagem variando o valor de r	83
B.1	Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 5$	96
B.2	Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 10$	97
B.3	Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 20$	98
B.4	Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 30$	99
B.5	Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 20$	100
B.6	Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 40$	101
B.7	Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 80$	102
B.8	Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 120$	103

B.9	Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 80$.	104
B.10	Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 160$.	105
B.11	Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 320$.	106
B.12	Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 480$.	107

LISTA DE TABELAS

2.1	Cota inferior da complexidade multiplicativa da DCT para diferentes comprimentos que sejam potência de dois.	25
2.2	Custo aritmético dos algoritmos para a DCT de comprimento 8	26
2.3	Transformadas obtidas a partir de arredondamentos por funções inteiras.	35
2.4	Transformadas eficientes no subespaço FW a partir da solução do problema de otimização.	36
2.5	Custo aritmético e medidas de avaliação das aproximações consideradas.	43
2.6	Correlação de Pearson entre as medidas de avaliação e o número de adições exigidos por cada aproximação. Entre parênteses o p -valor.	44
2.7	Estimativa dos parâmetros para análises de regressão entre as medidas de avaliação e a complexidade aditiva.	44
3.1	Casos particulares da classe parametrizada BAS.	47
3.2	Soluções das transformadas ortogonais ótimas de comprimento 8.	60
3.3	Medidas de avaliação das transformadas ótimas de comprimento 8.	61
3.4	Medidas de avaliação para as transformadas da série de aproximações BAS e demais transformadas consideradas para a comparação entre as transformadas de comprimento 8.	62
3.5	Medidas de avaliação das transformadas ótimas com comprimento 16.	65
3.6	Medidas de avaliação para as transformadas da série de aproximações BAS e demais transformadas consideradas para a comparação entre as transformadas com comprimento 16.	65
3.7	Medidas de avaliação das transformadas ótimas da parametrização BAS de comprimento 8 escalonadas para comprimento 16 a partir do método JAM.	68
3.8	Medidas de avaliação das transformadas ótimas.	69
3.9	Medidas de avaliação das transformadas ótimas da parametrização BAS de comprimento 16 escalonadas para comprimento 32 a partir do método JAM.	71
3.10	Medidas de avaliação para as transformadas da série de aproximações BAS e demais transformadas consideradas para a comparação de comprimento 32.	71

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	14
1.2	Estado da arte	15
1.3	Objetivos	18
1.4	Estrutura do documento	18
2	KLT E DCT	20
2.1	Introdução	20
2.2	A KLT e a DCT como aproximação assintótica	20
2.3	Complexidade Computacional	22
2.3.1	Complexidade aritmética	24
2.3.2	Algoritmos rápidos	25
2.3.3	Transformadas aproximadas	26
2.4	Obtenção de transformadas aproximadas para a DCT	27
2.5	Aproximações da DCT de comprimento 8	29
2.5.1	Transformada de Walsh-Hadamard	29
2.5.2	DCT Sinalizada	30
2.5.3	Aproximação de nível 1 de Lengwehasatit e Ortega	31
2.5.4	DCT arredondada e DCT arredondada modificada	31
2.5.5	Aproximação para <i>RF imaging</i>	32
2.5.6	Modificação da matriz de Brahimi-Bouguezel	33
2.5.7	Aproximações baseadas em funções inteiras	34
2.5.8	Aproximações a partir da parametrização da fatoração de Feig-Winograd	35
2.5.9	Aproximação baseada em medida angular	36
2.6	Série de aproximações Bouguezel, Ahmad e Swamy	37
2.6.1	BAS_1	37
2.6.2	BAS_2	37
2.6.3	BAS_3	38
2.6.4	BAS_4	38
2.6.5	$BAS_5(a)$	39
2.6.6	BAS_6	40

2.7	Comparação entre as aproximações de comprimento 8	40
2.7.1	Medidas de qualidade	40
2.7.2	Avaliação	41
3	PARAMETRIZAÇÃO DAS BAS	45
3.1	Introdução	45
3.2	Parametrização de comprimento 8	46
3.2.1	Algoritmo rápido	47
3.2.2	Complexidade computacional	49
3.2.3	Ortogonalidade e ortonormalidade	52
3.3	Parametrização de comprimento 16	54
3.3.1	Concentração de energia	55
3.3.2	Algoritmo rápido	56
3.3.3	Complexidade computacional	57
3.3.4	Ortogonalidade e ortonormalidade	57
3.4	Resultados da Busca	58
3.4.1	Otimização multicritério	58
3.4.2	A ordem de Pareto \mathbb{R}^p	58
3.4.3	Solução do problema de otimização	59
3.4.4	Resultados da busca para transformadas de comprimento 8	60
3.4.5	Resultados da busca para transformadas de comprimento 16	64
3.5	O Método de Jrid-Alfalou-Meher	65
3.5.1	Escalonamento da parametrização BAS de comprimento 8 para 16	67
3.5.2	Escalonamento das parametrizações BAS de comprimento 16 para 32	67
3.6	Análise de <i>trade-off</i>	72
4	EXPERIMENTO COMPUTACIONAL	75
4.1	Metodologia	75
4.2	Resultados e discussão	77
4.2.1	Comprimento 8	77
4.2.2	Comprimento 16	79
4.2.3	Comprimento 32	81
4.2.4	Comparação entre as transformadas com comprimento 8 e que necessitam 20 adições para seu cômputo	81
5	CONCLUSÕES	84
	Referências	86
APÊNDICE A	QUANTIZAÇÃO EM TRANSFORMAÇÕES 2-D	93
APÊNDICE B	IMAGENS COMPRIMIDAS	95

INTRODUÇÃO

1.1 Motivação

As transformadas discretas são ferramentas centrais no contexto de processamento de sinais [1], séries temporais [2], codificação de imagem e vídeo [3], criptografia [4], análise espectral [5], solução de equações diferenciais e equações diferenciais parciais [6], entre outros. Em especial, no contexto de codificação de imagem e vídeo, a transformada discreta do cosseno (DCT) vem se revelando como uma ferramenta fundamental [7–9]. Existem diversas versões da DCT, entre elas as mais utilizadas são a DCT-I [10], a DCT-II [1], a DCT-III [1] e a DCT-IV [11]. A DCT-II é uma transformada de grande aplicação no contexto de compressão de imagens [7–9]. Portanto, neste trabalho, abordaremos somente a DCT-II, sendo a mesma simplesmente referenciada como DCT. A DCT vem sendo amplamente utilizada, em padrões como JPEG [12], H.264 [13] e HEVC [14]. O grande interesse pela DCT pode ser justificado por ela ser uma aproximação assintótica da transformada de Karhunen-Loève (KLT) [7] que é uma transformada que possui propriedades ótimas de compactação de energia [15].

Nos dias atuais, é notável a transmissão de grandes massas de dados, portanto, tal tipo de transmissão vem merecendo grande atenção. Logo, ao aplicar a DCT, uma mudança na etapa do processamento pode afetar a velocidade e o consumo de energia do processo de forma significativa [16, 17]. Quando os dados seguem processo Markoviano de primeira ordem altamente correlacionado, como no caso de imagens naturais, aplica-se a DCT em blocos do sinal de entrada para maior eficiência computacional. Contudo, o cômputo da DCT pode se tornar mais eficiente com a utilização de algoritmos rápidos, os quais não podem ser aplicados na KLT devido a sua dependência do sinal de

entrada [7]. Para o cálculo de algoritmos rápidos da DCT, existem diversas opções na literatura [18–21].

Contudo, mesmo com a utilização de algoritmos rápidos e do sistema de blocos, ainda temos a exigência de aritmética de ponto flutuante para o cálculo da DCT, devido a multiplicações não-triviais que são exigidas em seu cômputo [7]. Desta forma, alternativamente, podem ser utilizadas transformadas que são aproximações da DCT. Caso os valores de entrada de tais transformadas sejam racionais diádicos, serão obtidas aproximações livres de multiplicações não-triviais, assim como, em aproximações baseadas em *lifting scheme* [22]. Alternativamente, podem ser utilizados valores de entrada restritos ao subconjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$, sendo, desta forma, livre de multiplicações não-triviais [23–28]. Com isso, o cômputo dessas transformadas aproximadas envolverá apenas somas e deslocamentos de *bit*. Na literatura, existem diversos trabalhos que abordam essas aproximações [16, 24, 25, 27–37]. De acordo com figuras de mérito relevantes ao problema, tais transformadas são consideradas próximas a DCT. Essas transformadas não fornecem o cômputo exato da DCT, mas sim, uma aproximação a um baixíssimo custo aritmético.

No âmbito de aproximações da DCT, separamos o grupo de transformadas de complexidade multiplicativa nula introduzidas por Bouguezel-Ahmad-Swamy [27, 28, 34–37]. Neste trabalho, serão denominadas de série de aproximações BAS. As transformadas BAS possuem alta eficiência de transformação e alto ganho de codificação, apesar de não apresentarem tão boas medidas de proximidade com a DCT [26]. Quanto maior o ganho de codificação de uma transformada maior será sua capacidade de decorrelação do sinal [38], esperando-se melhor desempenho em compressão de imagem.

1.2 Estado da arte

A mais tradicional transformada de complexidade multiplicativa nula da literatura para processamento de imagem é a transformada de Walsh-Hadamard (WHT) [39], porém a mesma não foi proposta como aproximação da DCT. A primeira aproximação proposta para a DCT foi a DCT sinalizada (SDCT) [23]. A SDCT é obtida a partir da aplicação da função sinal em cada elemento da matriz da DCT exata. O cômputo da SDCT de comprimento 8 com algoritmo rápido exige 24 adições. A SDCT é uma transformada não-ortogonal, porém possui inversa simples e seu cômputo com o algoritmo rápido proposto em [40] exige 20 adições e 4 deslocamentos de *bit*. Desta forma, a SDCT, mesmo sendo não-ortogonal, possui inversa com baixa complexidade computacional.

Após a SDCT ser proposta, Lengwehasatit-Ortega [41] propuseram uma série de aproximações

para a DCT de baixa complexidade. Entre tais aproximações, destaca-se a “aproximação de nível 1” por ser uma transformada de comprimento 8 com complexidade multiplicativa nula. Essa transformada é ortogonal, ou seja, sua inversa é igual a sua transposta, requerendo 24 adições e 2 deslocamentos de *bit* para o seu cômputo. Essa transformada possui um bom desempenho considerando a relação sinal-ruído de pico (PSNR) em compressão de imagens.

Um grupo de transformadas de baixa complexidade computacional de comprimento 8 que serão amplamente estudados nesta dissertação são as transformadas BAS [27, 28, 34–37]. Essas transformadas têm por característica não ter grande proximidade com a DCT em relação às métricas erro quadrático médio [7] e erro de energia total [25], porém possuem altos ganhos de codificação unificado [38] e a eficiência [42], tornando-as adequadas para aplicações em compressão de imagens. Além das transformadas de comprimento 8, a transformada proposta em [37] foi escalonada para os comprimentos 16 e 32. Já em [28] foi dada uma expressão geral, para a transformada proposta, a fim de obter seu escalonamento para qualquer comprimento N . Esse grupo de transformadas, em geral, são variações de outras transformadas de baixa complexidade da literatura, principalmente da SDCT.

A DCT arredondada (RDCT) [25] é uma transformada de comprimento 8 que é obtida a partir da aplicação de uma função de arredondamento na DCT exata. A complexidade computacional de seu algoritmo rápido é de 22 adições. Já a RDCT modificada (MRDCT) [24] é uma transformada obtida a partir da modificação conveniente da RDCT e exige apenas 14 adições para sua implementação. A MRDCT é a transformada com menor complexidade computacional encontrada na literatura.

Outra aproximação da DCT é uma transformada criada para aplicação em imagens de rádio-frequência [29], obtida a partir do argumento mínimo do erro de energia total, sendo o resultado de um problema de otimização. Em [43] é proposta uma transformada que é uma modificação da matriz proposta por Brahimi e Bouguezel [44]. Essa transformada necessita apenas 16 adições para o seu cômputo e tem boas medidas de ganho de codificação e eficiência em relação ao seu baixo custo aritmético.

Aplicando funções inteiras aos valores de entrada da DCT exata de comprimento 8 surgiu uma classe de transformadas discretas de baixa complexidade computacional [33]. Essa classe considera um parâmetro de busca que varia em um determinado intervalo e isso gera um grande número de novas transformadas. Introduzindo as restrições de ortogonalidade, *quase*-ortogonalidade e, ainda, no caso de não ocorrer uma dessas duas restrições, as transformadas inversas são simples e com baixa complexidade. Desta forma, o grande número de transformadas geradas pelo método é reduzido a poucas novas transformadas com boas propriedades.

A partir da parametrização do algoritmo rápido de Feig-Winograd [21], em [26] foi obtida uma nova classe de transformadas aproximadas da DCT de comprimento 8, sendo um subespaço matricial que depende de parâmetros. Esse método consiste em realizar uma busca computacional considerando que os parâmetros desse subespaço matricial possuem apenas valores, $\{0, \pm 1/2, \pm 1, \pm 2\}$. Desta forma, encontra-se um grande número de novas transformadas que são submetidas a um problema de otimização multicritério [45, 46]. Como resultado, encontram-se transformadas ótimas segundo um conjunto de métricas de distância e qualidade, que inclui as seguintes medidas: erro quadrático médio [7], erro de energia total [25], ganho de codificação unificado [38] e eficiência [42].

Considerando a minimização de medidas angulares, em [47] é proposta uma nova transformada ortogonal de comprimento 8. Esta transformada possui custo aritmético de 24 adições e 6 deslocamentos de *bit*. O método de busca consiste em comparar as linhas da DCT exata com vetores candidatos, onde seus elementos pertençam ao conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$, a partir de uma certa métrica angular. A linha escolhida é a que minimizar o ângulo entre os vetores a partir da métrica escolhida. Ainda, essa transformada foi escalonada para comprimento 16 e 32 pelo método de Jridi, Alfalou e Meher (JAM) [48], e para seus cálculos são necessárias 64 e 160 adições, respectivamente.

O método de Jridi, Alfalou e Meher [48] é um método de escalonamento de transformadas de comprimentos menores para comprimentos maiores. Por esse método é possível obter transformadas de comprimento $2N$, a partir de transformadas de comprimento N . Ao propor o método em [48], os autores também escalonam a transformada RDCT para comprimento 16 e 32, que necessitam de 60 e 152 adições, respectivamente.

Outras transformadas de comprimento 16 são dadas em [31, 49, 50]. Substituindo os valores de entrada da DCT por valores do conjunto $\{0, \pm 1\}$ respeitando ortogonalidade, simetria em relação a DCT e boas propriedades de compactação de energia, em [31] foi proposta uma aproximação para DCT de comprimento 16 que exige 72 adições para o seu cálculo. A transformada proposta em [50] é obtida a partir de aproximações por inteiros. Tal transformada é ortogonal e necessita 60 adições para o seu cálculo. Em [49], é proposta uma transformada a partir de uma metodologia semelhante a MRDCT e, a melhor de nosso conhecimento, é a transformada de comprimento 16 com menor complexidade aritmética da literatura, necessitando apenas de 44 adições para o seu cálculo. A comparação entre algumas dessas transformadas pode ser visualizada na Tabela 2.5 no Capítulo 2.

1.3 Objetivos

O objetivo geral deste trabalho é: propor uma ampla classe de transformadas de comprimento 8 por meio da parametrização das transformadas propostas por Bouguezel, Ahmad e Swamy. Como as transformada de comprimento 16 são pouco exploradas na literatura, também será proposta uma classe de transformadas de comprimento 16. Ainda como parte do objetivo geral, as parametrizações encontradas serão escalonadas pelo método de Jridi, Alfalou e Meher [48] para comprimentos 16 e 32, investigando e desenvolvendo algoritmos computacionalmente eficientes focados em compressão de imagem e vídeo.

Para alcançar o objetivo geral proposto, nesta dissertação são pontuados os seguintes objetivos específicos:

1. Revisão de literatura do estado da arte em transformadas aproximadas de complexidade multiplicativa nula;
2. Implementação computacional do método de compressão de imagens baseado em transformadas aproximadas;
3. Estudo e implementação computacional de diferentes algoritmos rápidos para fatoração em matrizes esparsas da matriz da DCT;
4. Desenvolvimento e investigação de novas aproximações para a DCT de comprimento 8, 16 e 32;
5. Avaliação computacional da qualidade de aproximação e aplicação das transformadas desenvolvidas em compressão de imagens.

1.4 Estrutura do documento

Esta dissertação está organizada da seguinte forma.

No Capítulo 2, são introduzidas as bases teóricas e as considerações do formalismo matemático da KLT, e apresentada a DCT como aproximação assintótica da KLT. Posteriormente, são apresentadas as aproximações para a DCT de comprimento 8 consideradas neste trabalho e é feita uma revisão na literatura de aproximações de baixa complexidade computacional para a DCT. Finalmente, é apresentado e comparado o custo aritmético e medidas de avaliação destas aproximações.

O Capítulo 3 introduz um método geral para a obtenção de aproximações para a DCT em uma nova fatoração baseada na série de aproximações BAS. Serão apresentados a estrutura matemática da

classe de transformadas propostas e os algoritmos rápidos para as transformadas de comprimentos 8 e 16. Posteriormente, são propostos problemas de otimização multicritério sobre um conjunto de medidas de avaliação com o objetivo de identificar soluções eficientes e identificar novas transformadas ótimas dentro da classe de transformadas propostas. As aproximações obtidas são avaliadas e comparadas com outras transformadas aproximadas já conhecidas. As novas parametrizações foram escalonadas pelo método de Jridi, Alfalou e Meher [48] obtendo novas transformadas de comprimento 16 e 32. Todas as novas transformadas, conjuntamente com as transformadas que são referência na literatura, foram submetidas a uma análise de *trade-off* a fim de comparar o desempenho da transformada e seu custo aritmético conjuntamente.

No Capítulo 4, é feita uma análise de compressão de imagens utilizando as transformadas ótimas obtidas e as referências na literatura como comparação, considerando medidas objetivas de qualidade. Finalmente, no Capítulo 5, são apresentadas as conclusões finais do presente trabalho.

KLT E DCT

2.1 Introdução

Neste capítulo, é descrito o formalismo teórico da KLT e mostra-se que a DCT é assintoticamente equivalente a KLT. Tal equivalência é válida quando é considerado que o sinal de entrada é um processo Markoviano de primeira ordem e o coeficiente de correlação do sinal tende para a unidade ($\rho \rightarrow 1$) [7, 51]. Como a KLT possui boas propriedades de compactação de energia e de decorrelação do sinal, a DCT acaba herdando essas propriedades, o que faz dela uma ferramenta de longa aplicação no contexto de compressão de imagens [7]. Será feita uma discussão sobre complexidade computacional e como obter novas aproximações da DCT a partir de uma matriz de baixa complexidade. A seguir são descritas aproximações da DCT úteis no contexto de compressão de imagens e que possuem baixa complexidade computacional. Também é apresentada uma comparação entre suas complexidades computacionais e medidas de avaliação.

2.2 A KLT e a DCT como aproximação assintótica

Sejam um sinal de entrada $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^T$ e \mathbf{C} é a matriz da transformada, temos:

$$\mathbf{y} = \mathbf{C}^T \cdot \mathbf{x},$$

em que $\det(\mathbf{C}) \neq 0$ e \mathbf{y} é o sinal no domínio da transformada que tende a ser decorrelacionado e usualmente terá a energia concentrada nos primeiros coeficientes.

A matriz de covariância de \mathbf{y} é dada por:

$$\boldsymbol{\Sigma}_{\mathbf{y}} = E(\mathbf{y} \cdot \mathbf{y}^{\top}) = \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_{N-1}), \quad (2.1)$$

em que $E(\cdot)$ é o operador valor esperado. Dada a Equação (2.1), temos:

$$\boldsymbol{\Sigma}_{\mathbf{y}} = E(\mathbf{y} \cdot \mathbf{y}^{\top}) = E(\mathbf{C}^{\top} \cdot \mathbf{x} \cdot \mathbf{x}^{\top} \cdot \mathbf{C}) = \mathbf{C}^{\top} \cdot E(\mathbf{x} \cdot \mathbf{x}^{\top}) \cdot \mathbf{C} = \mathbf{C}^{\top} \cdot \boldsymbol{\Sigma}_{\mathbf{x}} \cdot \mathbf{C}.$$

Como deseja-se preservar a energia do sinal toma-se \mathbf{C} ortogonal. Então, temos:

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbf{C}^{\top} \cdot \boldsymbol{\Sigma}_{\mathbf{y}} \cdot \mathbf{C}.$$

Considerando o sistema

$$\boldsymbol{\Sigma}_{\mathbf{x}} \cdot \mathbf{c}_k = \lambda_k \cdot \mathbf{c}_k, \quad (2.2)$$

com $k = 0, 1, \dots, N - 1$ e sendo $\lambda_k > 0$ o autovalor de $\boldsymbol{\Sigma}_{\mathbf{x}}$ associado ao autovetor \mathbf{c}_k , ao ordenar os λ_k em ordem crescente então a matriz \mathbf{C} é a KLT [7].

Assumindo que o sinal \mathbf{x} é um processo Markoviano de primeira ordem, temos:

$$[\boldsymbol{\Sigma}_{\mathbf{x}}]_{m,n} = \rho^{|m-n|},$$

em que ρ é o coeficiente de correlação de Pearson do sinal e sua magnitude é menor que a unidade [7].

A m -ésima componente do k -ésimo autovetor que satisfaz o sistema de equações expresso em (2.2) é:

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

e seu respectivo autovalor é

$$\lambda_k = \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos \mu_k}, \quad (2.3)$$

em que $\mu = [\mu_1, \dots, \mu_k]^{\top}$ são as raízes positivas da seguinte equação transcendental:

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

Tomando $\rho \rightarrow 1$, temos:

$$\begin{aligned} \lambda_k &= 0, \\ \tan N\mu_k &= 0. \end{aligned}$$

Desta forma, temos $\mu_k = k\pi/N$. Logo:

$$\begin{aligned} c_{k,m} &= \sqrt{\frac{2}{N}} \sin \left[\frac{k\pi}{N} \left(m + 1 - \frac{N-1}{2} \right) + \frac{(k+1)\pi}{2} \right], \\ &= \sqrt{\frac{2}{N}} \cos \left(\frac{k\pi(2m+1)}{2N} \right). \end{aligned}$$

Porém, temos inconsistências para o cálculo de λ_k para $k = 0$ na Equação (2.3). Sabendo que:

$$\text{tr}(\Sigma_{\mathbf{x}}) = \sum_{n=0}^{N-1} [\Sigma_{\mathbf{x}}]_{n,n} = \sum_{k=0}^{N-1} \lambda_k,$$

e que $\lambda_1 = \lambda_2 = \dots = \lambda_{N-1} = 0$ e $[\Sigma_{\mathbf{x}}]_{n,n} = 1$ quando $\rho \rightarrow 1$, temos:

$$\sum_{k=0}^{N-1} \lambda_k = \lambda_0 = \sum_{n=0}^{N-1} [\Sigma_{\mathbf{x}}]_{n,n} = N,$$

e então, como $\lambda_0 = N$, temos

$$c_{0,m} = \sqrt{\frac{1}{N}}.$$

Assim, de forma geral, podemos definir os (k,m) -ésimos elementos da matriz da DCT da seguinte maneira:

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

sendo,

$$v_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{se } k = 0, \\ 1, & \text{se } k > 0. \end{cases}$$

Desta forma, a DCT é construída como uma aproximação assintótica da KLT quando a autocorrelação tende à unidade. Como a DCT acaba sendo um caso particular da KLT, a DCT herda suas boas propriedades [7].

2.3 Complexidade Computacional

A complexidade computacional de algoritmos depende de vários fatores, considerando desde aspectos puramente matemáticos até detalhes de sua realização física em *hardware* ou implementação em *software* [52]. Alguns fatores que gerenciam a complexidade computacional incluem [52]:

1. Simetrias.

Transformações de sequências com uma certa simetria podem ser salvos por algoritmos de pré e pós-processamento. Portanto, a simetria do sinal de entrada pode influenciar na complexidade computacional;

2. Transformações múltiplas.

Em muitas aplicações, as transformadas não são computadas uma por vez, mas sim, mais de uma por vez. Isso geralmente ocorre quando os problemas são colocados em mais de uma dimensão espacial, como por exemplo, em filtragem de imagens bidimensionais ou em soluções de problemas de valores limite em uma região tridimensional.

3. Operações adicionais.

São operação muitas vezes negligenciadas que afetam o cômputo da transformada. Exemplos dessas operações são matrizes de armazenamento adicionais, computação e armazenamento de potência, permutações, movimentos de dados, a aritmética e armazenamento necessários por mapeamentos de índice.

4. Estratégias híbridas.

Consiste em computar transformadas de certo comprimento a partir de transformadas de comprimentos menores. Seria difícil enumerar, analisar ou comparar as várias opções de transformações híbridas para um grande valor de N .

5. Arquitetura.

Transformadas residem em microchips dentro dos sistemas de navegação embarcados espaciais. Portanto, é evidente que o desempenho da transformada depende de questões de *hardware* e arquitetura. Algumas transformadas existentes se encaixam perfeitamente em algumas arquiteturas (por exemplo, Cooley-Tukey base 2 [53] em uma arquitetura de hipercubo). Outras arquiteturas podem exigir algoritmos totalmente novos. Esse fator assegura que nunca haverá um melhor algoritmo, a resposta depende do ambiente e as ferramentas disponíveis.

6. *Software*.

Transformadas são utilizadas em *softwares* para tudo, desde microcomputadores até supercomputadores. Nenhum pacote será único a todos os aplicativos e ambientes de computação, o que torna o seu custo imensurável.

7. Complexidade aritmética.

É composta pelas complexidades multiplicativa, aditiva, e outras operações como deslocamentos de *bit*.

Combinar todos esses fatores em uma única figura de mérito é uma tarefa considerada intratável, tanto por questões de mensuração, quanto por questões de dimensionalidade e combinação dos fatores. Entretanto, dentre esses fatores, temos a complexidade aritmética que é uma medida objetiva,

muito bem posta, determinística e explica parcialmente a capacidade de *hardware*. Desta forma a complexidade aritmética será utilizada para aproximar e explicar parcialmente a complexidade computacional.

2.3.1 Complexidade aritmética

Como foi dito anteriormente, a complexidade aritmética é composta pelas complexidades multiplicativa, aditiva, e outras operações como, deslocamentos de *bit*. Portanto a complexidade aritmética pode ser descrita da seguinte forma:

$$\text{complexidade aritmetica} = g(\text{complexidade multiplicativa, complexidade aditiva, deslocamentos de bit, } \dots),$$

em que $g(\cdot, \cdot, \dots, \cdot)$ é uma função que relaciona seus argumentos. Entre essas complexidades, a complexidade multiplicativa é a que possui as operações mais custosas, pois essas operações consistem em operações de ponto flutuante. Desta forma, implica em maiores exigências de *hardware*. Sendo assim, podemos representar a complexidade multiplicativa como uma aproximação para a complexidade aritmética, da seguinte forma:

$$\text{complexidade aritmetica} \approx \text{complexidade multiplicativa}.$$

A complexidade multiplicativa é dada pelo número de operações de multiplicação.

Temos duas formas de multiplicações que são: as multiplicações triviais e as não-triviais. As multiplicações triviais são multiplicações que não necessitam de aritmética de ponto flutuante, como por exemplo, multiplicações pelos elementos do conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$. Os elementos $\{1/2, \pm 2\}$ são apenas deslocamentos de *bit* quando se trabalha em base 2. Desta forma, apenas multiplicações não-triviais incrementam a complexidade multiplicativa.

A complexidade multiplicativa recebeu um tratamento teórico por Heideman [54]. Em [54], são demonstrados vários resultados sobre a complexidade multiplicativa da transformada discreta de Fourier (DFT) [2], transformada discreta de Hartley (DHT) [55] e da DCT. É notável a obtenção de cotas inferiores para a complexidade multiplicativa de tais transformadas discretas para qualquer comprimento de bloco [54]. Desta forma, a complexidade multiplicativa mínima para a DCT de qualquer comprimento N é dada por [54]:

$$\mathcal{M}(DCT_N) = \frac{\mathcal{M}(DFT_{4N}) - \mathcal{M}(DFT_{2N})}{2},$$

em que $\mathcal{M}(\cdot)$ representa a complexidade multiplicativa, DCT_k é a DCT com comprimento k e DFT_k é a transformada discreta de Fourier [2] de comprimento k . Para a DCT de comprimentos que sejam potências de 2 a cota inferior da complexidade multiplicativa é dada por [54]:

$$\mathcal{M}(DCT_{2^r}) = 2^{r+1} - r - 2.$$

Desta forma, podemos ver na Tabela 2.1 a cota inferior da complexidade multiplicativa para diferentes comprimentos da DCT que sejam potência de dois. Para atingir as cotas inferiores da Tabela 2.1 é necessária a utilização de algoritmos rápidos.

Tabela 2.1: Cota inferior da complexidade multiplicativa da DCT para diferentes comprimentos que sejam potência de dois.

N	Multiplicações
2	1
4	4
8	11
16	26
32	57
64	120
128	247

2.3.2 Algoritmos rápidos

Após a DCT ser proposta em [1], o método mais conveniente para a sua implementação era através da transformada rápida de Fourier (FFT). Porém, implementar a DCT por este método requer aritmética complexa. Como a DCT é amplamente utilizada, foram desenvolvidos algoritmos rápidos para a sua implementação. Alguns dos algoritmos rápidos mais conhecidos são, Chen *et al.* [18], Lee [19], Wang [56], Vetterli e Nussbaumer [57], Suehiro e Hatori [58], Hou [59], Arai *et al.* [60], Loeffler *et al.* [20] e Feig-Winograd [21]. As complexidades aritméticas desses algoritmos rápidos podem ser visualizadas na Tabela 2.2.

Na Tabela 2.2, podemos ver que o algoritmo rápido de Loeffler *et al.* [20] alcançou cota inferior da complexidade multiplicativa para a DCT de comprimento 8 dada por [54]. Porém, apesar de alcançar a complexidade multiplicativa mínima, tal complexidade ainda pode ser significativa para certas aplicações como em processamento de quantidades massivas de dados [61, 62], severa limitação de recursos [29], limitação em potência [63]. Exemplos dessas aplicações são: transmissão e processamento de vídeo em tempo real [61, 62], sistemas de comunicação [7], aplicações de computação

Tabela 2.2: Custo aritmético dos algoritmos para a DCT de comprimento 8

	multiplicações	adições
Arai <i>et al.</i> [60]	5+8	29
Loeffler <i>et al.</i> [20]	11	29
Lee [19]	12	29
Vetterli e Nussbaumer [57]	12	29
Hou [59]	12	29
Wang [56]	13	29
Suehiro e Hatori [58]	13	29
Chen [18]	16	26
Feig-Winograd [21]	22	28

móvel [7], redes de antenas inteligentes de rádio-frequência [29], redes de sensores sem fio [64], entre outras. Nesse contexto, é impossível determinar algoritmo rápido com menor complexidade para a DCT exata. Contudo, uma possível alternativa é calcular a DCT de forma aproximada com complexidade multiplicativa menor que a cota inferior de [54].

2.3.3 Transformadas aproximadas

Quando se trabalha com aproximações para a DCT, deve-se encontrar resultados próximos a DCT com complexidade multiplicativa menor que cota inferior proposta em [54]. Uma das características da matriz da DCT, é que suas entradas são números irracionais. Portanto, é impossível implementar a DCT de forma exata em uma estrutura de *hardware*, tendo assim um erro embutido em sua implementação [65]. Então calcular a DCT de forma aproximada, desde que seja uma aproximação razoável, não é um problema, pois mesmo a DCT exata possui um erro em seu cálculo quando realizada fisicamente.

Aproximações para a DCT podem ser feitas através de matrizes cujos elementos sejam números racionais. Números racionais podem ser representados de forma exata e finita em números binários, portanto podem ser implementados de forma exata em estruturas de *hardware*. Porém, operação com números racionais podem requerer aritmética de ponto flutuante. Uma alternativa para obter aproximações com ausência de aritmética de ponto flutuante é encontrar matrizes onde seus elementos de entrada sejam racionais diádicos [7]. Racionais diádicos podem ser representados da seguinte forma [7]:

$$\frac{p}{2^k}, p \in \mathbb{Z}, k \in \mathbb{N}, p \in \mathbb{Z}, p \text{ ímpar.}$$

Essa é uma representação finita e exata em números binários e não necessitam de operações com ponto flutuante. Portanto, aproximações provenientes de representações utilizando racionais diádicos são muito úteis, como as aproximações baseadas em *lifting scheme* [22]. Porém, aproximações cujas entradas da matriz são racionais diádicos ainda podem exigir um grande número de adições e deslocamentos de *bit*.

Do ponto de vista computacional, uma alternativa para os racionais diádicos, é utilizar elementos que não introduzam adições extras como elementos de entrada da matriz que será a aproximação. Tal fato ocorre para potências de dois, ao multiplicar um valor por 2^k , com $k \in \mathbb{Z}$, tal cálculo irá requerer apenas r deslocamentos de *bit* para sua implementação. Apesar deste método não introduzir adições ou multiplicações não-triviais, ainda pode requerer uma quantidade significativa de deslocamentos de *bit*.

De maneira a minimizar o número de somas e deslocamentos de *bit*, podem ser consideradas aproximações que envolvem apenas elementos do conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$. Tal cenário não incrementa o número de adições e os valores com entradas $\{\pm 1/2, \pm 2\}$ introduzem apenas um deslocamento de *bit* por entrada, incrementando apenas uma quantidade reduzida de deslocamentos de *bit*.

2.4 Obtenção de transformadas aproximadas para a DCT

Ortogonalidade é uma propriedade desejável em aproximações para a DCT [7, 41]. Dada que uma matriz é ortogonal temos que sua inversa é dada por sua transposta. Assim, uma matriz de baixa complexidade terá sua inversa de baixa complexidade, o que não acontece na maioria dos casos para matrizes não-ortogonais. Neste trabalho consideraremos as seguintes definições para ortogonalidade e ortonormalidade.

Definição 1: Dizemos que uma matriz quadrada \mathbf{T} é ortogonal se $\mathbf{T} \cdot \mathbf{T}^\top$ resulta em uma matriz diagonal.

Definição 2: Dizemos que uma matriz quadrada \mathbf{T} é ortonormal se $\mathbf{T} \cdot \mathbf{T}^\top = \mathbf{T}^\top \cdot \mathbf{T}$ resulta na matriz identidade.

A matriz da DCT é ortonormal. Portanto, uma maneira de encontrar aproximações da DCT é a partir de uma matriz ortogonal \mathbf{T} de baixa complexidade, com ausência de elementos multiplicativos, e ortonormaliza-la a partir de uma matriz diagonal de ajuste. De acordo com o método de ortonormalização baseado na decomposição polar [66], a matriz \mathbf{T} é ortonormalizada por meio apenas de uma matriz diagonal de ajuste \mathbf{S} [67]. A partir da decomposição polar, a matriz diagonal de ajuste pode

ser obtida por

$$\mathbf{S} = \sqrt{[\mathbf{T} \cdot \mathbf{T}^\top]^{-1}}, \quad (2.4)$$

em que $\sqrt{\cdot}$ denota a raiz quadrada matricial [68]. Então a aproximação para a DCT é dada por,

$$\hat{\mathbf{C}} = \mathbf{S} \cdot \mathbf{T}.$$

Em algumas aplicações, como em compressão de imagens, o cálculo da transformada é apenas uma etapa de um procedimento maior e a matriz \mathbf{S} pode ser incorporada em outras etapas do procedimento, como pode ser visto no Apêndice A. Desta forma, a matriz \mathbf{S} não irá incrementar a complexidade computacional do processamento.

Quando temos uma matriz de baixa complexidade \mathbf{T} que é não-ortogonal, a matriz \mathbf{S} não será uma matriz diagonal. Desta forma, a matriz \mathbf{S} não poderá ser facilmente incorporada na etapa de quantização. Quando \mathbf{T} é não-ortogonal, para obter uma aproximação para a DCT, podemos generalizar (2.4). Tal generalização é dada por [67]

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

em que $\text{diag}(\cdot)$ retorna uma matriz diagonal cujos elementos são a diagonal principal da matriz dada como argumento de entrada e se o argumento de entrada for um vetor retornará uma matriz diagonal cujos elementos do vetor será a diagonal principal de uma matriz diagonal. Assim, para \mathbf{T} não-ortogonal, uma aproximação não-ortonormal para a DCT pode ser obtida por:

$$\tilde{\mathbf{C}} = \tilde{\mathbf{S}} \cdot \mathbf{T}.$$

Neste método de obtenção de aproximações não-ortogonais para DCT, serão excluídos os elementos fora da diagonal principal de $\mathbf{T} \cdot \mathbf{T}^\top$. Desta forma, será pedida energia ao construir a matriz diagonal de ajuste $\tilde{\mathbf{S}}$. Portanto, quanto mais elementos estiverem fora da diagonal principal da matriz $\mathbf{T} \cdot \mathbf{T}^\top$, pior será a qualidade da aproximação.

Dado um vetor $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^\top$, sua transformada aproximada $\mathbf{X} = [X_0, X_1, \dots, X_{N-1}]^\top$ é dada por:

$$\mathbf{X} = \tilde{\mathbf{C}} \cdot \mathbf{x} = \tilde{\mathbf{S}} \cdot \mathbf{T} \cdot \mathbf{x},$$

e sua transformada inversa é dada por:

$$\mathbf{x} = \tilde{\mathbf{C}}^{-1} \cdot \mathbf{X} = \mathbf{T}^{-1} \cdot \tilde{\mathbf{S}}^{-1} \cdot \mathbf{X}.$$

Considerando ortogonalidade a transformada direta e inversa são dadas, respectivamente, por,

$$\begin{aligned}\mathbf{X} &= \hat{\mathbf{C}} \cdot \mathbf{x} = \hat{\mathbf{S}} \cdot \mathbf{T} \cdot \mathbf{x}, \\ \mathbf{x} &= \hat{\mathbf{C}}^\top \cdot \mathbf{X} = \mathbf{T}^\top \cdot \hat{\mathbf{S}} \cdot \mathbf{X}.\end{aligned}$$

Ainda, pode-se notar que a transformação inversa, no caso de ortogonalidade, tem as mesmas propriedades de baixa complexidade da transformação direta.

Dada uma matriz \mathbf{A} , sua transformada bidimensional direta e inversa, são dadas, respectivamente, por

$$\begin{aligned}\mathbf{B} &= \tilde{\mathbf{C}} \cdot \mathbf{A} \cdot \tilde{\mathbf{C}}^\top, \\ \mathbf{A} &= \tilde{\mathbf{C}}^{-1} \cdot \mathbf{B} \cdot (\tilde{\mathbf{C}}^\top)^{-1}.\end{aligned}$$

Dado ortogonalidade, temos

$$\begin{aligned}\mathbf{B} &= \hat{\mathbf{C}} \cdot \mathbf{A} \cdot \hat{\mathbf{C}}^\top, \\ \mathbf{A} &= \hat{\mathbf{C}}^\top \cdot \mathbf{B} \cdot \hat{\mathbf{C}}.\end{aligned}$$

Como no caso unidimensional, no caso bidimensional, dado ortogonalidade, a transformada inversa tem as mesmas propriedades de baixa complexidade da transformada direta.

2.5 Aproximações da DCT de comprimento 8

Nesta seção, serão apresentadas aproximações da DCT de baixo custo computacional, que são amplamente utilizadas no contexto de processamento de imagem. Serão apresentadas apenas as matrizes de baixa complexidade sem a matriz diagonal de ajuste.

2.5.1 Transformada de Walsh-Hadamard

Seja \mathbf{T}_{WHT} a matriz da transformada de Walsh-Hadamard (WHT) [39] de comprimento N , em que os valores de seus elementos são apenas $\{\pm 1\}$ e satisfazem a seguinte equação:

$$\mathbf{T}_{\text{WHT}} \cdot \mathbf{T}_{\text{WHT}}^\top = N \cdot \mathbf{I}_N, \quad (2.5)$$

em que \mathbf{I}_N é a matriz identidade de ordem N . De acordo com a Equação (2.5), pode-se verificar que a WHT é uma transformada ortogonal. Particularmente, a WHT de comprimento 8 é induzida pela

seguinte matriz:

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

Apesar da WHT não ser definida como aproximação da DCT, ela é muito usada em processamento de imagem pela sua simplicidade e seu bom desempenho. Em particular, a WHT é largamente utilizada em aplicações como codificação, compressão de imagem e análise espectral [39, 69, 70]. Por ter entradas apenas em $\{\pm 1\}$, a WHT é de baixo custo computacional e o caso particular de comprimento 8 necessita de 24 adições para o seu cômputo, assim como para o cômputo da sua inversa, através do uso de um algoritmo rápido.

2.5.2 DCT Sinalizada

A SDCT [23] foi a primeira transformada da literatura de baixa complexidade que foi proposta como aproximação da DCT. A SDCT é obtida a partir da aplicação da função sinal na DCT exata. A função sinal é dada por:

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

A SDCT de comprimento 8 é expressa de forma matricial da seguinte maneira:

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

A SDCT é uma transformada não-ortogonal, porém para o caso de comprimento 8, salvo por um fator de $1/8$, a sua inversa é de baixa complexidade e é dada por:

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

No caso de comprimento 8, a inversa também é de baixa complexidade, pois o fator $1/8$ em aplicações de compressão de imagens pode ser escrito como a matriz diagonal de ajuste \mathbf{S} , e será absorvido na etapa de quantização. Desta forma, a transformada direta necessita de 24 adições e a inversa de 20 adições e 4 deslocamentos de *bit* com a utilização de algoritmos rápidos. Como a SDCT é não-ortogonal, para outros comprimentos, a inversa pode não ter baixa complexidade computacional.

2.5.3 Aproximação de nível 1 de Lengwehasatit e Ortega

Lengwehasatit e Ortega em [41] propuseram cinco aproximações para a DCT de baixa complexidade aritmética. Essas transformadas foram propostas com intuito de minimizar o PSNR. A aproximação de nível 1 é uma entre essas cinco transformadas propostas em [41] e é a única ortogonal e de complexidade multiplicativa nula. Tal transformada é dada por:

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

Essa transformada possui boas medidas de qualidade sendo reconhecida na literatura por essas propriedades [26]. Porém, dentre as transformadas aproximadas de complexidade multiplicativa nula, essa transformada é conhecida por possuir um custo aritmético relativamente alto.

2.5.4 DCT arredondada e DCT arredondada modificada

A DCT arredondada (RDCT) [25] é uma aproximação da DCT de comprimento 8 resultante da aplicação da função de arredondamento na matriz da DCT exata. Seja $y \in \mathbb{R}$ e $\lfloor y \rfloor$ o arredondamento

para o maior inteiro que não excede y , a função de arredondamento é dada por:

$$\text{round}(y) = \lfloor x + 0,5 \rfloor.$$

Em matrizes a função $\text{round}(\cdot)$ é aplicada elemento a elemento.

Desta forma, sendo \mathbf{C} a matriz da DCT exata de comprimento 8, a matriz da RDCT é definida da seguinte maneira:

$$\mathbf{T}_{\text{RDCT}} = \text{round}(2 \cdot \mathbf{C}). \quad (2.6)$$

Efetuando o cálculo da Equação (2.6), a matriz da transformada é dada explicitamente por:

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

Como as entradas desta transformada possuem apenas valores pertencentes ao conjunto $\{0, \pm 1\}$, ela necessita apenas de somas para o seu cômputo. Essa transformada é ortogonal, portanto a transformada inversa, assim como a transformada direta, também é de baixa complexidade.

A partir da RDCT, Bayer e Cintra [24] propuseram a DCT arredondada modificada (MRDCT) [24] como uma nova aproximação para a DCT de comprimento 8. Essa transformada é, até o momento, a aproximação da DCT com menor complexidade aritmética da literatura, necessitando apenas de 14 adições para o seu cômputo. Tal transformada é dada por:

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

O grande número de entradas com valor zero explicam o baixo custo computacional associado.

2.5.5 Aproximação para *RF imaging*

A aproximação para *RF imaging* é uma aproximação da DCT com aplicação focada em imagens de rádio frequência. Esta transformada é obtida a partir de um problema de otimização dado da

seguinte forma [29]:

$$\mathbf{T}_{\text{RF}} = \arg \min_{\mathbf{T}'} \epsilon(\mathbf{T}'),$$

em que $\epsilon(\cdot)$ é o erro de energia total [25]. É considerada uma parametrização a partir da DCT exata e entradas do conjunto $\{0, \pm 1, \pm 2\}$. Desta forma, considera-se todas transformadas deste subespaço matricial.

Resolvendo este problema de otimização, obtém-se a seguinte transformada de baixa complexidade:

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

Essa transformada também é ortogonal e de baixa complexidade, com complexidade computacional de 24 adições e 6 deslocamentos de *bit*.

2.5.6 Modificação da matriz de Brahimi-Bouguezel

Em [43] é proposta uma transformada a partir da modificação da matriz de Brahimi-Bouguezel (BB) [44]. A transformada Brahimi-Bouguezel modificada (MBB) foi obtida a partir da substituição de dois elementos da matriz de Brahimi-Bouguezel de zero para um. A matriz de Brahimi-Bouguezel é dada por:

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

e a MBB é dada por:

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

A transformada MBB necessita apenas de 16 adições para seu cômputo, e difere da MRDCT apenas em 4 entradas que são modificadas de zero para um, o que justifica sua baixa complexidade.

2.5.7 Aproximações baseadas em funções inteiras

Cintra, Bayer e Tablada (CBT) [33] propuseram uma série de aproximações da DCT baseadas em funções de arredondamento que retornam como respostas números inteiros. Essas funções de arredondamento em matrizes são aplicadas ponto a ponto. Todas as transformadas propostas se encaixaram na seguinte fatoração:

$$\mathbf{T}_{\text{CBT}} = \frac{1}{2} \cdot \tilde{\mathbf{P}} \cdot \tilde{\mathbf{K}}(\gamma) \cdot \tilde{\mathbf{B}}_1 \cdot \tilde{\mathbf{B}}_2 \cdot \tilde{\mathbf{B}}_3,$$

em que $\tilde{\mathbf{P}}$ é uma matriz de permutação, $\gamma = (\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6)^\top$ é o vetor de parâmetros, $\tilde{\mathbf{B}}_1$, $\tilde{\mathbf{B}}_2$ e $\tilde{\mathbf{B}}_3$ são matrizes distintas dadas em [33] e $\tilde{\mathbf{K}}(\gamma)$ é dado por:

$$\hat{\mathbf{K}}(\gamma) = \begin{bmatrix} \gamma_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_5 & \gamma_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\gamma_1 & \gamma_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma_4 & -\gamma_6 & \gamma_2 & \gamma_0 \\ 0 & 0 & 0 & 0 & -\gamma_0 & \gamma_4 & -\gamma_6 & \gamma_2 \\ 0 & 0 & 0 & 0 & -\gamma_2 & -\gamma_0 & \gamma_4 & -\gamma_6 \\ 0 & 0 & 0 & 0 & \gamma_6 & -\gamma_2 & -\gamma_0 & \gamma_4 \end{bmatrix}.$$

A partir da metodologia utilizada, as matrizes encontradas que já existem na literatura são a SDCT, RDCT e a aproximação para *RF imaging*. As novas transformadas (CBT_j), encontradas de acordo com os diferentes valores de γ , estão na Tabela 2.3.

Pode-se notar na Tabela 2.3 que o vetor γ possui, em algumas de suas entradas, o valor 3. Contudo, mesmo não pertencendo ao conjunto $\{0, \pm 1/5, \pm 1, \pm 2\}$, ele é de baixa complexidade e pode ser expresso por $2 + 1$, representando uma soma e um deslocamento de *bit*.

Tabela 2.3: Transformadas obtidas a partir de arredondamentos por funções inteiras.

j	γ	Ortogonal?
1	$[2\ 0\ 1\ 1\ 1\ 1\ 0]^\top$	Sim
2	$[2\ 2\ 1\ 1\ 1\ 0\ 0]^\top$	Sim
3	$[3\ 3\ 2\ 2\ 2\ 1\ 0]^\top$	Sim
4	$[1\ 1\ 1\ 1\ 1\ 1\ 0]^\top$	Sim
5	$[2\ 1\ 1\ 1\ 1\ 1\ 0]^\top$	Sim
6	$[3\ 2\ 2\ 2\ 1\ 1\ 1]^\top$	Sim
7	$[1\ 1\ 1\ 1\ 0\ 0\ 0]^\top$	Não
8	$[2\ 2\ 2\ 1\ 1\ 1\ 1]^\top$	Não
9	$[2\ 2\ 2\ 2\ 1\ 1\ 1]^\top$	Não

2.5.8 Aproximações a partir da parametrização da fatoração de Feig-Winograd

Tablada, Bayer e Cintra [26] propuseram uma classe de aproximações da DCT baseadas no algoritmo rápido de Feig-Winograd [21]. Os autores fatoram o algoritmo de uma maneira mais conveniente que é dada por:

$$\mathbf{T}_{\text{FW}} = \frac{1}{2} \cdot \mathbf{P} \cdot \mathbf{K}(\gamma) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3,$$

em que \mathbf{P} é uma matriz de permutação, $\gamma = (\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6)^\top$ é o vetor de parâmetros, \mathbf{B}_1 , \mathbf{B}_2 e \mathbf{B}_3 são matrizes distintas dadas em [26] e $\mathbf{K}(\gamma)$ é dado por:

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

Se os valores de γ pertencerem ao conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$ as transformadas deste subespaço matricial serão todas de baixa complexidade aritmética. De acordo com os valores de γ , a SDCT, a aproximação de nível 1, a RDCT, MRCDT, a aproximação para *RF imaging* e a matriz de Haar [71] pertencem a esse subespaço.

A fim de encontrar novas transformadas foi feita uma busca matricial neste subespaço considerando o conjunto $\{0, \pm 1/5, \pm 1, \pm 2\}$ para os valores de γ . Dado o grande número de transformadas geradas foi proposto um problema de otimização multicritério para selecionar apenas as transforma-

das ótimas. A partir do problema de otimização foram encontradas seis novas transformadas (FW_j) com diferentes valores de γ , dados na Tabela 2.4.

Tabela 2.4: Transformadas eficientes no subespaço FW a partir da solução do problema de otimização.

j	γ	Ortogonal?
1	$[1\ 2\ 0\ 1\ 0\ 1\ 0]^T$	Sim
2	$[0\ 1\ 1\ 1\ 1\ 0\ 0]^T$	Sim
3	$[0\ 2\ 1\ 1\ 1\ 1\ 0]^T$	Sim
4	$[0\ 2\ 2\ 1\ 1\ 1\ 0]^T$	Sim
5	$[2\ 2\ 0\ 1\ 0\ 1\ 0,5]^T$	Sim
6	$[1\ 1\ 1\ 1\ 0\ 0\ 0]^T$	Não

Todas as transformadas da Tabela 2.4 são de baixa complexidade aritmética, sendo cinco ortogonais e uma não ortogonal. Este grupo de transformadas, assim como no grupo de transformadas baseadas em funções inteiras, tem seu algoritmo rápido baseado na fatoração de Feig-Winograd. Porém, neste grupo foram selecionadas apenas as transformadas ótimas de acordo com a otimização multicritério. Além disso, também não foi considerado o valor 3 como possível valor dos parâmetros na busca computacional.

2.5.9 Aproximação baseada em medida angular

A aproximação baseada em medidas angulares (ABM) é uma transformada de comprimento 8 proposta em [47]. Neste trabalho também foram obtidas as suas versões escalonadas para comprimentos 16 e 32 a partir do método de Jridi, Alfalou e Meher [48]. A metodologia proposta consiste em minimizar o ângulo entre as linhas da DCT exata e das transformadas aproximadas. As linhas são consideradas vetores e as linhas das transformadas aproximadas possuem apenas valores que remetem baixa complexidade computacional. Tal transformada de comprimento 8 é dada por:

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

Essa transformada é ortogonal e apesar de ser baseada em medidas angulares não é a melhor transformada na literatura em relação a medidas de avaliação propostas em [47], entretanto se destaca

em várias outras medidas. A transformada ABM ainda foi escalonada para comprimentos 16 e 32 pelo método JAM.

2.6 Série de aproximações Bouguezel, Ahmad e Swamy

As transformadas propostas por Bouguezel, Ahmad e Swamy [27, 28, 34–37], que são denotadas nesse trabalho como série de aproximações BAS, ou apenas série BAS, são aproximações para a DCT exata e possuem baixa complexidade computacional. Em geral, essas transformadas são baseadas na SDCT e seus elementos de entrada estão no conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$. Apesar de, segundo algumas métricas, essas transformadas não possuem grande proximidade com a DCT, elas possuem altos ganho de codificação e eficiência o que torna o seu uso apropriado para compressão de imagens.

2.6.1 BAS_1

A transformada proposta em [34] será denotada nesse trabalho como BAS_1 . O método de obtenção desta transformada foi baseado em introduzir de maneira conveniente zeros e $1/2$ em algumas das entradas da matriz da SDCT. A BAS_1 é dada por:

$$BAS_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1/2 & -1/2 & -1 & -1 & -1/2 & 1/2 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1/2 & -1 & 1 & -1/2 & -1/2 & 1 & -1 & 1/2 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Essa transformada é ortogonal e necessita de 18 adições e 2 deslocamentos de *bit* para o seu cômputo a partir de um algoritmo rápido.

2.6.2 BAS_2

Denotaremos por BAS_2 a transformada proposta em [35]. Seu método de obtenção é semelhante ao método de obtenção da BAS_1 , porém neste caso são introduzidos zeros de forma conveniente em

algumas das entradas da SDCT. Desta forma, a BAS_2 é dada por:

$$BAS_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 0 & 0 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

O algoritmo rápido dessa transformada demanda 21 adições. A BAS_2 é não ortogonal, porém possui inversa que apresenta apenas termos que implicam em baixa complexidade computacional. Sua inversa é dada por:

$$BAS_2^{-1} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 1 & 1 & 0 & 0 & -1 & -1 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{bmatrix}.$$

Essa matriz tem como entrada apenas os valores em $\{0, \pm 1, \pm 2\}$ e, a partir de um algoritmo rápido, necessita de 21 adições e 3 deslocamentos de *bit* para o seu cômputo.

2.6.3 BAS_3

A transformada introduzida em [36], assim como a BAS_2 , foi baseada na introdução de zeros em algumas das entradas da SDCT e será denotada por BAS_3 . A BAS_3 é dada por:

$$BAS_3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix},$$

sendo ortogonal e requerendo apenas 18 adições para o seu cômputo através de um algoritmo rápido.

2.6.4 BAS_4

Em [37], foram propostas transformadas de comprimento 8, 16 e 32 baseadas na modificação da transformada de nível 4 de Lengwehasatit e Ortega [41]. A aproximação proposta de comprimento 8

será denotada neste trabalho por BAS_4 . A matriz da transformada BAS_4 é dada por:

$$BAS_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 1 & -1 & -2 & 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 1 & -2 & 2 & -1 & 1 & -2 & 2 & -1 \end{bmatrix}.$$

Essa matriz é ortogonal e a partir de um algoritmo rápido baseado no algoritmo rápido da WHT necessita de 24 adições e 4 deslocamentos de *bit* para o seu cômputo.

A transformada de comprimento 16 proposta em [37] é dada por:

$$BAS_{4_{16}} = \mathbf{P}_{16} \cdot \begin{bmatrix} BAS_4 & \mathbf{0}_8 \\ \mathbf{0}_8 & BAS_4 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_8 & \mathbf{I}_8 \\ \mathbf{I}_8 & -\mathbf{I}_8 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_8 & \mathbf{0}_8 \\ \mathbf{0}_8 & \bar{\mathbf{I}}_8 \end{bmatrix},$$

em que \mathbf{P}_{16} é uma matriz de permutação dada em [37], $\mathbf{0}_k$ é uma matriz de zeros, \mathbf{I}_k é a matriz identidade e $\bar{\mathbf{I}}_k$ é a matriz contra identidade, todas de ordem k . Já a matriz da transformada de comprimento 32 proposta em [37] é dada por:

$$BAS_{4_{32}} = \mathbf{P}_{32} \cdot \begin{bmatrix} BAS_{4_{16}} & \mathbf{0}_{16} \\ \mathbf{0}_{16} & BAS_{4_{16}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{16} & \mathbf{I}_{16} \\ \mathbf{I}_{16} & -\mathbf{I}_{16} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{16} & \mathbf{0}_{16} \\ \mathbf{0}_{16} & \bar{\mathbf{I}}_{16} \end{bmatrix},$$

em que \mathbf{P}_{32} é uma matriz de permutação dada em [37]. As transformadas $BAS_{4_{16}}$ e $BAS_{4_{32}}$ têm complexidades aritméticas de 64 adições e 8 deslocamentos de *bit* e 160 adições e 16 deslocamentos de *bit*, respectivamente.

2.6.5 $BAS_5(a)$

Em [27], foi proposta uma transformada paramétrica que neste trabalho denotaremos de BAS_5 . Esta transformada é uniparamétrica, tem comprimento 8 e é dada por:

$$BAS_5(a) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & a & -a & -1 & -1 & -a & a & 1 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ a & -1 & 1 & -a & -a & 1 & -1 & a \end{bmatrix}.$$

A partir de um algoritmo rápido, essa transformada requer 18 adições e duas multiplicações. Porém, os autores sugerem utilizar $a = 0, 1/2, 1$. Desta forma, a transformada é ortogonal e necessita de 16 adições para $a = 0$, 18 adições para $a = 1$ e 18 adições e 2 deslocamentos de *bit* para $a = 1/2$.

2.6.6 BAS₆

Neste trabalho, denotaremos por BAS₆ a transformada proposta em [28]. A BAS₆ é obtida da permutação das linhas da transformada de Hadamard [39] baseada na SDCT. Tal transformada é dada por:

$$\text{BAS}_6 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}.$$

É de fácil verificação que a aproximação BAS₆ é a própria WHT, ou seja, a WHT é tratada como uma aproximação da DCT. Desta forma, a mesma herda as suas propriedades, é ortogonal e necessita de 24 adições para o seu cômputo. Em [28] é dada expressão para escalonar a BAS₆ para qualquer comprimento N , tal expressão é dada por:

$$\hat{C}_N(n, k) = (-1)^{\sum_{i=0}^{r-1} (n_i + n_{i+1})k_{r-i-1}},$$

em que $0 \leq n, k \leq N - 1$, n e k são dígitos binários dados por, $n = n_{r-1}2^{r-1} + n_{r-2}2^{r-2} + \dots + n_12 + n_0$, $k = k_{r-1}2^{r-1} + k_{r-2}2^{r-2} + \dots + k_12 + k_0$, $0 \leq i \leq r - 1$ e $r = \log_2 N$ é um número inteiro positivo. Os casos particulares de comprimentos 16 e 32 necessitam de 64 e 160 adições para o seu cômputo, respectivamente. Ainda, existe outra transformada de baixa complexidade proposta em [28], porém, a mesma é uma aproximação para a DHT e não será considerada neste trabalho.

2.7 Comparação entre as aproximações de comprimento 8

Nesta seção, será explorada a complexidade aritmética e algumas medidas de qualidade das transformadas aproximadas descritas anteriormente. Tais transformadas serão avaliadas em relação à medidas de aproximação com a DCT, como o erro quadrático médio [7] e o erro de energia total [25], e medidas de codificação, como o ganho de codificação unificado [38] e a eficiência [42].

2.7.1 Medidas de qualidade

Para avaliação de desempenho das transformadas consideradas foram utilizadas as seguintes medidas: erro quadrático médio (MSE), erro de energia total (ϵ), ganho de codificação unificado (C_g^*)

e eficiência de transformação (η). O erro quadrático médio [7] e o erro de energia total [25] são medidas de proximidade da DCT e medem empiricamente a distância entre as matrizes aproximadas e a DCT exata. Já o ganho de codificação [38] e eficiência da transformada [42], são medidas de codificação, e medem o poder de decorrelação e compactação de energia.

Seja \mathbf{C} uma matriz de transformação exata e $\hat{\mathbf{C}}$ uma matriz que aproxima \mathbf{C} . Uma medida que mensura o erro na saída entre a aproximação $\hat{\mathbf{C}}$ e \mathbf{C} é o MSE, e é dado por [7]:

$$\text{MSE}(\hat{\mathbf{C}}) = \frac{1}{N} \cdot \text{tr} \left\{ (\mathbf{C}_N - \hat{\mathbf{C}}) \cdot \mathbf{R}^{(\mathbf{x})} \cdot (\mathbf{C}_N - \hat{\mathbf{C}})^\top \right\},$$

em que $\text{tr}(\cdot)$ é a função traço [72].

A similaridade entre a transformada aproximada e a matriz exata pode ser dada pelo erro de energia total [25]:

$$\epsilon(\hat{\mathbf{C}}) = \pi \cdot \|\mathbf{C}_N - \hat{\mathbf{C}}\|^2.$$

O ganho de codificação unificado de uma transformada $\hat{\mathbf{C}}$ é dado por [38]:

$$C_g^*(\hat{\mathbf{C}}) = 10 \cdot \log_{10} \left\{ \prod_{k=1}^N \frac{1}{\sqrt[N]{A_k \cdot B_k}} \right\},$$

em que $A_k = \text{su} \{ (\mathbf{h}_k^\top \cdot \mathbf{h}_k) \odot \mathbf{R}^{(\mathbf{x})} \}$, \mathbf{h}_k é o k -ésimo vetor linha de $\hat{\mathbf{C}}$, a função $\text{su}(\cdot)$ retorna a soma dos elementos de seu argumento matricial, \odot é o produto matricial de Hadamard [72], $\mathbf{R}^{(\mathbf{x})}$ é a matriz de covariância de um processo Markoviano de primeira ordem, $B_k = \|\mathbf{g}_k\|^2$ e \mathbf{g}_k é o k -ésimo vetor linha de $\hat{\mathbf{C}}^{-1}$.

Outra medida de codificação é a eficiência da transformada, dada por [42]:

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

em que $r_{i,j}$ é o (i, j) -ésimo elemento de $(\hat{\mathbf{C}} \cdot \mathbf{R}^{(\mathbf{x})} \cdot \hat{\mathbf{C}}^\top)$. Essas duas últimas medidas dizem respeito o desempenho em compressão de dados e não medem a proximidade da aproximação com a DCT.

2.7.2 Avaliação

Ao escolher uma transformada aproximada para aplicações, em geral, não se deve dar atenção exclusivamente à complexidade aritmética da transformada. Também deve-se considerar outras medidas de avaliação, que irão indicar o quão boa cada transformada é para cada aplicação. Nesse sentido, deve-se ponderar o custo aritmético de cada transformada com alguma medida de qualidade.

Na Tabela 2.5, podemos observar a complexidade aritmética e as medidas de qualidade das aproximações discutidas nas Seções 2.5 e 2.6. Nota-se que a aproximação MRDCT possui o menor custo

aritmético e a aproximação CBT_5 possui o maior custo aritmético entre todas as transformadas listadas. Ainda, observando a Tabela 2.5, podemos notar que medidas de avaliação, em geral, melhoram de acordo com o aumento da complexidade aritmética das transformadas. Nota-se, também, que as transformadas da série BAS possuem, em geral, maior ganho de codificação e eficiência que as transformadas concorrentes com o mesmo custo aritmético. Dado este fato, frisa-se a necessidade de generalizar a série das aproximações BAS em uma ampla classe.

As correlações de Pearson entre as medidas de avaliação e a complexidade aditiva das transformadas aproximadas são apresentadas na Tabela 2.6. O MSE e o ϵ , que são medidas de proximidade com a DCT, são correlacionados com coeficiente de correlação de 0,75. Assim como as medidas de proximidade com a DCT, as medidas que avaliam a codificação, que são η e C_g^* , também são correlacionados com coeficiente de correlação de 0,68. As medidas MSE, ϵ e C_g^* possuem correlação significativa em relação à complexidade aditiva com nível de significância de 5%, porém MSE e ϵ possuem correlação negativa com a complexidade aditiva e C_g^* possui correlação positiva com a complexidade aditiva. Ou seja, quanto maior o número de adições maior é o C_g^* , e quanto menor é o número de adições maior é o MSE e o ϵ .

Foram consideradas regressões entre as medidas de avaliação e a complexidade aditiva. Quatro modelos de regressão linear simples são considerados, onde em todos eles a complexidade aditiva é o regressor. Em cada um dos modelos a variável dependente considerada é uma das medidas de avaliação. Tais resultados são apresentados na Tabela 2.7. A partir destes resultados é possível concluir que o aumento de uma adição impacta no aumento médio de 0,066 em relação ao C_g^* . O η tem o aumento médio de 0,325 ao utilizar uma transformada aproximada que necessite uma adição a mais para o seu cômputo. Ao utilizar uma transformada que necessite de uma adição a mais o MSE terá um decréscimo esperado de 0,822. Finalmente, para o ϵ , ao aumentar uma adição o ϵ irá diminuir em média 0,003.

Nos próximos capítulos serão propostas novas aproximações para a DCT. Essas aproximações serão obtidas através da parametrização das transformadas que foram discutidas na Seção 2.6 e de outra parametrização baseada nas transformadas de comprimento 16 propostas em [28, 37]. Após a obtenção de uma transformada multiparamétrica serão considerados valores dos parâmetros que pertencem ao conjunto $\{0, \pm 0,5, \pm 1, \pm 2\}$. As transformadas obtidas serão transformadas concorrentes com as transformadas discutidas nas Seções 2.5 e 2.6 e que tem suas complexidades aritméticas expressas na Tabela 2.5.

Tabela 2.5: Custo aritmético e medidas de avaliação das aproximações consideradas.

Aproximação	ϵ	MSE	C_g^*	η	Mult.	Ad.	Desloc. bit
MRDCT [24]	8,66	0,06	7,33	80,90	0	14	0
BBM [43]	6,85	0,03	7,91	85,64	0	16	0
BAS ₅ (0) [27]	26,86	0,07	7,91	85,64	0	16	0
FW ₁ [26]	7,73	0,06	7,54	81,99	0	16	2
BAS ₅ (1) [27]	26,86	0,07	7,91	85,38	0	18	0
BAS ₃ [36]	6,85	0,03	7,91	85,38	0	18	0
CBT ₇ [33]	3,31	0,02	7,81	83,08	0	18	0
FW ₂ [26]	8,66	0,06	7,37	81,18	0	18	0
FW ₆ [26]	3,32	0,02	6,05	83,08	0	18	0
BAS ₁ [34]	5,93	0,02	8,12	86,86	0	18	2
BAS ₅ (1/2) [27]	26,40	0,07	8,12	86,86	0	18	2
FW ₃ [26]	7,73	0,06	7,58	82,27	0	20	2
FW ₄ [26]	7,53	0,05	7,56	82,70	0	20	6
FW ₅ [26]	7,41	0,05	7,58	83,08	0	20	10
BAS ₂ [35]	4,19	0,02	6,27	83,17	0	21	0
RDCT [25]	1,79	0,01	8,18	87,43	0	22	0
CBT ₁ [33]	8,60	0,04	8,14	86,80	0	22	4
CBT ₂ [33]	1,79	0,01	8,14	86,80	0	22	6
SDCT [23]	3,32	0,02	7,79	82,62	0	24	0
BAS ₆ [28]	35,06	0,10	7,95	85,31	0	24	0
CBT ₄ [33]	1,79	0,01	8,18	87,16	0	24	0
Aproximação de nível 1 (LO) [41]	0,87	0,01	8,39	88,70	0	24	2
BAS ₄ [37]	4,09	0,02	8,33	88,22	0	24	4
CBT ₅ [33]	1,79	0,01	8,14	86,54	0	24	4
Aproximação para RF imaging [29]	2,41	0,10	6,67	60,36	0	24	6
ABM [47]	1,22	0,01	8,63	90,46	0	24	6
CBT ₈ [33]	0,58	0,00	8,48	89,70	0	28	10
CBT ₉ [33]	0,58	0,00	8,48	89,70	0	28	12
CBT ₃ [33]	0,53	0,00	8,50	91,01	0	30	16
CBT ₆ [33]	0,59	0,00	8,53	89,37	0	32	12
DCT (Loeffler <i>et al.</i>) [20]	0	0	8,83	93,99	11	29	0
DCT (Chen) <i>et al.</i> [18]	0	0	8,83	93,99	16	26	0
DCT (Feig e Winograd) [21]	0	0	8,83	93,99	22	28	0
DCT (pela definição) [1]	0	0	8,83	93,99	64	56	0

Tabela 2.6: Correlação de Pearson entre as medidas de avaliação e o número de adições exigidos por cada aproximação. Entre parênteses o p-valor.

	Ad.	C_g^*	η	MSE	ϵ
Ad.	-	-	-	-	-
C_g^*	0,47 (0,01)	-	-	-	-
η	0,28 (0,13)	0,68 (0,00)	-	-	-
MSE	-0,49 (0,01)	-0,39 (0,03)	-0,62 (0,00)	-	-
ϵ	-0,40 (0,03)	-0,04 (0,84)	-0,03 (0,89)	0,75 (0,00)	-

Tabela 2.7: Estimativa dos parâmetros para análises de regressão entre as medidas de avaliação e a complexidade aditiva.

	ϵ	MSE	C_g^*	η
Parâmetro estimado do regressor	-0,003	-0,822	0,066	0,352
Intercepto estimado	25,022	0,104	6,411	77,328

PARAMETRIZAÇÃO DAS BAS

3.1 Introdução

A DCT de comprimento 8 tem grande importância, pois é empregada no padrão JPEG [12]. Já a DCT de comprimento 16 tem como importância a aplicação em padrões de vídeo, como o HEVC [14]. Desta forma, torna-se relevante o estudo de transformadas aproximadas devido ao gargalo computacional em aplicações que necessitam o processamento de quantidades massivas de dados, severa limitação de recursos e limitação em potência. Tais restrições existem em: transmissão e processamento de vídeo em tempo real [61, 62], sistemas de comunicação [7], aplicações de computação móvel [7], redes de antenas inteligentes de rádio-frequência [29] e redes de sensores sem fio [64].

Dado este fato, serão propostas duas novas classes parametrizadas de transformadas discretas a partir da série de aproximações BAS. As classes propostas endereçarão transformadas de comprimento 8 e 16, respectivamente. Adicionalmente, buscar-se-á por algoritmos rápidos para as transformadas contidas nas classes propostas. As classes propostas serão dependentes de parâmetros, e de acordo com os valores desses parâmetros são geradas diversas transformadas, o que nos leva a um problema de otimização que tenha como resultado aproximações com o melhor desempenho em compressão de imagens. Neste problema de otimização, será levado em conta o custo aritmético da transformada e algumas medidas amplamente conhecidas no contexto de compressão de imagens.

Todas as transformadas encontradas também serão escalonadas pelo método de Jridi, Alfalou e Meher (JAM) [48], ou seja, serão geradas transformadas de comprimento $2N$ a partir de transformadas de comprimento N . O método JAM é uma alternativa para obtenção de transformadas de

comprimentos maiores. Tal fato é de grande importância, pois no padrão de vídeo HEVC [14] são necessárias transformadas de comprimento 8, 16 e 32. São poucos os estudos que trabalham com transformadas de complexidade multiplicativa nula de comprimento 32, como em [28, 37, 47, 48].

As novas transformadas encontradas serão comparadas com transformadas já arquivadas na literatura em relação a medidas de avaliação da transformada (erro de energia total, erro quadrático médio, ganho de codificação e eficiência). Desta forma, serão selecionadas as melhores transformadas para diferentes complexidades aritméticas.

3.2 Parametrização de comprimento 8

A parametrização da série de aproximações BAS é feita a partir de uma inspeção dos valores das entradas das matrizes das transformadas ortogonais da série de aproximações BAS, ou seja, das matrizes das transformadas BAS_1 , BAS_3 , BAS_4 , $BAS_5(a)$ e BAS_6 . A transformada BAS_2 é não ortogonal e parece ter outra natureza das demais transformadas, portanto, não foi considerada na parametrização. A fim de manter simetria e padrões, para realizar a parametrização, a matriz da transformada BAS_4 teve a segunda, a quarta, a sétima e a oitava linhas multiplicadas por $1/2$. A matriz da transformada $BAS_5(a)$ teve um inversão entre a sétima e a oitava linha. Foram observados os padrões, e as entradas que são fixas em todas transformadas se mantiveram iguais. Já as entradas que mudam com os mesmos padrões entre as diferentes transformadas se tornaram diferentes parâmetros de acordo com os diferentes padrões de mudança. Desta forma, a classe proposta é dada pela seguinte matriz paramétrica:

$$\mathbf{T}_8(\mathbf{a}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & a_1 & a_1 & -a_1 & -a_1 & -1 & -1 \\ 1 & a_2 & -a_2 & -1 & -1 & -a_2 & a_2 & 1 \\ a_1 & a_3 & -a_4 & -a_1 & a_1 & a_4 & -a_3 & -a_1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ a_5 & -a_5 & -a_1 & a_6 & -a_6 & a_1 & a_5 & -a_5 \\ a_2 & -1 & 1 & -a_2 & -a_2 & 1 & -1 & a_2 \\ a_7 & -a_6 & a_1 & -a_8 & a_8 & -a_1 & a_6 & -a_7 \end{bmatrix}, \quad (3.1)$$

em que $\mathbf{a} = [a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8]^T$ é o vetor de parâmetros da classe parametrizada BAS de comprimento 8.

A depender do valor do vetor paramétrico \mathbf{a} , a matriz dada na Equação (3.1) gera diferentes matrizes. Entre o conjunto de transformadas geradas estão as transformadas ortogonais da série BAS, que são casos particulares da classe proposta. Espera-se que variando o valor do vetor paramétrico \mathbf{a}

surjam novas boas transformadas.

A matriz parametrizada nos leva naturalmente ao seguinte mapeamento:

$$\begin{aligned} \text{BAS}_8 : \mathbb{R}^8 &\longrightarrow \mathcal{M}_8(\mathbb{R}) \\ \mathbf{a} &\longrightarrow \mathbf{T}_8(\mathbf{a}), \end{aligned}$$

em que $\mathbf{a} = [a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8]^\top \in \mathbb{R}^8$ e $\mathcal{M}_8(\mathbb{R})$ denota o grupo das matrizes de ordem 8 com elementos em \mathbb{R} .

Como pôde ser visto anteriormente, cada vetor $\mathbf{a} \in \mathbb{R}^8$ fornece uma matriz de baixa complexidade e quando normalizadas poderão ser aproximações para a DCT exata, dada pela matriz parametrizada dada em (3.1). Na Tabela 3.1, são apresentadas as matrizes que são casos particulares de parametrização BAS.

Tabela 3.1: Casos particulares da classe parametrizada BAS.

\mathbf{a}	Transformada
$[0, 1/2, 0, 1, 1, 0, 0, 1]^\top$	BAS ₁
$[0, 1, 0, 1, 1, 0, 0, 1]^\top$	BAS ₃
$[1, 1/2, 1/2, 1/2, 1, 1, 1/2, 1/2]^\top$	BAS ₄
$[0, a, 0, -1, 0, 1, 1, 0]^\top$	BAS ₅ (a)
$[1, 1, 1, 1, 1, 1, 1, 1]^\top$	BAS ₆

3.2.1 Algoritmo rápido

Fatorando a matriz parametrizada em (3.1) de uma maneira semelhante a decimação na frequência de Cooley-Tukey base-2 [53], teremos um algoritmo rápido geral para a classe de transformadas parametrizada. Essa fatoração é dada da seguinte maneira:

$$\mathbf{T}_8(\mathbf{a}) = \mathbf{PK}(\mathbf{a})\mathbf{A}_2\mathbf{A}_1,$$

em que:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix},$$

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

$$\mathbf{K}(\mathbf{a}) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & a_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 & a_1 & 1 & 1 \\ 0 & 0 & 0 & 0 & a_6 & -a_1 & -a_5 & a_5 \\ 0 & 0 & 0 & 0 & -a_1 & -a_4 & a_3 & a_1 \\ 0 & 0 & 0 & 0 & -a_8 & a_1 & -a_6 & a_7 \end{bmatrix},$$

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

em que \mathbf{P} é uma matriz de permutação. Tal fatoração gera matrizes com estrutura “borboleta”. Matrizes com essa estrutura tem a complexidade aritmética reduzida. Além das matrizes com estrutura “borboleta”, essa fatoração irá gerar uma matriz bloco diagonal que carregará a maior parte da complexidade aritmética. No caso da fatoração da matriz parametrizada proposta, a matriz bloco diagonal criada é a matriz $\mathbf{K}(\mathbf{a})$, de forma semelhante ao subespaço Feig-Winograd em [26].

O algoritmo rápido pode ser representado pelo diagrama de fluxo e sinal da Figura 3.1. Nesta representação cada seta será chamada de ramo e o encontro entre dois ramos será chamado de nó. Cada ramo representa uma multiplicação não-trivial [53], mas como nesse trabalho serão considerados apenas elementos do conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$, ramos representarão multiplicações triviais e não serão computadas. Quando dois ramos se encontram e formam um nó, será calculada a soma dos elementos desses ramos [53]. Representar os algoritmos rápidos dessa forma pode ser muito útil pois quando deseja-se implementar essas matrizes em *hardware* é oferecida uma boa visualização dos circuitos necessários.

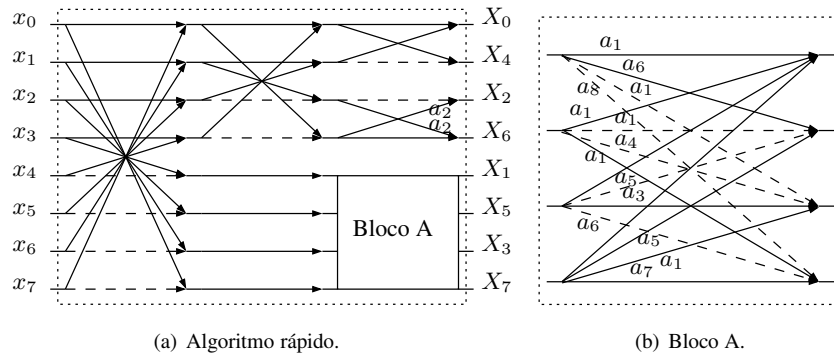


Figura 3.1: Caso geral do algoritmo rápido de comprimento 8, relacionando os dados de entrada \mathbf{x}_n , $n = 0, 1, 2, \dots, 7$, aos coeficientes da saída \mathbf{X}_k , $k = 0, 1, 2, \dots, 7$. Setas tracejadas representam multiplicação por -1 .

3.2.2 Complexidade computacional

A complexidade computacional da matriz parametrizada proposta é dada pela complexidade aritmética do algoritmo rápido. Considerando que os valores de \mathbf{a} sejam os valores do conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$, serão necessárias apenas somas e deslocamentos de *bit* para o cômputo da complexidade aritmética e a classe de matrizes proposta será finita e livre de multiplicações em sua complexidade aritmética.

De acordo com o algoritmo rápido dado na Figura 3.1, a complexidade aditiva $\mathcal{A}(\mathbf{a})$ e de deslocamento de *bit* $\mathcal{S}(\mathbf{a})$ são dadas pelas seguintes expressões, respectivamente:

$$\mathcal{A}(\mathbf{a}) = 28 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [6, 2, 1, 1, 2, 2, 1, 1]^\top$ e $\mathcal{I}_{\{.\}}$ é a função indicadora.

Esse algoritmo rápido é geral e não computa a complexidade aditiva mínima para todas as transformadas da classe parametrizada proposta, podendo ser simplificado, e a partir de nove restrições podem ser gerados algoritmos rápidos mais eficientes de acordo com o valor dos parâmetros. Dadas essas restrições, a matriz $\mathbf{T}(\mathbf{a})$ pode ser fatorada de maneira mais simplificada que a forma geral e as matrizes \mathbf{A}_2 e $\mathbf{K}(\mathbf{a})$ serão alteradas. Essas nove restrições e suas complexidades aditivas e de deslocamentos de *bit* são descritas da seguinte maneira:

1. $\pm a_1 = \pm a_4 = \pm a_6 = \pm a_8$:

$$\mathcal{A}(\mathbf{a}) = 26 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [6, 2, 1, 0, 2, 0, 1, 0]^\top$;

2. $(\pm a_1 = \pm a_3 = \pm 1) \wedge (\pm a_5 = \pm a_6 = \pm a_8)$:

$$\mathcal{A}(\mathbf{a}) = 26 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [0, 2, 0, 1, 3, 0, 1, 0]^\top$;

3. $(\pm a_1 = \pm 1) \wedge (\pm a_5 = \pm a_6) \wedge (\pm a_7 = \pm a_8)$:

$$\mathcal{A}(\mathbf{a}) = 26 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [0, 2, 1, 1, 3, 0, 1, 0]^\top$;

4. $(\pm a_1 = \pm a_5 = \pm a_6 = \pm 1) \wedge (\pm a_3 = \pm a_4)$:

$$\mathcal{A}(\mathbf{a}) = 26 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [0, 2, 1, 0, 0, 0, 1, 1]^\top$;

5. $\pm a_1 = \pm a_4 = \pm a_5 = \pm a_7 = \pm 1$:

$$\mathcal{A}(\mathbf{a}) = 26 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [0, 2, 1, 0, 0, 2, 0, 1]^\top$;

6. $(\pm a_1 = \pm a_3) \wedge (\pm a_6 = \pm a_7)$:

$$\mathcal{A}(\mathbf{a}) = 26 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [6, 2, 0, 1, 1, 2, 0, 1]^\top$;

7. $\pm a_1 = \pm a_3 = \pm a_4 = \pm a_6 = \pm a_7 = \pm a_8$:

$$\mathcal{A}(\mathbf{a}) = 24 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [6, 2, 0, 0, 1, 0, 0, 0]^\top$;

8. $\pm a_1 = \pm a_3 = \pm a_4 = \pm a_5 = \pm a_6 = \pm a_7 = \pm a_8 = \pm 1$:

$$\mathcal{A}(\mathbf{a}) = 24 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [0, 2, 0, 0, 0, 0, 0, 0]^\top$;

9. $(\pm a_1 = \pm a_5 = \pm a_6 = \pm 1) \wedge (\pm a_3 = \pm a_4) \wedge (\pm a_7 = \pm a_8)$:

$$\mathcal{A}(\mathbf{a}) = 24 - \sum_{i=1}^8 w_i \mathcal{I}_{\{0\}}(a_i),$$

$$\mathcal{S}(\mathbf{a}) = \sum_{i=1}^8 w_i \mathcal{I}_{\{\frac{1}{2}, 2\}}(a_i),$$

em que $\mathbf{w} = [0, 2, 1, 0, 0, 0, 1, 0]^\top$.

De acordo com as expressões que dão a complexidade aritmética das transformadas da classe parametrizada temos que $16 \leq \mathcal{A}(\mathbf{a}) \leq 28$ e $0 \leq \mathcal{S}(\mathbf{a}) \leq 16$. Desta forma, o limite inferior para a complexidade aritmética das transformadas da classe parametrizada de comprimento 8 é de 16 adições e o limite superior é de 28 adições e 16 deslocamentos de *bit*.

3.2.3 Ortogonalidade e ortonormalidade

Dada a classe parametrizada de matrizes, as aproximações ortonormais para a DCT são dadas por [24, 25, 29, 67]:

$$\hat{\mathbf{C}}(\mathbf{a}) = \mathbf{S}(\mathbf{a}) \cdot \mathbf{T}_8(\mathbf{a}),$$

em que

$$\mathbf{S}(\mathbf{a}) = \sqrt{\left[\mathbf{T}_8(\mathbf{a}) \cdot \mathbf{T}_8(\mathbf{a})^\top\right]^{-1}}.$$

Para a classe parametrizada, temos:

$$\mathbf{T}_8(\mathbf{a}) \cdot \mathbf{T}_8(\mathbf{a})^\top = \begin{bmatrix} 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau_1 & 0 & \tau_6 & 0 & \tau_7 & 0 & \tau_8 \\ 0 & 0 & \tau_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau_6 & 0 & \tau_3 & 0 & \tau_9 & 0 & \tau_{10} \\ 0 & 0 & 0 & 0 & 8 & 0 & 0 & 0 \\ 0 & \tau_7 & 0 & \tau_9 & 0 & \tau_4 & 0 & \tau_{11} \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_2 & 0 \\ 0 & \tau_8 & 0 & \tau_{10} & 0 & \tau_{11} & 0 & \tau_5 \end{bmatrix},$$

em que

$$\left\{ \begin{array}{l} \tau_1 = 4a_1^2 + 4, \\ \tau_2 = 4a_2^2 + 4, \\ \tau_3 = 4a_1^2 + 2a_3^2 + 2a_4^2, \\ \tau_4 = 2a_6^2 + 4a_5^2 + 2a_1^2, \\ \tau_5 = 2a_8^2 + 2a_7^2 + 2a_6^2 + 2a_1^2, \\ \tau_6 = 2a_1 - 2a_1^2 + 2a_3 - 2a_1a_4, \\ \tau_7 = 2a_1a_6 - 2a_1^2, \\ \tau_8 = 2a_1^2 - 2a_6 + 2a_7 - 2a_1a_8, \\ \tau_9 = 2a_1a_4 + 2a_1a_5 - 2a_3a_5 - 2a_1a_6, \\ \tau_{10} = 2a_1a_8 + 2a_1a_7 - 2a_3a_6 - 2a_1a_4, \\ \tau_{11} = 2a_5a_7 + 2a_5a_6 - 2a_1^2 - 2a_6a_8. \end{array} \right.$$

Para garantia de ortogonalidade precisamos que $\tau_6, \tau_7, \tau_8, \tau_9, \tau_{10}$ e τ_{11} sejam iguais a zero.

Dada a ortogonalidade de uma matriz, $\mathbf{S}(\mathbf{a})$ é uma matriz diagonal e a aproximação ortonormal $\hat{\mathbf{C}}(\cdot)$ também será de baixa complexidade, pois a matriz $\mathbf{S}(\mathbf{a})$ será suprimida na etapa de quantização

[24, 25, 27, 31, 34, 36, 41]. Para a classe parametrizada, temos:

$$\mathbf{S}(\mathbf{a}) = \text{diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{\tau_1}}, \frac{1}{\sqrt{\tau_2}}, \frac{1}{\sqrt{\tau_3}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{\tau_4}}, \frac{1}{\sqrt{\tau_2}}, \frac{1}{\sqrt{\tau_5}} \right).$$

Os valores de \mathbf{a} que garantem ortogonalidade são os que resolvem o seguinte sistema não linear de equações:

$$\begin{cases} a_1 - a_1^2 + a_3 - a_1 a_4 & = 0 \\ a_1 a_6 - a_1^2 & = 0 \\ a_1^2 - a_6 + a_7 - a_1 a_8 & = 0 \\ a_1 a_4 + a_1 a_5 - a_3 a_5 - a_1 a_6 & = 0 \\ a_1 a_8 + a_1 a_7 - a_3 a_6 - a_1 a_4 & = 0 \\ a_5 a_7 + a_5 a_6 - a_1^2 - a_6 a_8 & = 0 \end{cases}. \quad (3.2)$$

As soluções do sistema de equações dado em (3.2) são dadas por,

$$\begin{aligned} & \left\{ \{a_1 = a_3 = a_6 = a_7 = 0, (a_4, a_5, a_8 \in \mathbb{R})\}, \right. \\ & \{a_1 = 0, a_7 = a_6, a_8 = 2a_5, (a_4, a_5 \in \mathbb{R})\}, \\ & \{a_1 = a_5 = a_6 = 1, a_3 = a_4 = a_7 = a_8\}, \\ & \left. \left\{ a_3 = a_8 = \frac{(a_1 - 1)a_1}{a_1 + 1}, a_4 = a_7 = \frac{a_1 - a_1^2}{a_1 + 1}, a_5 = a_1^2, a_6 = a_1 \right\} \right\}. \quad (3.3) \end{aligned}$$

Pelas soluções dadas em (3.3), nota-se que o sistema de equações dado em (3.2) tem infinitas soluções.

Quando uma matriz é de baixa complexidade, mas não é uma matriz ortogonal, não temos garantia que sua inversa será de baixa complexidade. Por esse motivo é desejável ortogonalidade, pois como a inversa de uma matriz ortogonal é a sua transposta, temos a garantia que a inversa de uma matriz ortogonal é de baixa complexidade.

Uma medida para desvio de diagonalidade de uma matriz \mathbf{T} é dada por [73]:

$$\delta(\mathbf{T}) = 1 - \frac{\|\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)\|_F^2}{\|\mathbf{T} \cdot \mathbf{T}^\top\|_F^2},$$

em que $\|\cdot\|_F$ é a norma matricial de Frobenius [74]. Temos $0 \leq \delta(\cdot) \leq 1$, quando $\delta(\cdot)$ é zero \mathbf{T} é uma matriz ortogonal, e quanto mais distante de zero for esse valor, maior é o desvio de diagonalidade. Quanto maior for o desvio de diagonalidade, maior será a quantidade de elementos fora da diagonal da matriz $\mathbf{S}(\mathbf{a})$. A SDCT e a aproximação BAS_2 são não-ortogonais, são boas aproximações para a DCT, e têm medida de diagonalidade de 0,2 e 0,1774, respectivamente. Quando o valor

de $\delta(\cdot)$ é baixo, menor será a perda dos elementos fora da matriz ajuste e melhor será a qualidade da aproximação. Quando o valor de $\delta(\cdot)$ for muito alto terá muita perda pela grande quantidade de elementos fora da matriz de ajuste e não resultará em uma boa aproximação para a DCT.

Na Seção 3.4 será feita uma busca na classe parametrizada proposta. Nessa busca serão consideradas as matrizes da classe proposta, a complexidade aditiva, e medidas de qualidade da transformada. Assim espera-se encontrar novas transformadas com um bom desempenho em compressão de imagem pertencentes a classe proposta.

3.3 Parametrização de comprimento 16

A melhor de nosso conhecimento, não existe na literatura nenhuma parametrização de aproximações da DCT de comprimento 16. Portanto, iremos parametrizar as transformadas de comprimento 16 propostas por Bouguezel, Ahmed e Swamy. Para essa parametrização foram generalizadas as transformadas propostas em [28, 37]. Também foram feitas permutações nas linhas da transformada BAS proposta em [37], com intuito de facilitar a parametrização. A parametrização BAS de comprimento 16 é dada por:

$$\mathbf{T}_{16}(a) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ a & -a & -1 & 1 & -1 & 1 & a & -a & a & -a & -1 & 1 & -1 & 1 & a & -a \\ a & 1 & -1 & -a & -a & -1 & 1 & a & a & 1 & -1 & -a & -a & -1 & 1 & a \\ a & a & 1 & 1 & -1 & -1 & -a & -a & a & a & 1 & 1 & -1 & -1 & -a & -a \\ 1 & 1 & -a & -a & a & a & -1 & -1 & 1 & 1 & -a & -a & a & a & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ a & 1 & -1 & -a & a & 1 & -1 & -a & -a & -1 & 1 & a & -a & -1 & 1 & a \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & -a & a & -1 & -1 & a & -a & 1 & 1 & -a & a & -1 & -1 & a & -a & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\ 1 & -a & a & -1 & 1 & -a & a & -1 & -1 & a & -a & 1 & -1 & a & -a & 1 \\ 1 & -1 & a & -a & a & -a & 1 & -1 & 1 & -1 & a & -a & a & -a & 1 & -1 \end{bmatrix}. \quad (3.4)$$

A matriz dada em (3.4) nos leva naturalmente ao seguinte mapeamento:

$$\begin{aligned} \text{BAS}_{18} : \mathbb{R} &\longrightarrow \mathcal{M}_{16}(\mathbb{R}) \\ a &\longrightarrow \mathbf{T}_{16}(a), \end{aligned}$$

em que $a \in \mathbb{R}$ e $\mathcal{M}_{16}(\mathbb{R})$ denota o grupo das matrizes de ordem 16 com elementos em \mathbb{R} .

Note que a matriz dada em (3.4) tem apenas um parâmetro. Desta forma, essa matriz parametrizada, considerando as entradas em $\{0, \pm 1/2, \pm 1, \pm 2\}$, terá apenas 7 matrizes e suas versões permutadas. As versões permutadas são todas as permutações possíveis entre as linhas da matriz dada em (3.4). Serão sugeridas permutações mais eficientes em relação a compactação de energia de acordo com o método a seguir.

3.3.1 Concentração de energia

A DCT tem a propriedade de descorrelação de energia do vetor transformado que é herdada da KLT. Portanto, espera-se que as aproximações da DCT, também tenham essa propriedade. Ao aplicar a transformada, cada linha da matriz da transformada retém um percentual de energia diferente. Para maior concentração de energia do sinal transformado, espera-se que cada linha da transformada retenha energia em uma ordem decrescente, ou seja, a primeira linha retém mais energia que a segunda, a segunda retém mais energia que a terceira, até a linha $N - 1$ que retém mais energia que a linha N . Desta forma, ao aplicar a transformada em um sinal, o sinal transformado concentrará a energia nos primeiros coeficientes. Ao criar a matriz parametrizada dada em (3.4), foram necessárias permutações nas linhas da transformada BAS proposta em [37], com isso a concentração de energia das linhas das matrizes da classe parametrizada dada em (3.4) foi alterada. A concentração de energia é dada por:

$$\mathbf{E} = \frac{\mathbf{V} \odot \mathbf{V}}{\langle \mathbf{V}, \mathbf{V}^T \rangle},$$

em que

$$\mathbf{V} = \mathbf{T} \cdot \mathbf{v},$$

dado que \odot é a multiplicação elemento a elemento de uma matriz, $\langle \cdot, \cdot \rangle$ é o produto interno de duas matrizes, $\mathbf{v} = [v_1, v_2, \dots, v_N]^T$ é um sinal que segue um processo Markoviano de primeira ordem, \mathbf{T} é a transformada de interesse e $\mathbf{E} = [E_1, E_2, \dots, E_N]^T$ é um vetor que armazena a concentração de energia de cada linha.

Para sugerir uma nova permutação para as linhas da transformada é proposto o seguinte algoritmo:

1. Gera-se R sinais aleatórios de acordo com um processo Markoviano de primeira ordem com $\rho = 0,95$;
2. Para cada sinal gerado, calcula-se a concentração de energia das linhas da transformada;
3. Calcula-se a média da concentração de energia de cada linha;
4. Estima-se o percentual de concentração de energia de cada linha;
5. Gera-se uma matriz de permutação de acordo com o percentual de energia armazenado em cada linha.

Desta forma, a linha que concentra mais energia será a primeira linha, a segunda linha será a que concentra mais energia com exceção da primeira linha, até a última linha que é a linha que concentra

menos energia. Quanto maior o valor de R maior será a precisão da estimativa de Monte Carlo da concentração de energia de cada linha da transformada.

3.3.2 Algoritmo rápido

Para obtenção de um algoritmo rápido geral para a classe de transformadas parametrizada, fatorase a matriz parametrizada $\mathbf{T}_{16}(a)$ de uma maneira semelhante a decimação na frequência de Cooley-Tukey base-2 [53]. Tal fatoração, a menos da matriz de permutação, é dada da seguinte maneira:

$$\mathbf{T}_{16}(a) = \mathbf{PK}(a)\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1,$$

em que:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{I}_8 & \mathbf{I}_8 \\ \mathbf{I}_8 & -\mathbf{I}_8 \end{bmatrix},$$

$$\mathbf{A}_2 = \text{diag} \left(\begin{bmatrix} \mathbf{I}_4 & \bar{\mathbf{I}}_4 \\ \bar{\mathbf{I}}_4 & -\mathbf{I}_4 \end{bmatrix}, \begin{bmatrix} \mathbf{I}_4 & \mathbf{I}_4 \\ \mathbf{I}_4 & -\mathbf{I}_4 \end{bmatrix} \right),$$

$$\mathbf{A}_3 = \text{diag} \left(\begin{bmatrix} \mathbf{I}_2 & \bar{\mathbf{I}}_2 \\ \bar{\mathbf{I}}_2 & -\mathbf{I}_2 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} \mathbf{I}_2 & \bar{\mathbf{I}}_2 \\ \bar{\mathbf{I}}_2 & -\mathbf{I}_2 \end{bmatrix}, \begin{bmatrix} \mathbf{I}_2 & \mathbf{I}_2 \\ \mathbf{I}_2 & -\mathbf{I}_2 \end{bmatrix} \right),$$

$$\mathbf{K}(a) = \text{diag} \left(\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & a \\ -a & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & 0 & -a \\ -a & 0 & 1 & 0 \\ 0 & -a & 0 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & a \\ -a & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right),$$

\mathbf{I}_N é a matriz identidade de ordem N , $\bar{\mathbf{I}}_N$ é a matriz contra-identidade de ordem N , que é dada da seguinte forma,

$$\bar{\mathbf{I}}_N = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix},$$

e \mathbf{P} é uma matriz de permutação proveniente do algoritmo proposto em 3.3.1. A concentração de energia referente a cada linha mudou com as permutações das linhas da transformada BAS proposta em [37]. A Figura 3.2 ilustra o algoritmo rápido em forma de diagrama de fluxo e sinal.

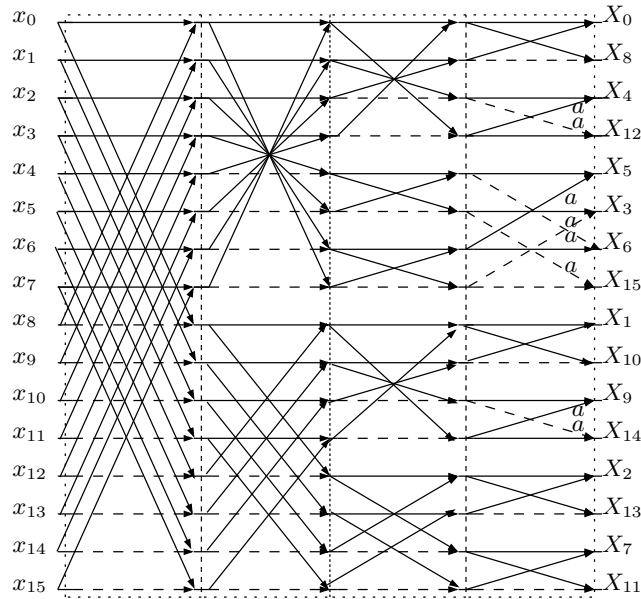


Figura 3.2: Algoritmo rápido da matriz parametrizada BAS de comprimento 16, relacionando os dados de entrada \mathbf{x}_n , $n = 0, 1, 2, \dots, 15$, aos coeficientes da saída \mathbf{X}_k , $k = 0, 1, 2, \dots, 15$. Setas tracejadas representando multiplicação por -1 .

3.3.3 Complexidade computacional

Como já discutido, a complexidade computacional da classe parametrizada com comprimento 16 é estimada por meio por sua complexidade aritmética. Considerando o algoritmo rápido dado na Figura 3.2, a complexidade aditiva e de deslocamento de *bit* é dada da seguinte maneira, respectivamente:

$$\mathcal{A}(a) = 64 - w\mathcal{I}_{\{0\}}(a),$$

$$\mathcal{S}(a) = w\mathcal{I}_{\{\frac{1}{2}, 2\}}(a),$$

em que $w = 8$. É fácil observar que as matrizes da classe parametrizada terão complexidade de 56 adições, 64 adições, ou 64 adições e 8 deslocamentos de *bit*.

3.3.4 Ortogonalidade e ortonormalidade

Para a classe parametrizada de comprimento 16 temos:

$$\mathbf{T}_{16}(a) \cdot \mathbf{T}_{16}(a)^\top = \text{diag}(8, 8, 8, \tau, \tau, \tau, \tau, 8, 8, \tau, 8, 8, \tau, 8, \tau),$$

em que $\tau = 8a^2 + 8$. Nota-se que todas as matrizes da classe parametrizada de comprimento 16 são ortogonais, assim a sua inversa é dada pela transposta. Para uma aproximação ortonormal temos:

$$\mathbf{S}(a) = \frac{1}{2\sqrt{2}} \text{diag} \left(1, 1, 1, 2\sqrt{\frac{2}{\tau}}, 2\sqrt{\frac{2}{\tau}}, 2\sqrt{\frac{2}{\tau}}, 2\sqrt{\frac{2}{\tau}}, 1, 1, 2\sqrt{\frac{2}{\tau}}, 1, 1, 2\sqrt{\frac{2}{\tau}}, 1, 2\sqrt{\frac{2}{\tau}}, 2\sqrt{\frac{2}{\tau}} \right).$$

3.4 Resultados da Busca

Nesta seção, será explorado o problema de otimização multicritério [45, 46] e a ordem de Pareto [46]. Dado isso, serão encontradas as transformadas ótimas das classes parametrizadas dados em (3.1) e (3.4). As transformadas ótimas encontradas serão comparadas com as transformadas concorrentes já arquivadas na literatura.

3.4.1 Otimização multicritério

Aqui iremos considerar um problema de otimização considerando os seguintes critérios:

- ▷ Erro quadrático médio (MSE) [7];
- ▷ Erro de energia total (ϵ) [25];
- ▷ Ganho de codificação unificado (C_g^*) [38];
- ▷ Eficiência (η) [42].

No contexto de compressão de imagens, algumas transformadas podem ter bons valores em certas medidas particulares e valores não tão bons em outras medidas. Consideramos um problema de otimização utilizando todas essas medidas ao invés de considerá-las separadamente. Portanto, iremos abordar o mesmo problema de otimização multicritério considerado em [26], que consiste no seguinte problema de minimização [45, 46],

$$\min_a (\epsilon(\hat{\mathbf{C}}), \text{MSE}(\hat{\mathbf{C}}), -C_g^*(\hat{\mathbf{C}}), -\eta(\hat{\mathbf{C}})). \quad (3.5)$$

Como $C_g^*(\hat{\mathbf{C}})$ e $\eta(\hat{\mathbf{C}})$ devem ser maximizados no problema de minimização serão considerados os seus valores negativos.

3.4.2 A ordem de Pareto \mathbb{R}^p

Em problemas de otimização simples, minimizar está relacionado diretamente com a ordem canônica \leq em \mathbb{R} [46]. Em problemas de otimização multicritério temos um problema de otimização em

\mathbb{R}^p , não existindo uma ordem canônica em \mathbb{R}^p com $p \geq 2$ [46]. Para definição de mínimo em (3.5), necessitamos determinar previamente alguma ordem em \mathbb{R}^p .

Definindo que o conjunto $\mathcal{C} \subset \mathbb{R}^p$ ($p \geq 2$) é um cone, se para cada elemento $\mathbf{d} \in \mathcal{C}$ e todo número real $\alpha > 0$, satisfaz $\alpha \mathbf{d} \in \mathcal{C}$. Dado que, em geral, ordens são definidas em um cone. Consideraremos o seguinte cone:

$$\mathcal{C} = \mathbb{R}_+^p = \{(x_1, \dots, x_p) \in \mathbb{R}^p : x_i \geq 0 \forall_i \text{ e } x_j \neq 0 \text{ para algum } j\}.$$

Definimos, assim, uma nova relação binária “ \leq ” em \mathbb{R}_+^p ($p \geq 2$) por:

$$\mathbf{x} \leq \mathbf{y} \iff \mathbf{x} - \mathbf{y} \in \mathbb{R}_+^p,$$

de forma alternativa, pode ser escrito da seguinte maneira,

$$\mathbf{x} \leq \mathbf{y} \iff x_i \leq y_i \forall_i \text{ e } x_j < y_j \text{ para algum } j. \quad (3.6)$$

Tal ordem parcial do cone \mathbb{R}_+^p é conhecida como ordem de Pareto em \mathbb{R}^p [46]. Portanto, a otimalidade de Pareto [46] é definida como a otimização multicritério conceituada com respeito a esta ordem.

3.4.3 Solução do problema de otimização

Para resolver o problema de otimização proposto na Equação (3.5), consideraremos a ordem de Pareto [46] dada em (3.6). Considerando que as entradas de \mathbf{a} são os valores do conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$ para a parametrização de comprimento 8, teremos $7^8 = 5764801$ matrizes a serem consideradas no problema de otimização. Já na parametrização de comprimento 16 teremos $7^{16} = 7$ matrizes a serem investigadas.

Para cada uma das matrizes serão calculadas as medidas de erro de energia total, erro quadrático médio, ganho de codificação unificado e eficiência da transformada. As matrizes selecionadas serão as matrizes que minimizam pelo menos uma das medidas citadas anteriormente para cada uma das complexidades aritméticas.

Desta forma, sendo o conjunto de soluções eficientes no sentido da otimalidade de Pareto [46] é definido, segundo (3.5), por:

$$\mathbf{a}^* = \arg \min_{\mathbf{a}} \theta(\hat{\mathbf{C}}(\mathbf{a})),$$

sendo $\theta(\cdot)$ uma função que representa qualquer uma das medidas utilizadas na otimização multicritério, e $\hat{\mathbf{C}}$ a transformada aproximada avaliada. Tal operação será realizada para as diferentes complexidades aditivas do conjunto de matrizes parametrizadas.

3.4.4 Resultados da busca para transformadas de comprimento 8

Na Tabela 3.2, estão as soluções ótimas referentes a otimização multicritério para a parametrização com comprimento 8.

Tabela 3.2: *Soluções das transformadas ortogonais ótimas de comprimento 8.*

\hat{C}_j	\mathbf{a}_j^*	Descrição
\hat{C}_1	$[0,0,0,1,1,0,0,1]^\top$	[43]
\hat{C}_2	$[0,1,0,1,1,0,0,1]^\top$	BAS ₃
\hat{C}_3	$[0,0,0,1,1/2,1,1,1]^\top$	Nova
\hat{C}_4	$[0,0,0,1,1,1,1,2]^\top$	Nova
\hat{C}_5	$[0,1/2,0,1,1,0,0,1]^\top$	BAS ₁
\hat{C}_6	$[1,0,0,0,1,1,0,0]^\top$	Nova
\hat{C}_7	$[0,1,0,1,1,1,1,2]^\top$	Nova
\hat{C}_8	$[0,1,0,1,1/2,1,1,1]^\top$	Nova
\hat{C}_9	$[0,1/2,0,1,1,1,1,2]^\top$	Nova
\hat{C}_{10}	$[0,1/2,0,1,1/2,1,1,1]^\top$	Nova
\hat{C}_{11}	$[1,0,1,1,1,1,1,1]^\top$	Nova
\hat{C}_{12}	$[1,1/2,0,0,1,1,0,0]^\top$	Nova
\hat{C}_{13}	$[1,0,1/2,1/2,1,1,1/2,1/2]^\top$	Nova
\hat{C}_{14}	$[1,1/2,1,1,1,1,1,1]^\top$	Nova
\hat{C}_{15}	$[1,1/2,1/2,1/2,1,1,1/2,1/2]^\top$	BAS ₄

As transformadas \hat{C}_j são as transformadas aproximadas que estão na fronteira de Pareto da matriz parametrizada proposta, com $j = 1, 2, \dots, 15$. De acordo com a Tabela 3.2, pode-se notar que as transformadas \hat{C}_j com, $j = 2, 5, 15$, são transformadas da série de aproximações BAS. A transformada \hat{C}_1 é uma transformada que foi obtida a partir de uma modificação da transformada proposta em [44] e como um dos autores da série de aproximações BAS é colaborador desse trabalho esta transformada pode ter a mesma natureza. As demais transformadas apresentadas na Tabela 3.2, a melhor de nosso conhecimento, não estão presentes na literatura.

Avaliação das transformadas ótimas

Foram calculadas as medidas de proximidade, a complexidade aritmética (\mathcal{A}) e a complexidade de deslocamento de *bit* (\mathcal{S}) para todas transformadas ótimas consideradas através da otimização multicritério. Na Tabela 3.3 estão apresentadas as medidas de proximidade e o custo aritmético das transformadas ótimas consideradas.

Tabela 3.3: Medidas de avaliação das transformadas ótimas de comprimento 8.

\hat{C}_j	$\epsilon(\mathbf{a}_j^*)$	$\text{MSE}(\mathbf{a}_j^*)$	$C_g^*(\mathbf{a}_j^*)$	$\eta(\mathbf{a}_j^*)$	$\mathcal{A}(\mathbf{a}_j^*)$	$\mathcal{S}(\mathbf{a}_j^*)$
\hat{C}_1	6,85	0,03	7,91	85,64	16	0
\hat{C}_2	6,85	0,03	7,91	85,38	18	0
\hat{C}_3	5,79	0,03	7,91	85,78	18	1
\hat{C}_4	5,05	0,03	7,91	85,51	18	1
\hat{C}_5	5,93	0,02	8,12	86,86	18	2
\hat{C}_6	6,85	0,03	7,93	85,80	20	0
\hat{C}_7	5,05	0,03	7,91	85,25	20	1
\hat{C}_8	5,79	0,03	7,91	85,52	20	1
\hat{C}_9	4,12	0,02	8,12	86,73	20	3
\hat{C}_{10}	4,87	0,02	8,12	87,01	20	3
\hat{C}_{11}	5,05	0,02	7,95	85,58	22	0
\hat{C}_{12}	5,93	0,02	8,14	87,02	22	2
\hat{C}_{13}	5,02	0,02	8,12	86,96	22	2
\hat{C}_{14}	4,12	0,02	8,15	86,79	24	2
\hat{C}_{15}	4,09	0,02	8,33	88,22	24	4

De acordo com os valores apresentados na Tabela 3.3, destaca-se que a transformada \hat{C}_{15} , entre as transformadas consideradas, possui as melhores medidas de avaliação em todas as métricas, porém é a que possui a maior complexidade aritmética. A transformada que possui menor complexidade aritmética é a transformada \hat{C}_1 . Entre as novas transformadas propostas, excluindo a transformada \hat{C}_{15} , a transformada que possui menor erro de energia total e erro quadrático médio é a transformada \hat{C}_9 , e a que possui maior ganho de codificação é a transformada \hat{C}_{14} . Já a que possui maior eficiência é a transformada \hat{C}_{12} .

As novas transformadas encontradas serão comparadas com outras aproximações que não pertencem à classe parametrizada. As transformadas consideradas são: FW_6 (18 adições), FW_5 (20 adições e 10 deslocamentos de *bit*), RDCT (22 adições), aproximação de nível 1 de Lengwehasatit e Ortega (24 adições e 2 deslocamentos de *bit*), ABM (24 adições e 6 deslocamentos de *bit*), além de todas as transformadas da série de aproximações BAS ortogonais. As medidas de avaliação das transformadas citadas e sua complexidade computacional são apresentados na Tabela 3.4. Por meio da Figura 3.3 podemos avaliar conjuntamente as medidas de qualidade com o custo das transformadas. Ainda podemos comparar alguma transformada com as transformadas concorrentes. A linha pontilhada na Figura 3.3 é a fronteira de Pareto para a classe parametrizada proposta.

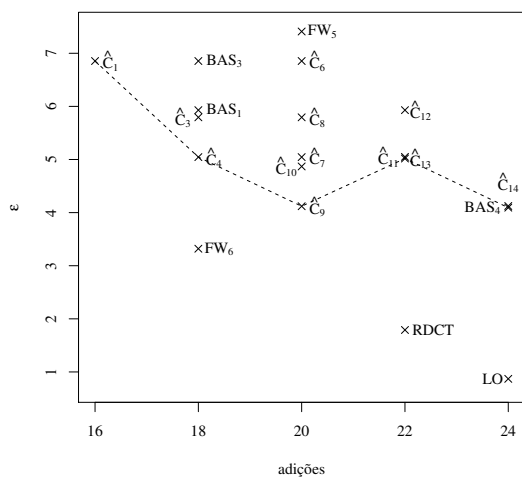
De acordo com a Figura 3.3, pode-se ver a comparação das transformadas propostas com as

Tabela 3.4: Medidas de avaliação para as transformadas da série de aproximações BAS e demais transformadas consideradas para a comparação entre as transformadas de comprimento 8.

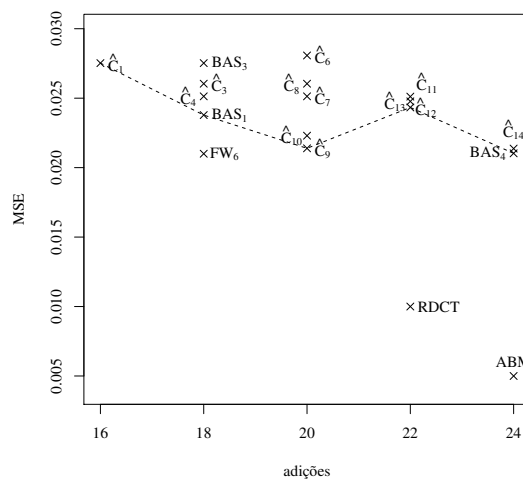
Transformada	ϵ	MSE	C_g^*	η	\mathcal{A}	\mathcal{S}
BAS ₅ (0) [27]	26,86	0,07	7,91	85,64	16	0
FW ₆ [26]	3,32	0,02	6,05	83,08	18	0
BAS ₃ [36]	6,85	0,03	7,91	85,38	18	0
BAS ₅ (1) [27]	26,86	0,07	7,91	85,38	18	0
BAS ₅ (1/2) [27]	26,40	0,07	8,12	86,86	18	2
BAS ₁ [34]	5,93	0,02	8,12	86,86	18	2
FW ₅ [26]	7,41	0,05	7,58	83,08	20	10
RDCT [25]	1,79	0,01	8,18	87,43	22	0
BAS ₆ [28]	35,06	0,10	7,95	85,31	24	0
aproximação de nível 1 (LO) [41]	0,87	0,01	8,39	88,70	24	2
BAS ₄ [37]	4,09	0,02	8,33	88,22	24	4
ABM [47]	1,22	0,01	8,63	90,46	24	6

transformadas já arquivadas na literatura. Para 16 adições a transformada proposta em [43] possui os melhores resultados em relação ao erro quadrático médio e erro de energia total, e empata com a BAS₅(0) em relação ao ganho de codificação e eficiência da transformada. Entre as transformadas que necessitam de 18 adições, a transformada FW₆ possui os melhores resultados em relação ao erro quadrático médio e erro de energia total, já em relação ao ganho de codificação e eficiência a BAS₁ possui os melhores resultados. Já para as transformadas que necessitam 20 adições, a transformada \hat{C}_9 possui as melhores medidas para erro quadrático médio, erro de energia total e ganho de codificação, já para eficiência a melhor transformada é a \hat{C}_{10} . Para 22 adições a RDCT é a melhor transformada em relação a todas as medidas. Por fim, para 24 adições a transformada de nível 1 proposta por Lengwehasatit e Ortega possui a melhor medida em relação ao erro quadrático médio e a transformada ABM possui os melhores resultados em relação as demais medidas. Nota-se que as transformadas de 16, 18 e 20 adições provenientes da série BAS parametrizada, em geral, possuem os melhores resultados em relação ao ganho de codificação e eficiência da transformada.

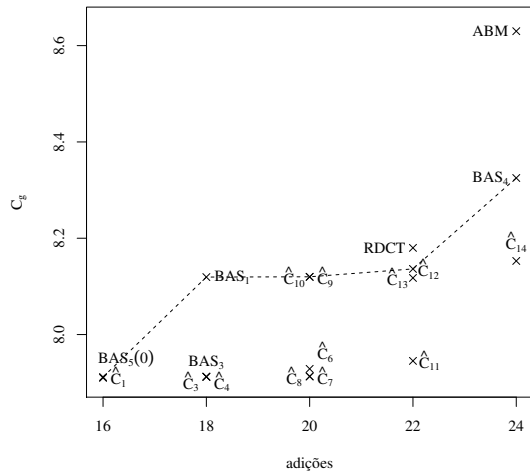
Ainda, cabe notar que a transformada BAS₅(0) é a mesma transformada proposta em [43], BAS₅(1) é a mesma BAS₃ e a transformada BAS₅(1/2) é a mesma BAS₁ a menos da seguinte matriz



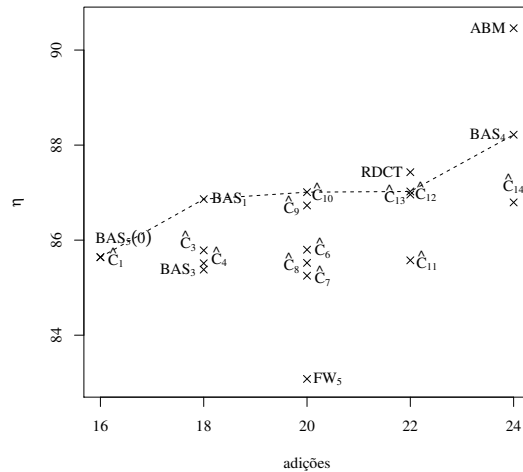
(a) Erro de energia total.



(b) Erro quadrático médio.



(c) Ganho de codificação.



(d) Eficiência da transformada.

Figura 3.3: Gráficos de avaliação das transformadas ótimas propostas de comprimento 8. A linha pontilhada é a fronteira de Pareto para a matriz parametrizada de comprimento 8 proposta.

de permutação:

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

Por este motivo, possuem os mesmos valores de ganho de codificação e eficiência e diferenciam os valores em relação às medidas de erro quadrático médio e erro de energia total que são medidas de proximidade com a DCT. Ainda, tais relações podem ser expressas da seguinte maneira,

$$\mathbf{T}_{BB} = \mathbf{P} \cdot \text{BAS}_5(0),$$

$$\text{BAS}_3 = \mathbf{P} \cdot \text{BAS}_5(1),$$

$$\text{BAS}_1 = \mathbf{P} \cdot \text{BAS}_5(1/2).$$

3.4.5 Resultados da busca para transformadas de comprimento 16

As soluções ótimas para essa parametrização são as que têm como solução $a = 0$, $a = 1$ e $a = 2$ com os seguintes esquemas de permutação cíclica: (1)(2)(3)(4 7 15 16)(5 13 10 11 12)(6 8)(9)(14), (1)(2)(3)(4 6 8 7 10 11 12)(5)(9)(13)(14)(15)(16) e (1)(2)(3)(4 6 10 11 12)(5)(7 8)(9)(13 14 15 16), respectivamente. Para $a = 1$ temos a transformada proposta em [28] e para $a = 2$ temos a transformada proposta em [37], porém são suas versões permutadas e suas medidas mudam em relação ao erro quadrático médio e erro de energia total. Para $a = 0$, com a permutação sugerida, temos uma nova transformada.

Avaliação das transformadas ótimas

Foram calculadas as medidas de proximidade e a complexidade aritmética para todas transformadas ótimas consideradas através da otimização multicritério. Na Tabela 3.5 estão apresentadas as medidas de proximidade e o custo aritmético das transformadas ótimas consideradas.

De acordo com a Tabela 3.5, nota-se que a transformada decorrente de $a = 2$ possui as melhores medidas para ganho de codificação e eficiência e para $a = 1$ possui as melhores medidas para erro de energia total e erro quadrático médio. A transformada com $a = 0$ é superada em todas as métricas, mas se aproxima muito da transformada que tem $a = 1$ em ganho de codificação e possui melhor resultado em relação a eficiência, exigindo 8 adições a menos para o seu cômputo.

Tabela 3.5: Medidas de avaliação das transformadas ótimas com comprimento 16.

a^*	ϵ	$\text{MSE}(a^*)$	$C_g^*(a^*)$	$\eta(a^*)$	$\mathcal{A}(a^*)$	$\mathcal{S}(a^*)$
0	51,87	0,09	8,17	70,80	56	0
1	17,43	0,06	8,19	70,65	64	0
2	44,83	0,08	8,52	73,63	64	8

Tabela 3.6: Medidas de avaliação para as transformadas da série de aproximações BAS e demais transformadas consideradas para a comparação entre as transformadas com comprimento 16.

Transformada	ϵ	MSE	C_g^*	η	\mathcal{A}	\mathcal{S}
SOBCM [49]	41,00	0,09	7,86	67,61	44	0
SBCKMK [50]	30,32	0,06	8,29	70,83	60	0
JAM [48]	14,74	0,05	8,43	72,23	60	0
BAS-2013 [28]	54,62	0,13	8,19	70,64	64	0
BAS-2010 [37]	16,41	0,056	8,52	73,63	64	8
ABM (JAM) [47]	13,70	0,05	8,88	76,81	64	12
BCEM [31]	8,08	0,05	7,84	65,28	72	0

Na Tabela 3.6, vemos que a transformada que requer menor custo na literatura é a SOBCM, porém possui as piores medidas de eficiência e ganho de codificação. As transformadas com $a = 1$ e $a = 2$ são as mesmas transformadas BAS-2013 e BAS-2010, respectivamente, com permutações em suas linhas e possuem os mesmos valores de ganho de codificação e eficiência. Porém, em relação as medidas de proximidade da DCT, possuem resultados diferentes devido a permutação proposta. A transformada ABM escalonada pelo método JAM possui as melhores medidas em relação a ganho de codificação e eficiência e a transformada BCEM possui os melhores resultados para as medidas de proximidade da DCT.

3.5 O Método de Jridi-Alfalou-Meher

Em [48] Jridi, Alfalou e Meher propuseram um método de escalonamento de aproximações da DCT. Com este método, a partir de uma transformada de comprimento $\frac{N}{2}$ de baixa complexidade, pode-se obter uma transformada de comprimento N também de baixa complexidade. A matriz da

transformada de comprimento N de baixa complexidade será obtida da seguinte maneira:

$$\mathbf{T}_N = \mathbf{M}_N^{per} \begin{bmatrix} \mathbf{T}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{T}_{\frac{N}{2}} \end{bmatrix} \mathbf{M}_N^{add}, \quad (3.7)$$

em que $\mathbf{T}_{\frac{N}{2}}$ é a matriz de comprimento $\frac{N}{2}$ de baixa complexidade, $\mathbf{0}_{\frac{N}{2}}$ é uma matriz de zeros de ordem $\frac{N}{2}$. Já \mathbf{M}_N^{add} é uma matriz de adições e \mathbf{M}_N^{per} é uma matriz de permutação, ambas são de ordem N e são dadas, respectivamente, por:

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

$$\mathbf{M}_N^{per} = \begin{bmatrix} \mathbf{P}_{N-1, \frac{N}{2}} & \mathbf{0}_{1, \frac{N}{2}} \\ \mathbf{0}_{1, \frac{N}{2}} & \mathbf{P}_{N-1, \frac{N}{2}} \end{bmatrix}.$$

Ainda, $\mathbf{0}_{1, \frac{N}{2}}$ é uma matriz de zeros de ordem $1 \times \frac{N}{2}$ e $\mathbf{P}_{N-1, \frac{N}{2}}$ é uma matriz de ordem $N-1 \times \frac{N}{2}$ cujas linhas são dadas por:

$$\mathbf{P}_{N-1, \frac{N}{2}} = \begin{cases} \mathbf{0}_{1, \frac{N}{2}}, & \text{se } i \text{ ímpar} \\ \mathbf{I}_{\frac{N}{2}} \left(\frac{i}{2} \right), & \text{se } i \text{ par} \end{cases},$$

em que $\mathbf{I}_{\frac{N}{2}} \left(\frac{i}{2} \right)$ é a $\left(\frac{i}{2} \right)$ -ésima linha da matriz identidade de ordem $\frac{N}{2} \times \frac{N}{2}$.

De acordo com a Equação (3.7), é possível encontrar uma equação para a complexidade aritmética das transformadas escalonadas. Como a matriz \mathbf{M}_N^{per} é uma matriz de permutação, ela não altera a complexidade aritmética, já a matriz \mathbf{M}_N^{add} possui N adições e a matriz

$$\begin{bmatrix} \mathbf{T}_{\frac{N}{2}} & \mathbf{0}_{\frac{N}{2}} \\ \mathbf{0}_{\frac{N}{2}} & \mathbf{T}_{\frac{N}{2}} \end{bmatrix}$$

possui duas vezes o número de adições de $\mathbf{T}_{\frac{N}{2}}$. Desta forma, o número de adições necessárias para o cômputo de \mathbf{T}_N é dado por:

$$\mathcal{A}(\mathbf{T}_N) = 2\mathcal{A}\left(\mathbf{T}_{\frac{N}{2}}\right) + N,$$

em que $\mathcal{A}(\cdot)$ é uma função que retorna o número de adições necessárias para o cômputo da transformada que será dada como argumento de entrada. De forma análoga, para o cálculo do número de deslocamentos de *bit* necessários, temos:

$$\mathcal{S}(\mathbf{T}_N) = 2\mathcal{S}\left(\mathbf{T}_{\frac{N}{2}}\right),$$

em que $\mathcal{S}(\cdot)$ é uma função que retorna o número de deslocamentos de *bit* necessários para o cômputo da transformada que será dada como argumento de entrada.

3.5.1 Escalonamento da parametrização BAS de comprimento 8 para 16

A partir do escalonamento da transformada parametrizada de comprimento 8 pelo método JAM temos a seguinte transformada parametrizada de comprimento 16:

$$\mathbf{T}_{16}(\mathbf{a}) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & a_1 & a_1 & -a_1 & -a_1 & -1 & -1 & -1 & -1 & -a_1 & -a_1 & a_1 & a_1 & 1 & 1 \\ 1 & 1 & a_1 & a_1 & -a_1 & -a_1 & -1 & -1 & 1 & 1 & a_1 & a_1 & -a_1 & -a_1 & -1 & -1 \\ 1 & a_2 & -a_2 & -1 & -1 & -a_2 & a_2 & 1 & 1 & a_2 & -a_2 & -1 & -1 & -a_2 & a_2 & 1 \\ 1 & a_2 & -a_2 & -1 & -1 & -a_2 & a_2 & 1 & -1 & -a_2 & a_2 & 1 & 1 & a_2 & -a_2 & -1 \\ a_1 & a_3 & -a_4 & -a_1 & a_1 & a_4 & -a_3 & -a_1 & -a_1 & -a_3 & a_4 & a_1 & -a_1 & -a_4 & a_3 & a_1 \\ a_1 & a_3 & -a_4 & -a_1 & a_1 & a_4 & -a_3 & -a_1 & a_1 & a_3 & -a_4 & -a_1 & a_1 & a_4 & -a_3 & -a_1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\ a_5 & -a_5 & -a_1 & a_6 & -a_6 & a_1 & a_5 & -a_5 & -a_5 & a_5 & a_1 & -a_6 & a_6 & -a_1 & -a_5 & a_5 \\ a_5 & -a_5 & -a_1 & a_6 & -a_6 & a_1 & a_5 & -a_5 & a_5 & -a_5 & -a_1 & a_6 & -a_6 & a_1 & a_5 & -a_5 \\ a_2 & -1 & 1 & -a_2 & -a_2 & 1 & -1 & a_2 & a_2 & -1 & 1 & -a_2 & -a_2 & 1 & -1 & a_2 \\ a_2 & -1 & 1 & -a_2 & -a_2 & 1 & -1 & a_2 & -a_2 & 1 & -1 & a_2 & a_2 & -1 & 1 & -a_2 \\ a_7 & -a_6 & a_1 & -a_8 & a_8 & -a_1 & a_6 & -a_7 & -a_7 & a_6 & -a_1 & a_8 & -a_8 & a_1 & -a_6 & a_7 \\ a_7 & -a_6 & a_1 & -a_8 & a_8 & -a_1 & a_6 & -a_7 & a_7 & -a_6 & a_1 & -a_8 & a_8 & -a_1 & a_6 & -a_7 \end{bmatrix}. \quad (3.8)$$

De acordo com a Tabela 3.7 podemos ver a avaliação das transformadas escalonadas pelo método JAM. Todas as transformadas encontradas são novas na literatura. Essas transformadas serão comparadas com as transformadas da Tabela 3.5 e da Tabela 3.6. Existiam na literatura apenas transformadas que exigiam 44, 60, 64 e 72 adições para o seu cômputo entre as transformadas de comprimento 16. Este trabalho introduziu transformadas que exigem 48, 52, e 56 adições, além de transformadas que exigem 60 e 64 adições.

A transformada $\hat{\mathbf{C}}_1$ é a única da literatura com 48 adições e pode ser comparada com a transformada SOBCM que possui 44 adições. A transformada $\hat{\mathbf{C}}_1$ possui medidas consideravelmente melhores que a transformada SOBCM e possui apenas 4 adições a mais. Entre as transformadas com 52 adições a transformada $\hat{\mathbf{C}}_4$ possui a melhor medida em relação ao erro de energia total e a transformada $\hat{\mathbf{C}}_5$ possui as melhores medidas para erro quadrático médio, ganho de codificação e eficiência. Já para as transformadas que possuem 56 adições a transformada $\hat{\mathbf{C}}_9$ possui as melhores medidas para todas as métricas. Para as transformadas que necessitam de 60 adições a transformada JAM possui as melhores medidas para erro de energia total, erro quadrático médio e ganho de codificação e a transformada $\hat{\mathbf{C}}_{13}$ possui a melhor medida para eficiência. A transformada ABM escalonada pelo método JAM possui as melhores medidas para todas as métricas entre as transformadas que necessitam 64 adições.

3.5.2 Escalonamento das parametrizações BAS de comprimento 16 para 32

Escalonando a transformada parametrizada dada na Equação 3.9, temos a seguinte transformada escalonada para comprimento 32:

Tabela 3.8: *Medidas de avaliação das transformadas ótimas.*

a^*	$\epsilon(a^*)$	$MSE(a^*)$	$C_g^*(a^*)$	$\eta(a^*)$	$\mathcal{A}(a^*)$	$\mathcal{S}(a^*)$
0	133,00	0,15	8,24	55,97	144	0
1	78,06	0,13	8,27	55,91	160	0
2	124,71	0,18	8,59	58,37	160	16

(1)(2)(3)(4)(5)(6)(7 11 15 13 19 21 23 24 25 8 12 16 14 20 22)(9)(10)(17 18)(26)(27)(28)(29)(30)
 (31)(32) e (1)(2)(3)(4)(5)(6 11 19 21 22 23 7)(8 12 20 24 27 31 29 26 28 32 30 25)(9)(10)(13 15)
 (14 16)(17)(18), para $a = 0$, $a = 1$ e $a = 2$, respectivamente.

A partir do escalonamento da transformada da Equação 3.8 pelo método JAM temos a seguinte transformada parametrizada de comprimento 32:

Na Tabela 3.9, podemos observar as medidas de avaliação das transformadas de comprimento 32 encontradas a partir da classe parametrizada dada na Equação 3.10.

Tabela 3.9: Medidas de avaliação das transformadas ótimas da parametrização BAS de comprimento 16 escalonadas para comprimento 32 a partir do método JAM.

\hat{C}_j	$\epsilon(\mathbf{a}^*)$	MSE(\mathbf{a}^*)	$C_g^*(\mathbf{a}^*)$	$\eta(\mathbf{a}^*)$	$\mathcal{A}(\mathbf{a}^*)$	$\mathcal{S}(\mathbf{a}^*)$
\hat{C}_1	68,13	0,13	8,23	56,18	128	0
\hat{C}_2	65,78	0,13	8,23	56,05	136	0
\hat{C}_3	60,57	0,13	8,23	56,43	136	4
\hat{C}_4	59,47	0,13	8,23	56,78	136	4
\hat{C}_5	63,93	0,12	8,44	56,72	136	8
\hat{C}_6	60,69	0,12	8,25	56,47	144	0
\hat{C}_7	57,12	0,12	8,23	56,65	144	4
\hat{C}_8	58,22	0,12	8,23	56,31	144	4
\hat{C}_9	55,27	0,12	8,44	57,33	144	12
\hat{C}_{10}	56,37	0,12	8,44	56,98	144	12
\hat{C}_{11}	52,23	0,12	8,27	56,03	152	0
\hat{C}_{12}	56,49	0,12	8,46	57,01	152	8
\hat{C}_{13}	52,93	0,12	8,44	57,57	152	8
\hat{C}_{14}	48,04	0,12	8,48	56,57	160	8
\hat{C}_{15}	48,73	0,12	8,65	58,14	160	16

Na Tabela 3.10, temos todas transformadas de comprimento 32 que, a melhor de nosso conhecimento, foram propostas na literatura e serão comparadas com as novas transformadas encontradas.

Tabela 3.10: Medidas de avaliação para as transformadas da série de aproximações BAS e demais transformadas consideradas para a comparação de comprimento 32.

Transformada	ϵ	MSE	C_g^*	η	\mathcal{A}	\mathcal{S}
JAM	48,10	0,11	8,50	56,97	152	0
BAS-2013	192,18	0,76	8,27	55,91	160	0
BAS-2010	117,07	0,24	8,50	58,50	160	16
ABM (JAM)	46,27	0,11	8,95	61,03	160	24

De acordo com a Tabela 3.9, a Tabela 3.8 e a Tabela 3.10 pode-se ver os resultados das medidas de avaliação das transformadas propostas e das transformadas já arquivadas na literatura que serão utilizadas para comparação dos resultados. Temos uma nova transformada que necessita de 128 adições para o seu cômputo, sendo a transformada com menor complexidade aritmética na literatura

para transformadas de comprimento 32. Todas transformadas que exigem 136 adições são novas transformadas e também não possuem concorrentes. A transformada \hat{C}_4 possui os melhores resultados para erro de energia total e eficiência e a transformada \hat{C}_5 os melhores resultados para erro quadrático médio e ganho de codificação. Todas transformadas que necessitam 144 adições também são novas e não possuem concorrentes. A transformada \hat{C}_6 possui o melhor resultado em relação ao erro quadrático médio e a transformada \hat{C}_9 os melhores resultados para erro de energia total, ganho de codificação e eficiência. Entre as transformadas que necessitam 152 adições a transformada JAM possui os melhores resultados para erro de energia total, erro quadrático médio e ganho de codificação e a transformada \hat{C}_{13} o melhor resultado para eficiência. A transformada ABM escalonada pelo método JAM possui os melhores resultados em todas as métricas entre as transformadas que necessitam 160 adições para o seu cômputo.

3.6 Análise de *trade-off*

É muito difícil analisar a complexidade computacional de uma transformada conjuntamente com a seu desempenho. Desta forma, em [75] é proposta a análise de *trade-off*. Esta métrica avalia conjuntamente o desempenho da transformada e seu custo aritmético através de uma combinação convexa. Tal medida é sugerida da seguinte forma [75]:

$$f = \gamma \cdot (\text{custo computacional normalizado}) + (1 - \gamma) \cdot (\text{desempenho normalizado}),$$

em que $\gamma \in [0, 1]$. O desempenho normalizado é obtido através da normalização de qualquer métrica de desempenho, como as descritas na Subseção 2.7.1. Para normalizar basta dividir a métrica de desempenho escolhida de cada transformada que será considerada na análise pela métrica de desempenho máxima das transformadas consideradas. A métrica de desempenho utilizada deve ser do tipo que sua otimização é dada pela minimização do valor, caso contrário deve ser utilizado o valor negativo da métrica considerada. A medida f deve ser minimizada.

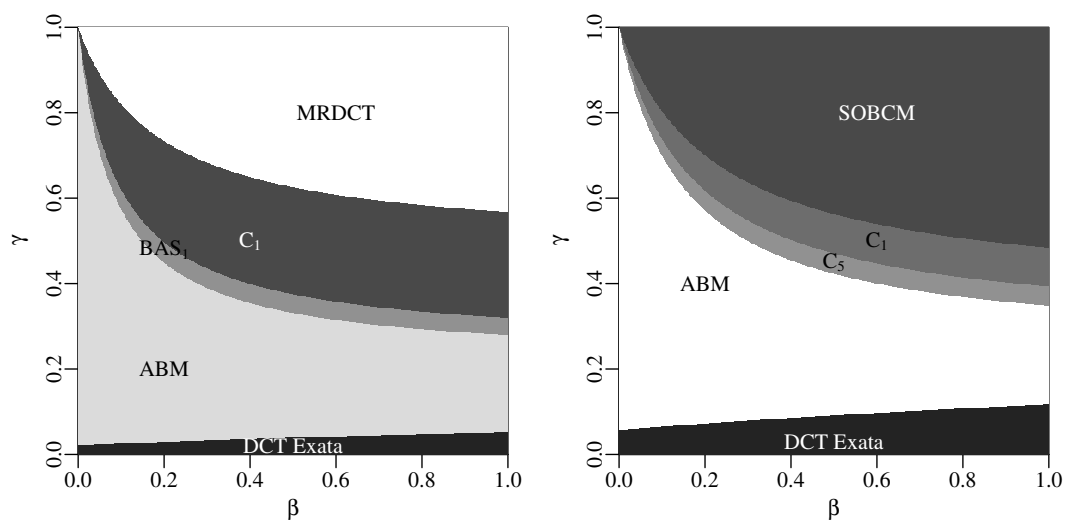
O custo computacional é definido pelo custo aritmético, que é dado da seguinte maneira:

$$(\text{custo aritmetico}) = M + \beta A,$$

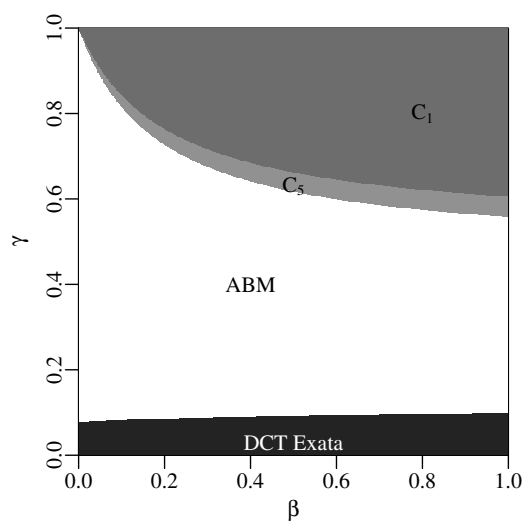
em que M é o número de multiplicações exigidas, A o número de adições e $\beta \in [0, 1]$. Para normalização basta dividir o custo aritmético de cada transformada pelo custo aritmético da transformada que tem maior custo entre as transformadas consideradas na análise. Para interpretação dessa análise temos o β que define o peso ao número de adições utilizadas e γ que pondera o custo total em relação ao desempenho da transformada.

Na Figura 3.4 temos o resultado da análise de *trade-off* das transformadas propostas e das transformadas utilizadas como comparação para as comprimentos 8, 16 e 32. Como o interesse desse trabalho é melhor qualidade na compressão de imagens e não proximidade com a DCT, na análise de *trade-off* foi utilizado o ganho de codificação como medida de desempenho, assim como em [75].

De acordo com a Figura 3.4(a), podemos visualizar as transformadas de comprimento 8 apontadas pela análise de *trade-off*, que são as transformadas MRDCT, \hat{C}_1 , BAS_1 , ABM e a DCT exata. Já para comprimento 16 temos as transformadas SOBCM, \hat{C}_1 , \hat{C}_5 e ABM. Finalmente, para comprimento 32 foram selecionadas as transformadas, \hat{C}_1 , \hat{C}_5 e ABM. Para a escolha das transformadas devem ser escolhidos valores de β e γ que sejam adequados de acordo com a aplicação que deseja-se utilizar tais transformadas.



(a) Análise de *trade-off* para as transformadas de comprimento 8. (b) Análise de *trade-off* para as transformadas de comprimento 16.



(c) Análise de *trade-off* para as transformadas de comprimento 32.

Figura 3.4: Análise de *trade-off* para transformadas de diferentes comprimentos.

EXPERIMENTO COMPUTACIONAL

Foram feitos experimentos de compressão de imagem com as transformadas ótimas obtidas no Capítulo 3, assim como com as suas versões escalonadas. As imagens comprimidas foram submetidas a métricas de avaliação as quais medem a qualidade da compressão de imagens, tais como a relação sinal-ruído de pico (PSNR) e o índice de similaridade estrutural médio (MSSIM).

4.1 Metodologia

Com o objetivo de avaliar o desempenho das aproximações selecionadas, adotamos a compressão de imagens tipo JPEG baseada em 45 imagens de 8 *bits* de tamanho 512×512 *pixels* em escala de cinza, obtidas do banco de imagens público “*The USC-SIPI Image Database*” [76]. O método de compressão de imagens utilizado segue o seguinte passo-a-passo:

1. Cada imagem foi dividida em blocos de comprimento 8, 16 e 32, de acordo com o comprimento da transformada considerada;
2. Posteriormente, a cada bloco foi submetido às transformadas propostas;
3. Empregando-se a sequência zig-zag padrão [65], r coeficientes iniciais de cada bloco foram retidos e usados para reconstruir a imagem [25];
4. Os demais coeficientes são atribuídos com valor zero;
5. Aplica-se a transformada inversa em cada bloco;
6. Recompõe-se os blocos comprimidos em seus respectivos lugares.

Transformadas com comprimento 8 fornecem 64 coeficientes no domínio da transformada para cada bloco enquanto que, transformadas de comprimento 16 fornecem 256 coeficientes e transformadas de comprimento 32 fornecem 1024 coeficientes para cada bloco. Foi adotado $1 \leq r \leq 48$, $1 \leq r \leq 192$ e $1 \leq r \leq 768$ para as transformadas de comprimento 8, 16 e 32, respectivamente. Tais valores de r indicam compressão de 25% à, aproximadamente, 99%. O percentual de compressão da imagem (PCI) é dado por:

$$\text{PCI} = 1 - \frac{r}{N^2}$$

Após a compressão a qualidade da imagem compactada foi avaliada a partir da PSNR e do MSSIM. A metodologia apresentada anteriormente é também descrita em [23, 27, 34–37]. Diferentemente dos experimentos de compressão do tipo JPEG realizados em [23, 27, 34–37], utilizaremos a média das medidas de avaliação de qualidade de imagens considerando o conjunto das 45 imagens como em [24, 25, 31]. Utilizando esse tipo de experimento os resultados serão menos afetados pela variabilidade dos dados de entrada, proporcionando, uma metodologia mais robusta [77].

A PSNR e o MSSIM medem a degradação da imagem compactada. O PSNR é uma medida amplamente utilizada em processamento de imagens [78] e o MSSIM é considerado como um método complementar para avaliação da qualidade de imagens [79]. O MSSIM considera luminância, contraste e estrutura da imagem para quantificar a degradação da imagem, resultando numa medida de qualidade subjetiva [80]. O PSNR é dado por:

$$\text{PSNR} = 10 \log_{10} \left(\frac{\text{MAX}}{\text{MSE}} \right),$$

sendo que

$$\text{MSE} = \frac{\sum_{i=1}^N \sum_{j=1}^N (\mathbf{A}_{i,j} - \mathbf{B}_{i,j})^2}{N^2},$$

em que \mathbf{A} é a imagem original, \mathbf{B} é a imagem reconstituída através de alguma transformada, e MAX é o valor máximo possível dos elementos de \mathbf{A} e \mathbf{B} . Neste contexto, como trabalharemos com imagens em escala de cinza com 8 *bits*, temos $\text{MAX} = 2^8 - 1$. Já o MSSIM é dado por:

$$\text{MSSIM} = \frac{1}{M} \sum_{j=1}^M \text{SSIM}(\mathbf{a}_j, \mathbf{b}_j),$$

em que \mathbf{a}_j e \mathbf{b}_j são os j -ésimos blocos de \mathbf{A} e \mathbf{B} identicamente ordenados, respectivamente, e M é o total de blocos. Em imagens de tamanho 512×512 temos 4096, 1024 e 256 blocos por imagem utilizando transformadas de comprimento 8, 16 e 32, respectivamente. O cálculo do SSIM, ou seja,

o cálculo para cada bloco, é dado por:

$$\text{SSIM}(\mathbf{a}_j, \mathbf{b}_j) = \frac{(2\mu_a\mu_b + C_1)(2\sigma_{ab} + C_2)}{(\mu_a^2 + \mu_b^2 + C_1)(\sigma_a^2 + \sigma_b^2 + C_2)},$$

em que μ_a é a média dos elementos do bloco \mathbf{a}_j , μ_b é a média dos elementos do bloco \mathbf{b}_j , σ_a é o desvio padrão dos elementos do bloco \mathbf{a}_j , σ_b é o desvio padrão dos elementos do bloco \mathbf{b}_j , σ_{ab} é a covariância entre os blocos \mathbf{a}_j e \mathbf{b}_j , $C_1 = 6,5025$ e $C_2 = 58,5225$ [80].

A fim de melhor visualização dos resultados do experimento computacional, também foi considerado o erro absoluto percentual (APE) relativo à DCT das medidas de PSNR e MSSIM. O APE é calculado segundo a seguinte expressão:

$$\text{APE} = \left| \frac{C - T}{T} \right|,$$

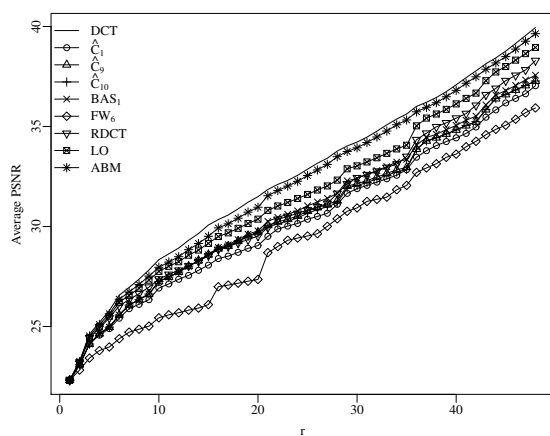
em que C e T são os valores de PSNR ou MSSIM considerando a DCT exata e uma dada aproximação T , respectivamente.

4.2 Resultados e discussão

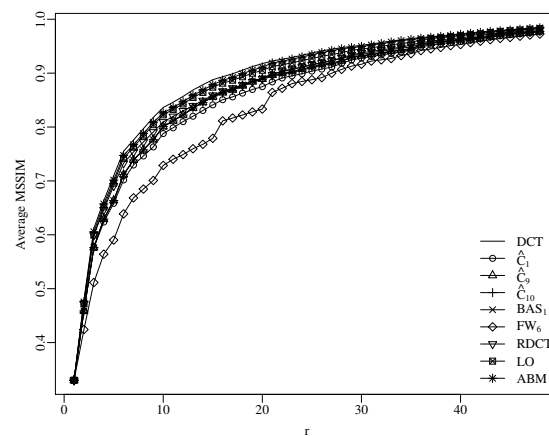
Aqui foram apresentados os resultados da avaliação das transformadas em relação ao PSNR e ao MSSIM. Serão consideradas separadamente as transformadas de comprimento 8, 16 e 32. Ainda será considerada um comparação entre as transformadas de comprimento 8 que necessitam 20 adições para o seu cômputo. Nesta seção, serão apresentados os resultados de forma quantitativa, ou seja, foram realizadas simulações e os resultados serão apresentados em forma de gráficos representando o PSNR e MSSIM para diferentes valores de r . No Apêndice B, os resultados dessa Seção foram parcialmente apresentados de forma qualitativa, e podem ser visualizadas as imagens comprimidas.

4.2.1 Comprimento 8

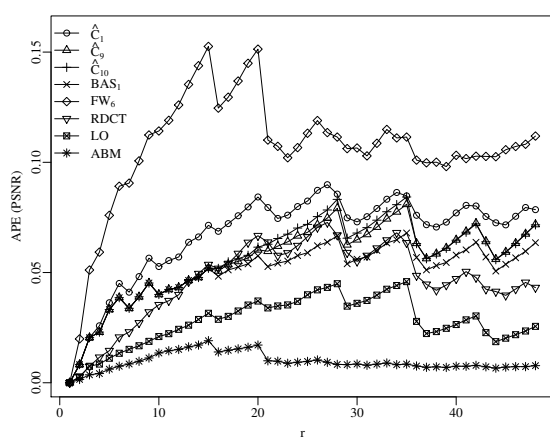
Na Figura 4.1, podemos visualizar as medidas de avaliação em relação a compressão de imagens das transformadas de comprimento 8 consideradas. Como esperado, o desempenho das transformadas em relação às medidas de compressão de imagem tiveram o mesmo comportamento que a medida de ganho de codificação. Desta forma, como podemos ver nas Figuras 4.1(a), 4.1(b), 4.1(c) e 4.1(d), a transformada ABM teve os melhores resultados e a transformada FW_6 os piores. Entre as transformadas comparadas que necessitam 18 adições para o seu cômputo, a transformada BAS_1 possui melhor comportamento em relação à PSNR e ao MSSIM. Já para as transformadas que necessitam



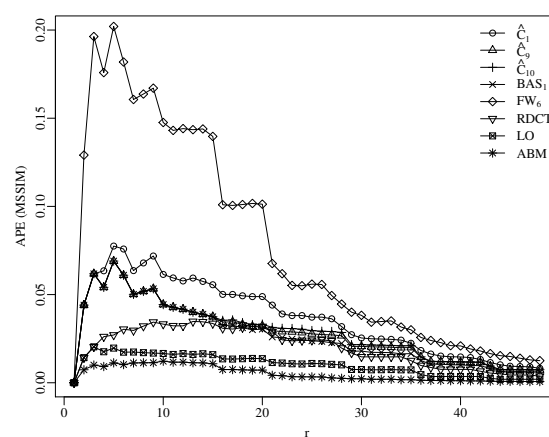
(a) PSNR.



(b) MSSIM.



(c) APE (PSNR).



(d) APE (MSSIM).

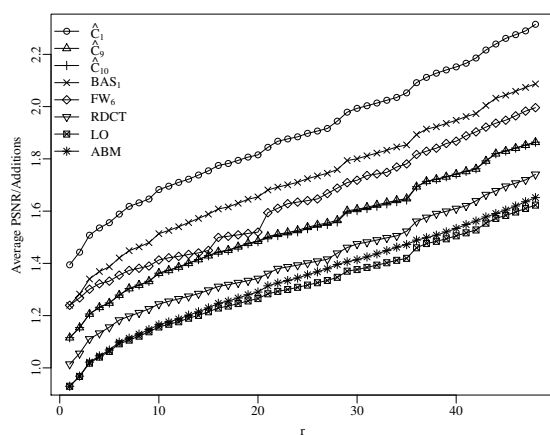
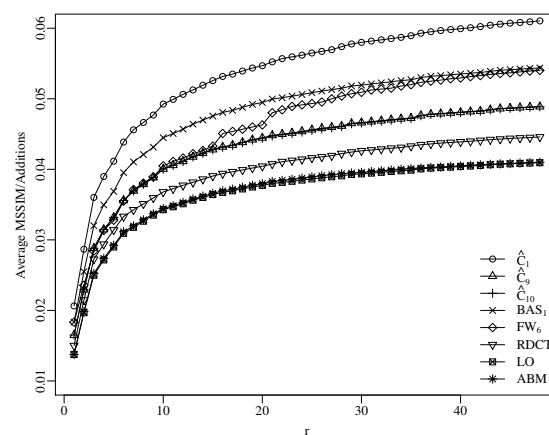
(e) PSNR/ \mathcal{A} .(f) MSSIM/ \mathcal{A} .

Figura 4.1: Medidas de avaliação das transformadas ótimas propostas com comprimento 8 com relação a compressão de imagem variando o valor de r .

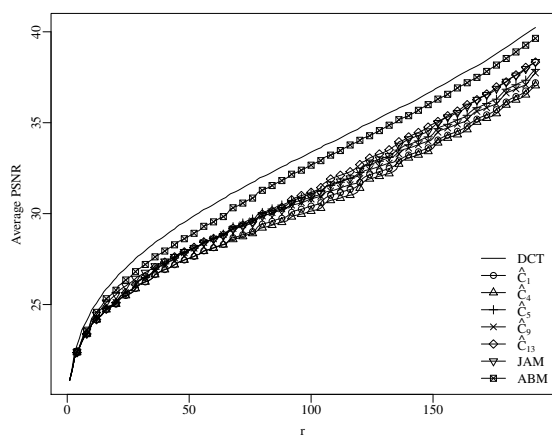
de 24 adições, a transformada ABM possui melhor comportamento. Em geral, as transformadas demonstram ter melhor comportamento em relação à PSNR e ao MSSIM conforme o aumento de sua complexidade aritmética, com exceção da transformada FW_6 . Porém, isso já era esperado, pois a aproximação FW_6 é a transformada que possui menor ganho de codificação.

Nas Figuras 4.1(e) e 4.1(f), podemos ver o ganho de PSNR e MSSIM por adição. A transformada \hat{C}_1 possui o maior ganho por adição e a transformada ABM, apesar de ter as melhores medidas de maneira geral, possui o menor ganho por adição. A transformada \hat{C}_9 tem um comportamento mediano em relação ao ganho de codificação, PSNR, MSSIM e ganho de PSNR e MSSIM por adição, sendo uma boa alternativa sugerida neste trabalho. Também, para valores $10 \leq r \leq 20$, a transformada \hat{C}_9 possui valores de PSNR próximos ou até melhores que a transformada RDCT que possui 22 adições. Em relação ao MSSIM, para valores de $r \geq 16$, os resultados, em geral, são próximos.

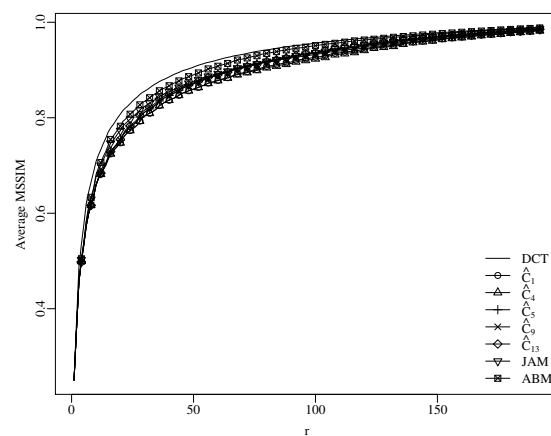
4.2.2 Comprimento 16

Visualizando a Figura 4.2 podem ser vistas as medidas de avaliação das transformadas de comprimento 16 consideradas em relação à compressão de imagens. De acordo com as Figuras 4.2(a), 4.2(b), 4.2(c) e 4.2(d), a transformada ABM teve os melhores resultados e a transformada \hat{C}_4 os piores. Entre as transformadas que necessitam 52 adições para o seu cômputo, de maneira geral, a transformada \hat{C}_5 possui melhor comportamento em relação à PSNR e ao MSSIM. Considerando as transformadas que necessitam 60 adições para o seu cômputo, para $r < 50$ a transformada JAM possui melhor comportamento. Já para $95 < r < 150$ a transformada \hat{C}_{13} possui melhor comportamento, para outros valores de r elas possuem valores de PSNR e MSSIM muito próximos. Em geral, as transformadas tem melhores resultados de PSNR e MSSIM conforme o aumento da complexidade aritmética.

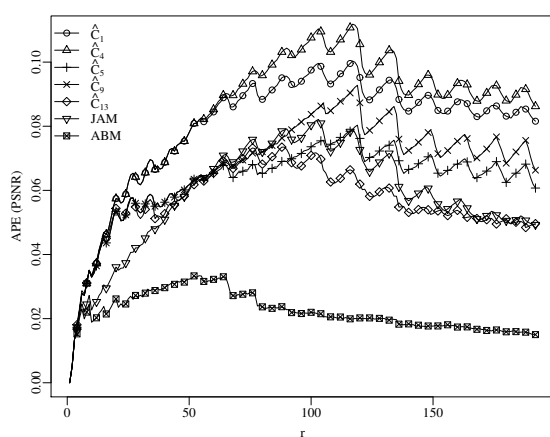
As Figuras 4.2(e) e 4.2(f), mostram que a transformada \hat{C}_1 possui o maior ganho por adição e a transformada ABM, apesar de ter as melhores medidas, de maneira geral, possui o menor ganho por adição. A transformada \hat{C}_9 tem um comportamento mediano em relação ao ganho de PSNR e MSSIM por adição e um desempenho mediano em relação à compressão de imagem em relação a todas as transformadas. Ainda, se destaca a transformada \hat{C}_5 que possui um alto ganho de PSNR e MSSIM por adição e um comportamento geral médio bom em relação ao PSNR e MSSIM, necessitando apenas 2 adições a mais que a transformada \hat{C}_1 .



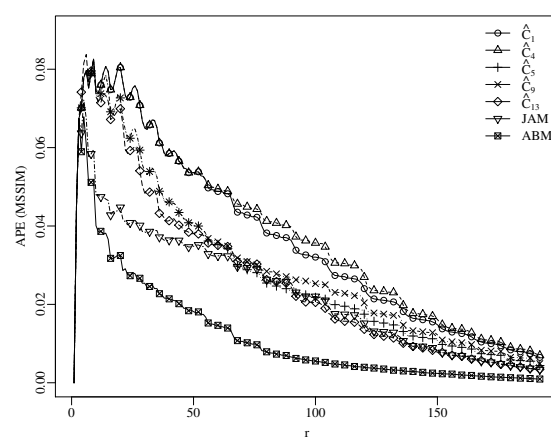
(a) PSNR.



(b) MSSIM.



(c) APE (PSNR).



(d) APE (MSSIM).

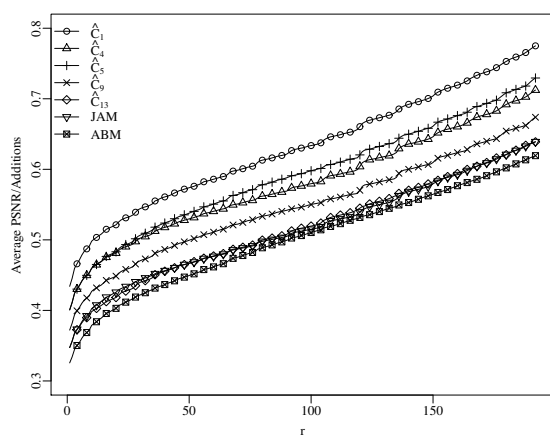
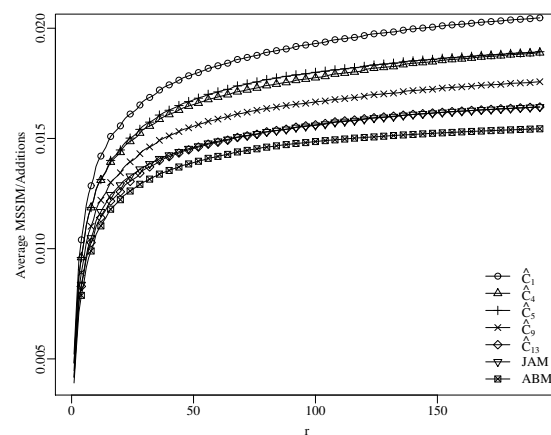
(e) PSNR/ \mathcal{A} .(f) MSSIM/ \mathcal{A} .

Figura 4.2: Medidas de avaliação das transformadas ótimas propostas com comprimento 16 com relação a compressão de imagem variando o valor de r .

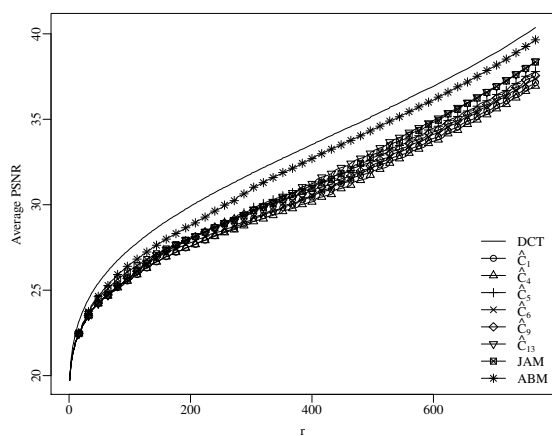
4.2.3 Comprimento 32

De acordo com a Figura 4.3, podemos visualizar as medidas de avaliação das transformadas de comprimento 32 em relação a compressão de imagens. Podemos ver que as transformadas escalonadas de comprimento 16 para comprimento 32 se comportam de maneira muito semelhante às transformadas escalonadas de comprimento 8 para comprimento 16. As transformadas \hat{C}_1 , \hat{C}_4 , \hat{C}_5 , \hat{C}_6 e \hat{C}_9 possuem complexidade aritmética mais baixa que as transformadas de comprimento 32 existentes na literatura e a transformada \hat{C}_1 possui menor complexidade aritmética. As transformadas \hat{C}_4 e \hat{C}_5 necessitam 136 adições para o seu cômputo, porém a transformada \hat{C}_5 possui, em geral, melhores resultados em relação à PSNR e ao MSSIM. Para as transformadas que necessitam 144 adições temos as transformadas \hat{C}_6 e \hat{C}_9 . Em geral, a transformada \hat{C}_9 tem melhor comportamento em relação à PSNR e ao MSSIM. Podemos ver que a transformada \hat{C}_{13} ainda pode ser concorrente para a transformada JAM para alguns valores de r , sendo que e as duas transformadas necessitam 152 adições para o seu cômputo.

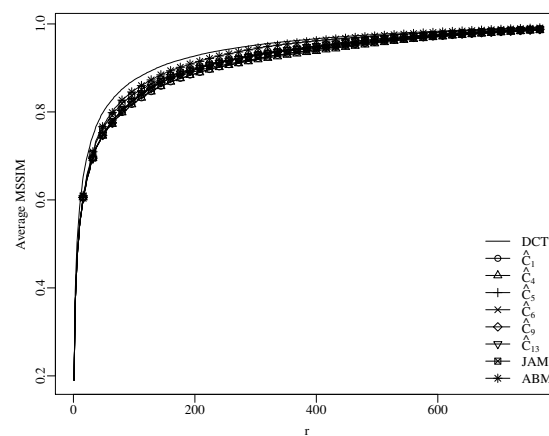
Podemos ver o ganho de PSNR e MSSIM por adição nas Figuras 4.3(e) e 4.3(f). A transformada \hat{C}_1 possui o maior ganho por adição e a transformada ABM possui o menor ganho por adição. As transformadas \hat{C}_6 e \hat{C}_9 tem um comportamento mediano em relação ao ganho de PSNR e MSSIM por adição e um desempenho mediano em relação a compressão de imagem em relação a todas as transformadas, porém a transformada \hat{C}_9 , tem um comportamento melhor.

4.2.4 Comparação entre as transformadas com comprimento 8 e que necessitam 20 adições para seu cômputo

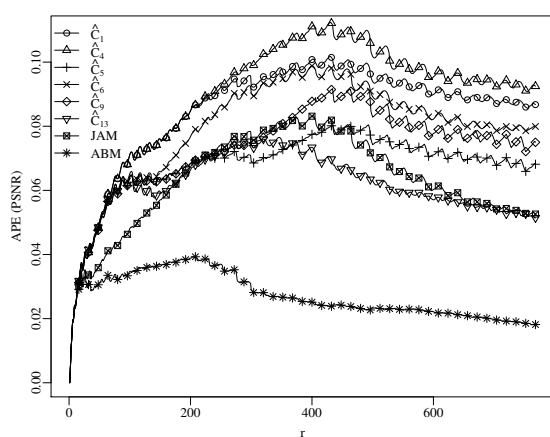
Aqui destacamos duas das transformadas de 20 adições propostas comparadas com a transformada referência de 20 adições da literatura. Na Figura 4.4 podem ser vistas as medidas de avaliação em relação a compressão de imagens das transformadas de comprimento 8 que necessitam 20 adições. De acordo com a Figura 4.4 podemos ver que as duas transformadas propostas possuem resultados melhores que a transformada que é referência na literatura. Ainda, podemos ver que a transformada \hat{C}_9 possui valores de MSSIM e PSNR melhores ou iguais à transformada \hat{C}_{10} .



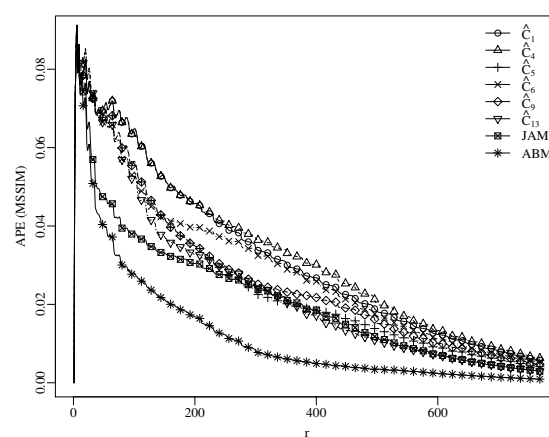
(a) PSNR.



(b) MSSIM.



(c) APE (PSNR).



(d) APE (MSSIM).

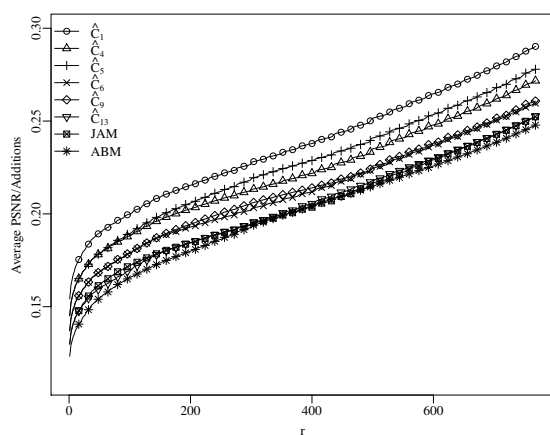
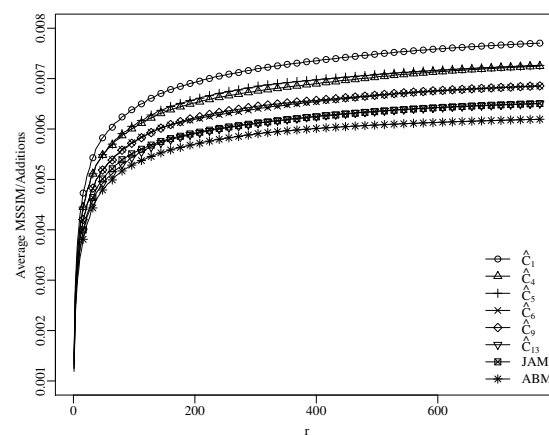
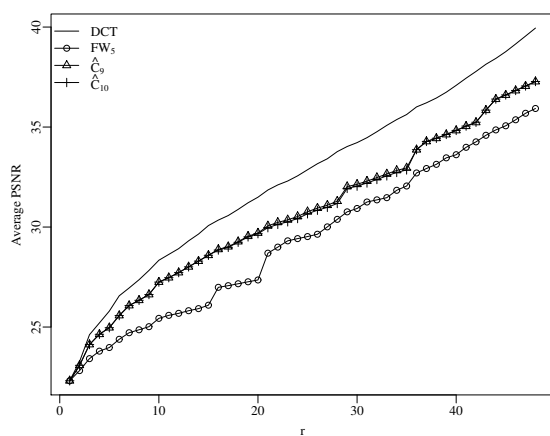
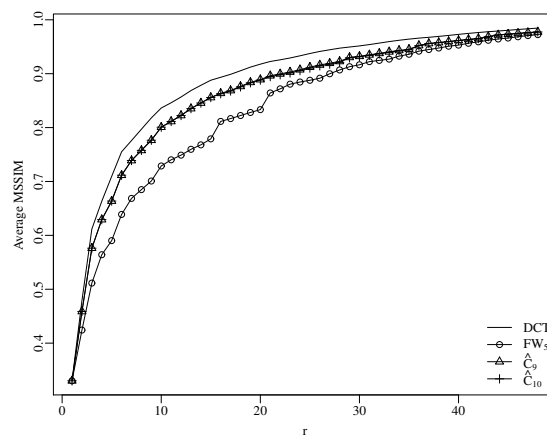
(e) PSNR/ \mathcal{A} .(f) MSSIM/ \mathcal{A} .

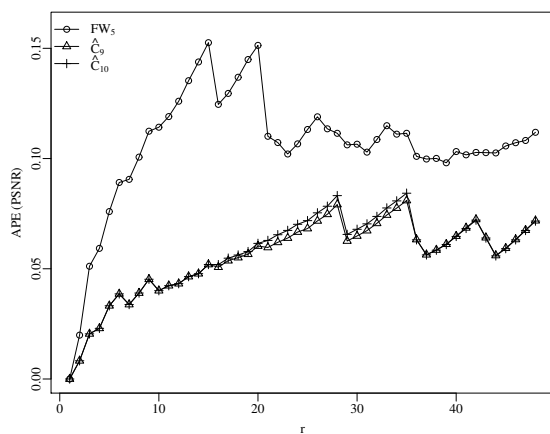
Figura 4.3: Medidas de avaliação das transformadas ótimas propostas com comprimento 32 com relação a compressão de imagem variando o valor de r .



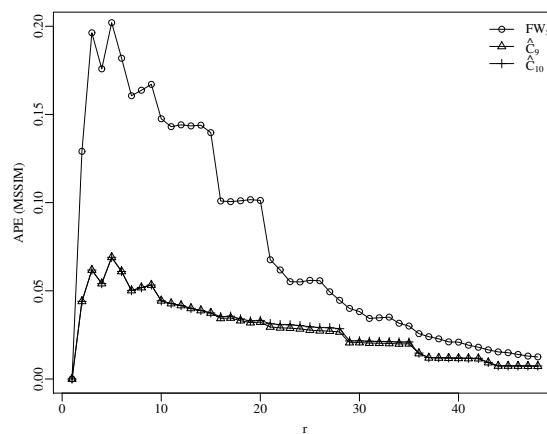
(a) PSNR.



(b) MSSIM.



(c) APE (PSNR).



(d) APE (MSSIM).

Figura 4.4: Medidas de avaliação das transformadas ótimas propostas com comprimento de 20 adições com relação a compressão de imagem variando o valor de r .

CONCLUSÕES

Neste trabalho, foram obtidas diferentes aproximações de baixa complexidade para a DCT de comprimentos 8, 16 e 32 para serem aplicadas principalmente no contexto de compressão de imagens. A obtenção destas transformadas baseou-se na parametrização da classe de aproximações BAS de comprimento 8 e 16 e seus escalonamentos pelo método JAM. Transformadas de baixa complexidade podem ser implementadas em *hardware* e são atrativas por requerer baixo custo e baixa demanda de potência, assim como transmissão e processamento de imagem e vídeo em tempo real. Também podem ser utilizadas em equipamentos portáteis que requerem baixo custo e baixo consumo de energia, entre outros.

Nos Capítulos 1 e 2, foram apresentadas uma introdução e as considerações teóricas da KLT e a DCT como aproximação assintótica da KLT, assim como apresentação de diversas aproximações para a DCT que estão presentes na literatura. Esses capítulos motivam o uso de aproximações de baixa complexidade no contexto de compressão de imagens, assim como provêm uma revisão geral da literatura nesse contexto.

As contribuições desta dissertação estão no Capítulo 3, onde foram propostas duas novas classes de transformadas parametrizadas baseada na série BAS de comprimento 8 e 16. Essa nova classe parametrizada de transformadas contém um grande número de transformadas potencialmente interessantes. Foi proposto um problema de otimização multicritério a fim de selecionar as transformadas que tenham um desempenho ótimo em relação a medidas de avaliação, tais como: erro de energia, erro quadrático médio, ganho em codificação e eficiência da transformada. As transformadas obtidas a partir do problema de otimização foram escalonadas pelo método JAM e obtivemos suas versões escalonadas com comprimento 16 e 32. Todas transformadas obtidas foram comparadas com ou-

tras aproximações já listadas na literatura e que não pertencem a classe multiparamétrica proposta. Foi proposto um algoritmo rápido geral e todas as transformadas propostas possuem praticamente a mesma estrutura de algoritmo rápido e podem ser implementadas com modificações mínimas de *hardware*. Ainda, foi feita uma análise de *trade-off* para avaliar conjuntamente o custo das transformadas e seus desempenhos.

No Capítulo 4, todas as transformadas obtidas foram avaliadas e comparadas, no contexto de compressão de imagem, com outras transformadas em relação ao PSNR e MSSIM. Essas medidas proporcionam valores que permitem comparar a qualidade de compressão das transformadas propostas. Tais medidas evidenciam que as transformadas obtidas são adequadas para aplicações no contexto de compressão de imagens. Ainda, de forma mais específica, para comprimento 8 foi encontrada uma nova transformada que necessita de 20 adições para seu cômputo, mostrando melhor desempenho que as transformadas conhecidas na literatura que exigem 20 adições. Já para transformadas de comprimento 16 e 32 foram encontradas várias transformadas novas de baixa complexidade que podem se tornar referências para transformadas com suas complexidades aritméticas.

REFERÊNCIAS

- [1] N. U. AHMED & K. R. RAO, **Orthogonal Transforms for Digital Signal Processing**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1975.
- [2] A. V. OPPENHEIM & R. W. SCHAFER, **Discrete-Time Signal Processing**, 3^a ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2009.
- [3] K. R. RAO & P. YIP, **Discrete cosine transform: algorithms, advantages, applications**. Academic press, 2014.
- [4] P. REFREGIER & B. JAVIDI, Optical image encryption based on input plane and Fourier plane-random encoding, *Optics Letters*, v. 20, n. 7, p. 767–769, Apr 1995.
- [5] S. L. MARPLE, **Digital spectral analysis: with applications**. Prentice-Hall Englewood Cliffs, NJ, 1987, v. 5.
- [6] R. HABERMAN, **Elementary applied partial differential equations**. Prentice Hall Englewood Cliffs, NJ, 1983, v. 987.
- [7] V. BRITANAK, P. YIP, & K. R. RAO, **Discrete Cosine and Sine Transforms**. Academic Press, 2007.
- [8] R. C. GONZALEZ & R. E. WOODS, **Digital Image Processing (3rd Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.
- [9] W. K. PRATT, **Digital Image Processing: PIKS Inside**, 3^a ed. New York, NY, USA: John Wiley & Sons, Inc., 2001.
- [10] Z. WANG & B. HUNT, The discrete cosine transform—a new version, In: **ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing**, v. 8, Apr 1983, p. 1256–1259.

- [11] A. K. JAIN, A sinusoidal family of unitary transforms, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. PAMI-1, n. 4, p. 356–365, 1979.
- [12] JPEG, Joint photographic experts group, 2012.
- [13] T. WIEGAND, G. J. SULLIVAN, G. BJONTEGAARD, & A. LUTHRA, Overview of the H.264/AVC video coding standard, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 13, n. 7, p. 560–576, jul 2003.
- [14] M. T. POURAZAD, C. DOUTRE, M. AZIMI, & P. NASIOPOULOS, HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC? *IEEE Consumer Electronics Magazine*, v. 1, n. 3, p. 36–46, jul. 2012.
- [15] R. CLARK, Relation between Karhunen-Loève and cosine transform, In: **IEE Proceedings-F, Communications, Radar and Signal Processing**, v. 128, n. 6, 1981, p. 359–360.
- [16] U. S. POTLURI, A. MADANAYAKE, R. J. CINTRA, F. M. BAYER, S. KULASEKERA, & A. EDIRISURIYA, Improved 8-point approximate DCT for image and video compression requiring only 14 additions, *IEEE Transactions on Circuits and Systems I*, v. 61, n. 6, p. 1727–1740, June 2014.
- [17] V. A. COUTINHO, R. J. CINTRA, F. M. BAYER, S. KULASEKERA, & A. MADANAYAKE, A multiplierless pruned DCT-like transformation for image and video compression that requires ten additions only, *Journal of Real-Time Image Processing*, v. 12, n. 2, p. 247–255, aug 2016.
- [18] W. H. CHEN, C. SMITH, & S. FRALICK, A fast computational algorithm for the discrete cosine transform, *IEEE Transactions on Communications*, v. 25, n. 9, p. 1004–1009, 1977.
- [19] B. G. LEE, A new algorithm for computing the discrete cosine transform, *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. ASSP-32, p. 1243–1245, dez. 1984.
- [20] C. LOEFFLER, A. LIGTENBERG, & G. MOSCHYTZ, Practical fast 1D DCT algorithms with 11 multiplications, In: **Proceedings of the International Conference on Acoustics, Speech, and Signal Processing**, 1989, p. 988–991.
- [21] F. FEIG & S. WINOGRAD, Fast algorithms for the discrete cosine transform, *IEEE Transactions on Signal Processing*, v. 40, n. 9, 1992.
- [22] J. LIANG & T. D. TRAN, Fast multiplierless approximations of the DCT with the lifting scheme, *IEEE Transactions on Signal Processing*, v. 49, n. 12, p. 3032–3044, Dec 2001.

- [23] T. I. HAWEEL, A new square wave transform based on the DCT, *Signal Processing*, v. 82, p. 2309–2319, 2001.
- [24] F. BAYER & R. CINTRA, DCT-like transform for image compression requires 14 additions only, *Electronics Letters*, v. 48, n. 15, p. 919–921, 2012.
- [25] R. J. CINTRA & F. M. BAYER, A DCT approximation for image compression, *IEEE Signal Processing Letters*, v. 18, n. 10, p. 579–582, 2011.
- [26] C. J. TABLADA, F. M. BAYER, & R. J. CINTRA, A class of DCT approximations based on the Feig-Winograd algorithm, *Signal Processing*, v. 11, p. 1–20, 2015.
- [27] S. BOUGUEZEL, M. O. AHMAD, & 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.
- [28] ———, Binary discrete cosine and Hartley transforms, *IEEE Transactions on Circuits and Systems I*, v. 60, n. 4, p. 989–1002, 2013.
- [29] U. S. POTLURI, A. MADANAYAKE, R. J. CINTRA, F. M. BAYER, & N. RAJAPAKSHA, Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing, *Measurement Science and Technology*, v. 23, n. 11, p. 114003, 2012.
- [30] F. M. BAYER & R. J. CINTRA, Image compression via a fast DCT approximation, *IEEE Latin America Transactions*, v. 8, n. 6, p. 708–713, dez. 2010.
- [31] F. M. BAYER, R. J. CINTRA, A. EDIRISURIYA, & A. MADANAYAKE, A digital hardware fast algorithm and FPGA-based prototype for a novel 16-point approximate DCT for image compression applications, *Measurement Science and Technology*, v. 23, n. 8, p. 114010, 2012.
- [32] F. BAYER, R. CINTRA, A. MADANAYAKE, & U. POTLURI, Multiplierless approximate 4-point DCT VLSI architectures for transform block coding, *Electronics Letters*, v. 49, n. 24, p. 1532–1534, November 2013.
- [33] R. J. CINTRA, F. M. BAYER, & C. J. TABLADA, Low-complexity 8-point DCT approximations based on integer functions, *Signal Processing*, v. 99, p. 201–214, 2014.
- [34] S. BOUGUEZEL, M. O. AHMAD, & M. N. S. SWAMY, Low-complexity 8×8 transform for image compression, *Electronics Letters*, v. 44, n. 21, p. 1249–1250, set. 2008.

- [35] —, A multiplication-free transform for image compression, In: **2nd International Conference on Signals, Circuits and Systems**, nov. 2008, p. 1–4.
- [36] —, A fast 8×8 transform for image compression, In: **2009 International Conference on Microelectronics (ICM)**, dez. 2009, p. 74–77.
- [37] —, A novel transform for image compression, In: **53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)**, Aug. 2010, p. 509–512.
- [38] N. S. JAYANT & P. NOLL, Digital coding of waveforms: principles and applications to speech and video, *Englewood Cliffs, NJ*, p. 115–251, 1984.
- [39] R. TALLIA, P. MORELLO, & CASTELLANO, The Walsh-Hadamard transform: an alternative means of obtaining phase and amplitude maps, *Journal of Nuclear Medicine*, v. 25, n. 5, p. 608–612, 1984.
- [40] R. S. OLIVEIRA & R. J. CINTRA, Uma fatoração para SDCT inversa, Trabalho de IC da UFPE, 2013.
- [41] K. LENGWEHASATIT & A. ORTEGA, Scalable variable complexity approximate forward DCT, *IEEE Transactions on Circuits and Systems for Video Technology*, v. 14, n. 11, p. 1236–1248, nov. 2004.
- [42] J. TAKALA & J. NIKARA, Unified pipeline architecture for discrete sine and cosine transforms of type IV, In: **Proceedings of the 3-rd Internacional Conference on Information Communication and Signal Processing**, 2001.
- [43] R. OLIVEIRA, R. CINTRA, F. BAYER, & C. J. TABLADA, Uma aproximação ortogonal para a DCT, In: **XXXI Simpósio Brasileiro de Telecomunicações**, 2013.
- [44] N. BRAHIMI & S. BOUGUEZEL, An efficient fast integer DCT transform for images compression with 16 additions only, In: **Int'l Workshop on Systems, Signal Processing and their Applications**, 2011, p. 71–74.
- [45] K. MIETTINEN, **Nonlinear Multiobjective Optimization**, ser. International Series in Operations Research and Management Science. Kluwer Academic Publishers, Dordrecht, 1999, v. 12.
- [46] M. EHRGOTT, **Multicriteria optimization**, ser. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag, 2000.

- [47] R. OLIVEIRA, Aproximações para a DCT baseadas em medida angular: Baixa complexidade e compressão de imagens, Trabalho de conclusão de curso da UFPE, 2016.
- [48] M. JRIDI, A. ALFALOU, & P. K. MEHER, A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal approximation of DCT, *IEEE Transactions on Circuits and Systems I*, v. 62, n. 2, p. 449–457, 2015.
- [49] T. L. T. DA SILVEIRA, R. S. OLIVEIRA, F. M. BAYER, R. J. CINTRA, & A. MADANAYAKE, Multiplierless 16-point DCT approximation for low-complexity image and video coding, *Signal, Image and Video Processing*, v. 11, n. 2, p. 227–233, 2017.
- [50] T. L. DA SILVEIRA, F. M. BAYER, R. J. CINTRA, S. KULASEKERA, A. MADANAYAKE, & A. J. KOZAKEVICIUS, An orthogonal 16-point approximate DCT for image and video compression, *Multidimensional Systems and Signal Processing*, v. 27, n. 1, p. 87–104, 2016.
- [51] N. AHMED, T. NATARAJAN, & K. RAO, Discrete cosine transform, *IEEE Transactions on Computers*, v. 23, n. 1, p. 90–93, 1974.
- [52] W. L. BRIGGS & V. E. HENSON, **The DFT: an owner’s manual for the discrete Fourier transform**. SIAM, 1995.
- [53] R. BLAHUT, **Fast Algorithms for Signal Processing**. Cambridge University Press, 2010. [Online]. Disponível: <https://books.google.com.br/books?id=8sbS2Bi4BmIC>
- [54] M. HEIDEMAN, **Multiplicative complexity, convolution, and the DFT**. Springer, 1988.
- [55] R. N. BRACEWELL, Discrete Hartley transform, *Journal of the Optical Society of America*, v. 73, n. 12, p. 1832–1835, Dec 1983.
- [56] Z. WANG, Fast algorithms for the discrete W transform and for the discrete Fourier transform, *IEEE Transactions on Acoustics, Speech and Signal Processing*, v. ASSP-32, p. 803–816, Aug. 1984.
- [57] M. VETTERLI & H. NUSSBAUMER, Simple FFT and DCT algorithms with reduced number of operations, *Signal Processing*, v. 6, p. 267–278, Aug. 1984.
- [58] N. SUEHIRO & M. HATORI, Fast algorithms for the DFT and other sinusoidal transforms, *IEEE Transactions on Acoustic, Signal, and Speech Processing*, v. 34, n. 6, p. 642–644, 1986.

- [59] H. S. HOU, A fast recursive algorithm for computing the discrete cosine transform, *IEEE Transactions on Acoustic, Signal, and Speech Processing*, v. 6, n. 10, p. 1455–1461, 1987.
- [60] Y. ARAI, T. AGUI, & M. NAKAJIMA, A fast DCT-SQ scheme for images, *Transactions of the IEICE*, v. E-71, n. 11, p. 1095–1097, nov. 1988.
- [61] Z. WANG & A. C. BOVIK, Reduced- and no-reference image quality assessment, *IEEE Signal Processing Magazine*, v. 28, n. 6, p. 29–40, Nov 2011.
- [62] S. SAPONARA, Real-time and low-power processing of 3D direct/inverse discrete cosine transform for low-complexity video codec, *Journal of Real-Time Image Processing*, v. 7, n. 1, p. 43–53, mar 2012.
- [63] A. EDIRISURIYA, A. MADANAYAKE, V. S. DIMITROV, R. J. CINTRA, & J. ADIKARI, VLSI architecture for 8-point AI-based Arai DCT having low area-time complexity and power at improved accuracy, *Journal of Low Power Electronics and Applications*, v. 2, n. 2, p. 127–142, 2012.
- [64] V. LECUIRE, L. MAKKAOUI, & J.-M. MOUREAUX, Fast zonal DCT for energy conservation in wireless image sensor networks, *Electronics Letters*, v. 48, n. 2, 2012.
- [65] G. K. WALLACE, The JPEG still picture compression standard, *IEEE Transactions on Consumer Electronics*, v. 30, n. 1, p. xviii–xxxiv, 1992.
- [66] N. J. HIGHAM, Computing the polar decomposition—with applications, *SIAM Journal on Scientific and Statistical Computing*, v. 7, n. 4, p. 1160–1174, ago. 1986.
- [67] R. J. CINTRA, An integer approximation method for discrete sinusoidal transforms, *Circuits, Systems, and Signal Processing*, 2011, to appear.
- [68] N. J. HIGHAM, Computing real square roots of a real matrix, *Linear Algebra and its Applications*, p. 88–89:405–430, 1987.
- [69] D. F. ELLIOTT & K. R. RAO, **Fast transforms algorithms, analyses, applications**. Elsevier, 1983.
- [70] K. J. HORADAM, **Hadamard Matrices and Their Applications**. Princeton University Press, 2007.

- [71] A. K. JAIN, **Fundamentals of Digital Image Processing**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989.
- [72] G. A. SEBER, **A matrix handbook for statisticians**. John Wiley & Sons, 2008, v. 15.
- [73] B. FLURY & W. GAUTSCHI, An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form, *SIAM Journal on Scientific and Statistical Computing*, v. 7, n. 1, p. 169–184, 1986.
- [74] D. S. WATKINS, **Fundamentals of Matrix Computations**, ser. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2004.
- [75] V. A. COUTINHO, R. J. CINTRA, & F. M. BAYER, Low-complexity multidimensional DCT approximations for high-order tensor data decorrelation, *IEEE Transactions on Image Processing*, v. 26, n. 5, p. 2296–2310, May 2017.
- [76] T. U.-S. IMAGE DATABASE. (2011) University of Southern California, Signal and Image Processing Institute. <http://sipi.usc.edu/database/>.
- [77] S. M. KAY, **Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory**, ser. Prentice Hall Signal Processing Series, A. V. OPPENHEIM, Ed. Upper Saddle River, NJ: Prentice Hall, 1993, v. 1.
- [78] Q. H. THU & M. GHANBARI, Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, v. 44, n. 13, p. 800–801, 2008.
- [79] Z. WANG, A. C. BOVIK, H. R. SHEIKH, & E. P. SIMONCELLI, Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing*, v. 13, n. 4, p. 600–612, 2004.
- [80] Z. WANG & A. C. BOVIK, Reduced- and no-reference image quality assessment, *IEEE Signal Processing Magazine*, v. 28, n. 6, p. 29–40, 2011.

QUANTIZAÇÃO EM TRANSFORMAÇÕES 2-D

Seja \mathbf{A} um bloco $N \times N$, então a sua transformação 2-D por uma aproximação $\hat{\mathbf{C}}$ é dada por:

$$\mathbf{B} = \hat{\mathbf{C}} \cdot \mathbf{A} \cdot \hat{\mathbf{C}}^{\top},$$

$$\mathbf{B} = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^{\top} \cdot \mathbf{S}.$$

Considerando

$$\hat{\mathbf{B}} = \hat{\mathbf{T}} \cdot \mathbf{A} \cdot \hat{\mathbf{T}}^{\top},$$

temos,

$$\mathbf{B} = \mathbf{S} \cdot \hat{\mathbf{B}} \cdot \mathbf{S},$$

$$\mathbf{B} = \hat{\mathbf{R}} \odot \hat{\mathbf{B}},$$

sendo o operador \odot a multiplicação matricial ponto-a-ponto e,

$$\hat{\mathbf{R}} = \text{diag}(\hat{\mathbf{S}}) \cdot \text{diag}(\hat{\mathbf{S}})^{\top}.$$

Em aplicações de compressão de imagem é fornecida uma matriz de quantização \mathbf{Q} [65]. A depender da aproximação considerada teremos diferentes matrizes \mathbf{Q} . Em aplicações de compressão de imagens uma matriz de coeficientes quantizados $\bar{\mathbf{B}}$ é calculada a partir de [65]:

$$\bar{\mathbf{B}} = \text{round}(\mathbf{B} \div \mathbf{Q}),$$

em que \div denota divisão matricial elemento a elemento.

Desta forma, temos,

$$\bar{\mathbf{B}} = \text{round}(\mathbf{R} \odot \hat{\mathbf{B}} \div \mathbf{Q}),$$

$$\bar{\mathbf{B}} = \text{round}(\hat{\mathbf{B}} \div \tilde{\mathbf{Q}}),$$

em que $\tilde{\mathbf{Q}} = \mathbf{Q} \div \mathbf{R}$.

Portanto, podemos ver que a matriz diagonal de ajuste das transformadas aproximadas pode ser suprimida na etapa da quantização.

IMAGENS COMPRIMIDAS

Aqui os resultados serão apresentados de maneira qualitativa e as imagens comprimidas podem ser visualizadas. Nas Figuras B.1, B.2, B.3 e B.4, podem ser visualizadas as imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 5, 10, 20, 30$, respectivamente. Nas Figuras B.5, B.6, B.7 e B.8, podem ser visualizadas as imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 20, 40, 80, 120$, respectivamente. Nas Figuras B.9, B.10, B.11 e B.12, podem ser visualizadas as imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 5, 10, 20, 30$, respectivamente. Os valores de r considerados para os diferentes comprimentos das transformadas indicam, aproximadamente, compressão de 92%, 84%, 69% e 53%. Foi considerada a imagem Lena padrão.



Figura B.1: *Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 5$.*



Figura B.2: *Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 10$.*



Figura B.3: *Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 20$.*



Figura B.4: *Imagens comprimidas com as transformadas de comprimento 8 consideradas e $r = 30$.*



Figura B.5: *Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 20$.*



Figura B.6: *Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 40$.*



Figura B.7: *Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 80$.*



Figura B.8: *Imagens comprimidas com as transformadas de comprimento 16 consideradas e $r = 120$.*



Figura B.9: *Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 80$.*



Figura B.10: *Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 160$.*



Figura B.11: *Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 320$.*



Figura B.12: *Imagens comprimidas com as transformadas de comprimento 32 consideradas e $r = 480$.*

SOBRE O AUTOR

O autor nasceu em Porto Alegre, Rio Grande do Sul, Brasil, no dia 27 de junho de 1991. Formou-se em Estatística, pela Universidade Federal de Santa Maria em 2015/2. Seus interesses de pesquisa incluem Estatística clássica, Modelos de Regressão, Teoria Assintótica e Processamento Digital de Sinais.

Endereço: Endereço

e-mail: diegocanterle@gmail.com

Esta dissertação foi diagramada usando $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ ¹ pelo autor.

¹ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ é uma extensão do $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ é uma coleção de macros criadas por Leslie Lamport para o sistema $\text{T}_{\text{E}}\text{X}$, que foi desenvolvido por Donald E. Knuth. $\text{T}_{\text{E}}\text{X}$ é uma marca registrada da Sociedade Americana de Matemática ($\mathcal{A}\mathcal{M}\mathcal{S}$). O estilo usado na formatação desta dissertação foi escrito por Dinesh Das, Universidade do Texas. Modificado por Renato José de Sobral Cintra (2001) e por Andrei Leite Wanderley (2005), ambos da Universidade Federal de Pernambuco. Sua última modificação ocorreu em 2010 realizada por José Sampaio de Lemos Neto, também da Universidade Federal de Pernambuco.