

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

APROXIMAÇÕES PARA A DCT BASEADAS NOS ALGORITMOS
DE FEIG-WINOGRAD E CHEN

CLAUDIO JAVIER TABLADA

Recife, fevereiro de 2014

APROXIMAÇÕES PARA A DCT BASEADAS NOS ALGORITMOS
DE FEIG-WINOGRAD E CHEN

CLAUDIO JAVIER TABLADA

Orientador: Prof. Dr. RENATO J. CINTRA
Co-orientador: Prof. Dr. FÁBIO M. BAYER

Área de Concentração: Estatística Aplicada

Dissertação submetida como requerimento parcial para obtenção do grau
de Mestre em Estatística pela Universidade Federal de Pernambuco

Recife, fevereiro de 2014

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Tablada, Claudio Javier

Aproximações para a DCT baseadas nos algoritmos de Feig-Winograd e Chen / Claudio Javier Tablada. - Recife: O Autor, 2014.

120 f.: il., fig., tab.

Orientador: Renato José de Sobral Cintra.

Dissertação (mestrado) - Universidade Federal de Pernambuco. CCEN, Estatística, 2014.

Inclui referências e apêndices.

1. Estatística aplicada. 2. Processamento de imagens. I. Cintra, Renato José de Sobral (orientador). II. Título.

310

CDD (23. ed.)

MEI2014 – 033

Universidade Federal de Pernambuco
Pós-Graduação em Estatística

21 de fevereiro de 2014

Nós recomendamos que a Dissertação de Mestrado de autoria de

Claudio Javier Tablada

Intitulada:

“Aproximações para a DCT baseadas nos Algoritmos de Feig-Winograd e Chen”

Seja aceita como cumprimento parcial dos requerimentos para o grau de Mestre em Estatística.

Coordenador da Pós-Graduação em Estatística

Banca Examinadora:

Renato José de Sobral Cintra

Orientador/UFPE

Alejandro César Frery Orgambide

UFAL

Hélio Magalhães de Oliveira

UFPE-CTG

Este documento será anexado à versão final da tese.

Dedicado a mi madre
“Todo lo que soy en la vida, fue tu causa”

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.

Seja a sua melhor versão !

Agradecimentos

A minha família, especialmente a minha mãe, Leontina Margarita, por seu amor incondicional.

A meu amor, Renilma Pereira, por compartilhar momentos de sua vida comigo e por seu apoio.

A meu amigo e colega, Giannini Italino, pelo apoio e a força outorgada ao longo do mestrado e por sua amizade.

A meu amigo, prof. Germán Torres, e aos professores Alejandro Frery e Oscar Bustos, pela ajuda, guia e apoio.

A meu orientador, prof. Renato J. Cintra, e meu co-orientador, prof. Fábio M. Bayer, pela paciência e dedicação outorgada em minha formação acadêmica, a orientação diligente, o apoio e a total confiança em mim.

A todos os professores que fizeram parte de minha formação acadêmica.

A todos os colegas de mestrado e do grupo de processamento de sinais, especialmente a Raabe Marques, Fernando Peña, Maria Priscila Souza, Cristiany Barros e Raíza Oliveira. A amizade e a hospitalidade de todos contribuíram para que eu, um estrangeiro, sinta-se como estivesse em sua própria casa.

A Luz Milena, por sua ajuda amável e generosa.

A Valéria Bittencourt, por sua bondade e diligência.

Aos membros da banca examinadora, pelas sugestões.

A CAPES, pelo apoio financeiro.

Resumo

Nos últimos anos, a comunidade de processamento e análise de sinais tem apresentado contribuições teóricas e práticas objetivando a proposição de aproximações para a transformada discreta do cosseno (DCT). A DCT tem a importância de ser a ferramenta matemática central empregada em vários padrões de compressão de imagens e vídeo, tais como JPEG, MPEG-1, MPEG-2, H.261, H.263, H.264 e o recente HEVC. Aproximações para a DCT são usualmente livres de multiplicação e podem ser implementadas em hardware com baixo custo computacional. Nesta dissertação é realizada uma revisão da literatura de aproximações para a DCT com os principais resultados obtidos neste campo. Como contribuições originais, são propostas: (i) uma classe de aproximações para a DCT baseada na parametrização da fatoração de Feig-Winograd e (ii) duas aproximações baseadas na fatoração de Chen. Para a classe de aproximações baseada na fatoração de Feig-Winograd, foi considerado um problema de otimização multiobjetivo para selecionar transformadas ótimas com respeito a algumas medidas objetivas de qualidade, tais como erro de energia, erro quadrático médio, ganho de codificação e eficiência da transformada. As aproximações introduzidas neste trabalho são avaliadas no contexto de compressão de imagens e comparadas com aproximações descritas na literatura. Para esta avaliação foram consideradas a relação sinal-ruído de pico e o índice de similaridade estrutural como figuras de mérito. Dos resultados obtidos, conclui-se que as novas aproximações propostas resultam ser boas transformadas para serem usadas no contexto de compressão de imagens em aplicações que requerem baixo custo de implementação.

Palavras-chave: Transformada discreta do cosseno. Transformadas aproximadas. Complexidade aritmética. Compressão de imagem.

Abstract

In recent years, the community of signal processing has been contributing to various theoretical and practical issues in order to obtain approximations for the discrete cosine transform (DCT). The DCT has the significance of being the central mathematical tool used in many compression standards for images and video, such as JPEG, MPEG-1, MPEG-2, H.261, H.263, H.264 and recently the HEVC. DCT approximations are usually multiplierless and can be implemented in hardware with low computational cost. In this dissertation, an extensive review of the approximate DCT literature is offered; being the main results summarized. Among the original contributions of this work, emphasize: (i) a new class of DCT approximations based on a parameterization of the Feig-Winograd factorization and (ii) two approximations based on Chen's factorization. For the class of approximations based on Feig-Winograd factorization, we considered a multi-objective optimization problem to select optimal transforms with respect to objective measures of quality, such as energy error, mean square error, coding gain, and transform efficiency. Proposed approximations were evaluated in the framework of image compression and compared with state-of-the-art methods in terms of peak signal-to-noise ratio and structural similarity of compressed images.

Keywords: Discrete cosine transform. Approximated transforms. Arithmetic complexity. Image compression.

Sumário

1. Introdução	11
1.1. Motivação e Contexto	11
1.2. Estado da Arte	12
1.3. Objetivos	14
1.4. Estrutura do Documento	15
2. A Transformada Discreta do Cosseno	17
2.1. Introdução	17
2.2. Análise de Componentes Principais e a KLT	17
2.3. A DCT como uma Aproximação para a KLT	20
2.4. Definição da DCT	22
2.5. Algoritmos rápidos para a DCT	27
3. Aproximações da DCT	30
3.1. Introdução	30
3.2. Métodos Aproximados para implementar a DCT	31
3.2.1. A Transformada de Walsh-Hadamard	33
3.2.2. A SDCT	35
3.2.3. Aproximação de nível 1 de Lengwehasatit e Ortega	36
3.2.4. Série de Aproximações BAS	36
3.2.5. A DCT arredondada	37
3.2.6. A DCT arredondada modificada	39

3.2.7.	Aproximação para <i>RF imaging</i>	39
3.3.	Custo aritmético das Aproximações	40
4.	Aproximações Baseadas na Fatoração de Feig-Winograd	42
4.1.	Introdução	42
4.2.	Fatoração de Feig-Winograd	43
4.3.	O Subespaço de Feig-Winograd	45
4.4.	Transformada Inversa	47
	Diagrama de fluxo de sinal da transformada inversa	51
4.5.	Aproximações Conhecidas	52
4.6.	CrITÉrios de Aproximação	54
4.6.1.	Complexidade Computacional	54
4.6.2.	Ortogonalidade e Quasi-ortogonalidade	56
4.6.3.	Estrutura e Complexidade da Transformada Inversa	59
4.6.4.	Medidas de Proximidade	60
	Erro de energia total	61
	Erro quadrático médio	62
	Ganho em codificação unificado	63
	Eficiência da transformada	64
4.7.	Otimização sobre o Subespaço de Feig-Winograd	64
4.7.1.	Otimização Multicritério	64
	A ordem de Pareto em \mathbb{R}^p	65
	Resolução do problema de otimização	65
	Suporte computacional	67
4.7.2.	Soluções Eficientes e Transformadas Ótimas	67
	Aproximações Ortonormais	69
	Aproximação Quasi-ortogonal	72
4.7.3.	Avaliação das Novas Transformadas	73
4.7.4.	Discussão e Comparação	74
4.8.	Compressão de Imagens	78
4.8.1.	Metodologia	78
4.8.2.	Resultados e discussão	79
4.9.	Conclusão	81

5. Aproximações da DCT Baseadas na Fatoração de Chen	85
5.1. Introdução	85
5.2. Fatoração de Chen para a DCT	86
5.3. Aproximações Propostas	88
5.4. Medida de Proximidade e Custo Computacional	92
5.5. Aplicações em Compressão de Imagens	94
5.5.1. Compressão tipo JPEG e Medidas de Erro	94
5.5.2. Resultados	95
5.5.3. Análise de artefatos de blocagens (<i>blocking artifact</i>)	95
5.6. Conclusão	98
6. Conclusões finais	99
Referências	104
A. Compressão de Imagens tipo JPEG	113
B. Transformação 2-D e Etapa de Quantização	115
C. Acrônimos usados	119

Introdução

1.1 Motivação e Contexto

A Transformada Discreta do Cosseno (DCT) é uma ferramenta essencial em processamento de sinais [15, 65]. Embora existam diversas versões da DCT, a mais utilizada, principalmente no contexto de compressão de imagens, é a versão introduzida por Ahmed *et al.* [2], referenciada como a DCT tipo II (DCT-II). Isto acontece principalmente porque a DCT-II é assintoticamente equivalente à Transformada de Karhunen-Loève (KLT), quando o sinal de entrada considerado pode ser modelado como um processo de Markov de primeira ordem [2, 15]. Dado que a KLT tem propriedades ótimas de decorrelação e compactação de energia, a DCT-II herda tais propriedades para esta classe de sinais. Como é conhecido no contexto de processamento de imagens, muitas imagens naturais podem ser modeladas como processos markovianos de primeira ordem altamente correlacionados.

Em particular, a DCT-II de comprimento 8 é adotada em diversos padrões de compressão de imagens e vídeo, tais como JPEG [81], MPEG-1 [67], MPEG-2 [43], H.261 [44], H.263 [45], H.264 [90] e o recente HEVC [63]. Como consequência, vários algoritmos rápidos para o cômputo eficiente da DCT-II foram propostos [4, 18, 26, 40, 53, 56, 76, 80, 84, 91]. Estes algoritmos baseiam-se em fatorações matriciais com o objetivo de reduzir o número de multiplicações para implementar a DCT-II. Contudo, eles ainda precisam de aritmética de ponto flutuante, que demanda maiores exigências de hardware. Além disso, dado o grande avanço nesta área, alguns dos mé-

todos de fatoração existentes atingem o limite inferior teórico do custo aritmético da DCT-II [27]. Portanto, não é esperado que novas fatorações permitam diminuir ainda mais a complexidade para implementar a DCT-II de forma exata. Assim, métodos aproximados oferecem uma alternativa viável para reduzir o custo aritmético da DCT-II [4, 33, 55, 56, 57]. Embora não calculem a DCT-II exatamente, estas aproximações são estimativas úteis que demandam baixa complexidade computacional. Neste contexto, na literatura de processamento de sinais são introduzidos diferentes métodos aproximados para o cômputo da DCT-II. Uma alternativa é o uso de racionais diádicos, tais como as aproximações baseadas em *lifting scheme* [55], conseguindo assim obter aproximações livres de multiplicações, com o custo de incrementar a quantidade de adições e deslocamentos (*bit-shifting*) para a sua implementação.

Outra alternativa para reduzir o custo aritmético é considerar aproximações cujas fatorações envolvem somente elementos no conjunto $\mathcal{P} = \{0, \pm \frac{1}{2}, \pm 1, \pm 2\}$. Assim, tais aproximações têm complexidade multiplicativa nula e não incrementam o número de adições, necessitando somente de uma quantidade reduzida de deslocamentos de bits para a sua implementação. Estas aproximações podem substituir a DCT exata em implementação de hardware e em processamento/cômputo de alta velocidade [15], pois requerem baixo custo de *hardware* e baixa demanda de potência [23]. Com efeito, as aproximações para a DCT-II são consideradas para aplicações em transmissão e processamento de vídeo em tempo real [51, 68], sistemas de comunicação [15], aplicações de computação móvel [15], redes de antenas inteligentes de rádio-frequência [62] e redes de sensores sem fio [52], além de outras aplicações.

Pelas razões anteriormente apresentadas, neste trabalho considera-se somente aproximações para a DCT-II. Portanto, referenciamos a DCT-II simplesmente como a DCT, explicitando o tipo de DCT somente quando seja requerido pelo contexto.

1.2 Estado da Arte

A primeira aproximação para a DCT cuja fatoração para implementá-la envolve somente elementos no conjunto $\mathcal{P} = \{0, \pm \frac{1}{2}, \pm 1, \pm 2\}$ foi introduzida por Haweel [32]

e é referenciada como a *Signed DCT* (SDCT). Esta transformada é definida como

$$\mathbf{T} = \text{sign}(\mathbf{C}), \quad (1.1)$$

em que \mathbf{C} denota a matriz da DCT e $\text{sign}(\cdot)$ é a função sinal definida por:

$$\text{sign}(x) = \begin{cases} 1, & \text{se } x > 0, \\ 0, & \text{se } x = 0, \\ -1, & \text{se } x < 0. \end{cases} \quad (1.2)$$

A função sinal, quando aplicada sobre matrizes, opera elemento a elemento.

Em [32], foi proposto um algoritmo rápido para implementar a SDCT que requer de 24 adições. Contudo, no artigo seminal da SDCT não é proposto um algoritmo rápido para a SDCT inversa. Em [60] é derivado um algoritmo rápido para a SDCT inversa que requer de 20 adições e 4 deslocamentos. Embora esta transformada seja não-ortogonal, sua transformada inversa também possui elementos no conjunto \mathcal{P} e, portanto, é de baixa complexidade computacional.

Depois da introdução da SDCT, uma grande quantidade de algoritmos de baixo custo aritmético têm sido considerados para computar a DCT de forma aproximada. Em [54], Lengwehasatit e Ortega introduzem uma série de aproximações para a DCT. Em particular, a aproximação de nível 1 é uma transformada de baixo custo aritmético com elementos no conjunto \mathcal{P} . Esta transformada requer 24 adições e 2 deslocamentos para a sua implementação e apresenta um bom desempenho no contexto de compressão de imagens em termos da relação sinal-ruído de pico (PSNR) [78]. Posteriormente, Bouguezal, Ahmad e Swamy introduzem uma série de aproximações de baixo custo aritmético que referenciamos como a série BAS [7, 8, 9, 10, 12, 13]. Estas transformadas apresentam bom desempenho, principalmente em termos de ganho em codificação, sendo adequadas para aplicações em compressão de imagens. Especificamente, as transformadas BAS introduzidas em [7, 8, 9, 10, 13] são obtidas como variações de aproximações prévias, principalmente da SDCT. Contudo, a transformada BAS introduzida em [12] é obtida por inspeção direta da estrutura da DCT de ordem 4. A transformada BAS introduzida em [9] é não-ortogonal, embora sua transformada inversa mantenha a baixa complexidade computacional atingida pela transformada direta. Outras duas transformadas destacadas por seu desempenho em compressão de

imagens são introduzidas por Cintra e Bayer [5, 20], conhecidas como a *Rounded DCT* (RDCT) e a *Modified Rounded DCT* (MRDCT). A RDCT introduzida em [20] é obtida escalando a DCT de comprimento 8 por um fator de 2 e aplicando a função de arredondamento (*round-off*). Já a MRDCT, introduzida em [5], é obtida modificando convenientemente a RDCT. Em particular, a MRDCT possui o menor custo aritmético atingido por aproximações para a DCT, requerendo somente 14 adições para a sua implementação. Outra transformada de baixo custo aritmético destacada neste contexto é a aproximada introduzida em [62], a qual é obtida através de um processo de otimização baseado na estrutura da DCT exata. Todas as transformadas consideradas apresentam um bom equilíbrio entre desempenho e custo aritmético, sendo adequadas para implementações que exijam baixa complexidade computacional e alta velocidade de processamento.

1.3 Objetivos

Os métodos listados na literatura para gerar aproximações de baixo custo aritmético incluem:

- i) aproximações brutas [20, 32];
- ii) inspeção [12];
- iii) variações de aproximações prévias [5, 7, 8, 9, 10, 13];
- iv) procedimentos de otimização simples baseados na estrutura da DCT [62].

As aproximações existentes aparecem como casos isolados, sem um formalismo matemático unificado. Assim, o objetivo dos Capítulos 4 e 5 é introduzir uma nova técnica para obter aproximações baseadas em duas fatorações para a DCT: a fatoração de Feig-Winograd [26] e a fatoração de Chen *et al.* [18]. Este método generaliza as técnicas anteriores e pode, em princípio, ser aplicado usando qualquer fatoração para a DCT, o qual permite obter uma grande quantidade de aproximações de baixo custo aritmético. No caso da fatoração de Feig-Winograd discutida no Capítulo 4, a parametrização desta fatoração induz um subespaço de aproximações. Portanto, um objetivo é

obter aproximações neste subespaço que apresentem bom desempenho segundo medidas de avaliação normalmente adotadas no contexto de compressão de imagens. Para isso, é apresentado um problema de otimização que permite selecionar aproximações ótimas com respeito a essas medidas de avaliação. Finalmente, o objetivo do Capítulo 5 é introduzir aproximações de baixo custo aritmético baseadas na fatoração de Chen. Todas as aproximações são avaliadas no contexto de compressão de imagens segundo medidas objetivas de qualidade.

1.4 Estrutura do Documento

Este trabalho está organizado da seguinte forma:

No Capítulo 2, são introduzidas as bases teóricas e as considerações de implementação da DCT. Na Seção 2.2 é apresentada a KLT, conhecida por suas propriedades de decorrelação e compactação de energia. Posteriormente, na Seção 2.3, é derivada a DCT-II como uma transformada assintoticamente equivalente à KLT quando o sinal de entrada é um processo markoviano de primeira ordem e, na Seção 2.4, são introduzidas diferentes versões da DCT. Na Seção 2.5 são apresentados alguns dos algoritmos rápidos mais usados para implementar a DCT.

No Capítulo 3, são apresentadas as aproximações para a DCT de comprimento 8 consideradas neste trabalho. A Seção 3.2 descreve, de forma geral, a representação numérica empregada para implementar a DCT, tais como os racionais diádicos ou as potências de 2, que permitem obter aproximações para a DCT de baixo custo aritmético. Nas Seções de 3.2.1 à 3.2.7 são apresentadas algumas das aproximações de baixa complexidade computacional para a DCT de comprimento 8 mais conhecidas na literatura. Finalmente, na Seção 3.3, é apresentado o custo aritmético destas aproximações.

O Capítulo 4 introduz um método geral para a obtenção de aproximações para a DCT baseadas na fatoração de Feig-Winograd [26]. Assim, nas Seções 4.2 e 4.4 são apresentados a estrutura matemática da classe de transformadas propostas e os algoritmos rápidos para as transformações direta e inversa. Na Seção 4.5 são apresentadas algumas das aproximações já conhecidas na literatura que pertencem à classe de aproximações propostas. A seguir, na Seção 4.6, são discutidas as propriedades desejáveis que devem ser satisfeitas pelas aproximações da DCT, tais como baixo custo aritmé-

tico, ortogonalidade ou quasi-ortogonalidade, invertibilidade e proximidade à DCT exata. Posteriormente, na Seção 4.7, é proposto um problema de otimização multicritério sobre um conjunto de medidas de avaliação com o objetivo de identificar soluções eficientes e novas transformadas ótimas dentro da classe de transformadas propostas. As aproximações obtidas são avaliadas e comparadas com outras aproximadas já conhecidas. Na Seção 4.8 é feita uma análise de compressão de imagens utilizando as transformadas ótimas obtidas, considerando medidas objetivas de qualidade. Finalmente, na Seção 4.9, são dadas as conclusões do capítulo.

No Capítulo 5, são obtidas duas aproximações para a DCT baseadas na fatoração de Chen *et al.* [18]. Na Seção 5.2 é apresentada a estrutura matemática da fatoração de Chen e, na Seção 5.3, são introduzidas as duas aproximações propostas. Na Seção 5.4 é apresentada a medida de avaliação usada para avaliar o desempenho das aproximações propostas e seus custos aritméticos. Posteriormente, na Seção 5.5, é feita uma análise de compressão de imagens considerando as duas aproximações obtidas usando medidas objetivas de qualidade. A Seção 5.6 conclui o capítulo.

Finalmente, no Capítulo 6, são apresentadas as conclusões finais do presente trabalho. O esquema da compressão tipo JPEG é dado no Apêndice A e a teoria sobre as etapas de transformação 2-D e de quantização, que fazem parte do esquema de compressão, são dadas no Apêndice B. O Apêndice C apresenta a lista dos acrônimos usados.

A Transformada Discreta do Cosseno

2.1 Introdução

Neste capítulo, descrevemos o formalismo teórico da DCT e observamos que resulta ser assintoticamente equivalente à KLT. Isto acontece quando o sinal de entrada considerado é um processo markoviano estacionário de primeira ordem, em que o coeficiente de correlação ρ do sinal tende para a unidade ($\rho \rightarrow 1$). Dado que a KLT tem boas propriedades de decorrelação e compactação de energia, estas propriedades são refletidas na própria DCT, o que faz dela uma ferramenta crucial no contexto de compressão de imagens. Adicionalmente, são mencionados alguns dos algoritmos rápidos mais usados para implementar a DCT, os quais permitem diminuir sua complexidade computacional.

2.2 Análise de Componentes Principais e a KLT

No contexto de processamento de sinais, um sinal pode ser considerado como um vetor dado por:

$$\mathbf{x} = [x_0 \ x_1 \ \cdots \ x_{N-1}]^T \in \mathbb{R}^N.$$

Imagens geralmente podem ser modeladas como um processo de Markov estacionário de primeira ordem altamente correlacionado [64, p. 18]. Portanto, em muitas

operações de processamento (transmissão, compressão, etc.), é importante descorrelacionar o sinal \mathbf{x} através de uma transformada \mathbf{W}^\top tal que, se $\mathbf{y} = \mathbf{W}^\top \cdot \mathbf{x}$, então a matriz de covariância do sinal $\mathbf{y} = [y_0 \ y_1 \ \cdots \ y_{N-1}]^\top$ seja diagonal [15, p. 52]:

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

em que $E(\cdot)$ denota o operador esperança, $\text{diag}(\cdot)$ retorna uma matriz diagonal de seus argumentos e

$$\lambda_k = E(y_k^2), \quad k = 0, 1, \dots, N-1.$$

Observar que (2.1) pode ser expresso como:

$$\Sigma_{\mathbf{y}} = E(\mathbf{W}^\top \cdot \mathbf{x} \cdot \mathbf{x}^\top \cdot \mathbf{W}) = \mathbf{W}^\top \cdot E(\mathbf{x} \cdot \mathbf{x}^\top) \cdot \mathbf{W} = \mathbf{W}^\top \cdot \Sigma_{\mathbf{x}} \cdot \mathbf{W}, \quad (2.2)$$

em que $\Sigma_{\mathbf{x}}$ é a matriz de covariância do sinal \mathbf{x} .

Com o objetivo de que a transformada \mathbf{W}^\top preserve a energia do sinal, impõe-se que \mathbf{W} é uma matriz ortogonal. Portanto, de (2.2) segue que:

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

Dado que, por construção, a matriz de covariância $\Sigma_{\mathbf{x}}$ é real e simétrica [30, 69], é possível achar um conjunto de N autovetores ortonormais de $\Sigma_{\mathbf{x}}$. Assim, pelo teorema da fatoração espectral [71, p. 296], temos que a matriz \mathbf{W} pode ser determinada resolvendo o seguinte problema de autovalores:

$$\Sigma_{\mathbf{x}} \cdot \mathbf{w}_k = \lambda_k \cdot \mathbf{w}_k, \quad k = 0, 1, \dots, N-1, \quad (2.3)$$

em que $\lambda_k > 0 \ \forall k$. Portanto, λ_k é o autovalor da matriz $\Sigma_{\mathbf{x}}$ associado ao autovetor \mathbf{w}_k , que resulta ser a k -ésima coluna da matriz \mathbf{W} . Se os autovalores são ordenados em ordem decrescente, $\lambda_{i_0} > \lambda_{i_1} > \cdots > \lambda_{i_{N-1}}$, então a matriz \mathbf{W} obtida é conhecida como a Transformada de *Karhunen-Loève* ou Transformada de *Hotelling* [30, p. 678].

Com o objetivo de diminuir a dimensionalidade do conjunto de dados, é comum considerar a Análise de Componentes Principais [30, p. 675], a qual considera so-

mente r autovetores (com $r < N$) associados aos r maiores autovalores da matriz de covariância, e definir a transformada \mathbf{W}_r como a matriz $r \times N$ que tem como linhas aos r autovetores considerados. Seja $\hat{\mathbf{x}}$ o sinal reconstruído definido por:

$$\hat{\mathbf{x}} = \mathbf{W}_r \cdot \mathbf{y}.$$

Dado que os autovetores \mathbf{w}_k , $k = 0, 1, \dots, N-1$, fornecem uma base ortonormal de \mathbb{R}^N , então existem escalares α_k , tais que:

$$\begin{aligned} \mathbf{x} &= \sum_{k=0}^{N-1} \alpha_k \cdot \mathbf{w}_k, \\ \hat{\mathbf{x}} &= \sum_{k=0}^r \alpha_k \cdot \mathbf{w}_k. \end{aligned}$$

O erro quadrático médio (MSE) entre \mathbf{x} e $\hat{\mathbf{x}}$ é dado por [85]:

$$\begin{aligned} E((\mathbf{x} - \hat{\mathbf{x}})^\top (\mathbf{x} - \hat{\mathbf{x}})) &= E\left(\sum_{k=r+1}^{N-1} \alpha_k^2\right) \\ &= \sum_{k=r+1}^{N-1} \mathbf{w}_k^\top \cdot E(\mathbf{x} \cdot \mathbf{x}^\top) \cdot \mathbf{w}_k \\ &= \sum_{k=r+1}^{N-1} \mathbf{w}_k^\top \cdot \Sigma_{\mathbf{x}} \cdot \mathbf{w}_k \\ &= \sum_{k=r+1}^{N-1} \lambda_k \cdot \mathbf{w}_k^\top \cdot \mathbf{w}_k \\ &= \sum_{k=r+1}^{N-1} \lambda_k. \end{aligned} \tag{2.4}$$

Assim, o MSE pode ser minimizado ao considerar os autovetores associados aos r maiores autovalores na transformada \mathbf{W}_r . Portanto, a KLT é ótima no sentido de minimizar o MSE entre o sinal \mathbf{x} e sua aproximação $\hat{\mathbf{x}}$.

Segundo descrito anteriormente, a KLT tem as seguintes propriedades [15, p. 55]:

1. descorrelaciona completamente o sinal no domínio da transformada, como mos-

trado por (2.2);

2. minimiza o MSE na compressão dos dados, como indicado por (2.4);
3. concentra a maior variância (energia) em poucos coeficientes do sinal de saída.

Como ponto desfavorável, temos que a KLT não possui uma expressão em forma fechada para todo sinal de entrada, pois depende diretamente da matriz de correlação $\Sigma_{\mathbf{x}}$, que por sua vez depende do sinal de entrada \mathbf{x} . Portanto, a KLT não resulta em uma ferramenta prática no contexto de processamento de sinais [2, 15, 21].

2.3 A DCT como uma Aproximação para a KLT

Dada sua dependência do sinal de entrada e o fato de que não existe um algoritmo rápido para implementá-la, a KLT resulta ser uma ferramenta desejável mas pouco prática em processamento de sinais. Portanto, uma questão natural é ver quando é possível considerar vetores base predeterminados que sejam boas aproximações para os autovetores que diagonalizam a matriz de correlação de um sinal de entrada $\mathbf{x} \in \mathbb{R}^N$.

Em muitas situações, como acontece com imagens naturais, um sinal pode ser modelado como um processo de Markov estacionário de primeira ordem [64, p. 18]. Neste caso particular, a (m,n) -ésima entrada da matriz de correlação $\Sigma_{\mathbf{x}}$ do sinal de entrada $\mathbf{x} \in \mathbb{R}^N$ é dada por [2, 15]:

$$[\Sigma_{\mathbf{x}}]_{m,n} = \rho^{|m-n|}, \quad m,n = 0,1,\dots,N-1,$$

em que $0 < \rho < 1$ é o coeficiente de correlação. Como mostrado em [15, 66], neste caso, a m -ésima componente do k -ésimo autovetor que é solução de (2.3) é dada, para $k,m = 0,1,\dots,N-1$, por:

$$w_{k,m} = \sqrt{\frac{2}{N + \lambda_k}} \sin \left\{ \mu_k \left[(m+1) - \frac{N+1}{2} \right] + \frac{(k+1)\pi}{2} \right\}, \quad (2.5)$$

em que

$$\lambda_k = \frac{1 - \rho^2}{1 - 2\rho \cos(\mu_k) + \rho^2} \quad (2.6)$$

é o k -ésimo autovalor associado e μ_k , $k = 0, 1, \dots, N - 1$, são as raízes reais positivas da função transcendental:

$$\tan(N\mu) = -\frac{(1 - \rho^2) \sin(\mu)}{(1 + \rho^2) \cos(\mu) - 2\rho}. \quad (2.7)$$

Se assumimos que as componentes do sinal de entrada são altamente correlacionados, então $\rho \approx 1$. Neste caso, considerando $\rho \rightarrow 1$ na equação (2.7), obtemos:

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

Portanto, as N raízes reais positivas da equação (2.7) são dadas por:

$$\mu_k = \frac{k\pi}{N}, \quad k = 0, 1, \dots, N - 1. \quad (2.8)$$

Substituindo (2.8) em (2.6), obtemos que, quando $\rho \rightarrow 1$, resulta $\lambda_k = 0$ para $k \neq 0$. Além disso, de [71, p. 251], temos que o traço da matriz $\Sigma_{\mathbf{x}}$, definido por

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

é igual à soma dos N autovalores. Portanto:

$$\lim_{\rho \rightarrow 1} \text{tr}(\Sigma_{\mathbf{x}}) = \lim_{\rho \rightarrow 1} \sum_{n=0}^{N-1} [\Sigma_{\mathbf{x}}]_{n,n} = \lim_{\rho \rightarrow 1} \sum_{k=0}^{N-1} \lambda_k = \lambda_0.$$

Dado que $[\Sigma_{\mathbf{x}}]_{n,n} = 1$ para $n = 0, 1, \dots, N - 1$, então $\lambda_0 = N$ quando $\rho \rightarrow 1$.

Substituindo em (2.5), obtemos:

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

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

Introduzindo a constante c_k definida por:

$$c_k = \begin{cases} 1/\sqrt{2}, & \text{se } k = 0, \\ 1, & \text{se } k \neq 0, \end{cases}$$

podemos unificar as equações anteriores pela expressão:

$$w_{k,m} = \sqrt{\frac{2}{N}} c_k \cos \left\{ \frac{k(2m+1)\pi}{2N} \right\}, \quad k, m = 0, 1, \dots, N-1.$$

Cada valor de $w_{k,m}$ coincide com a (k,m) -ésima componente da DCT como definida em [2]. Dado que os autovetores w_k são as soluções de (2.3), temos que a DCT é assintoticamente equivalente à KLT quando $\rho \rightarrow 1$. Este fato explica por que a DCT tem boas propriedades de decorrelação e compactação de energia quando o sinal de entrada pode ser modelado como um processo markoviano estacionário de primeira ordem altamente correlacionado.

2.4 Definição da DCT

Ahmed *et al.* [2] propuseram a DCT-II como uma aproximação para a KLT quando o sinal considerado é um processo de Markov estacionário de primeira ordem altamente correlacionado. Dado que imagens naturais podem ser modeladas deste modo [64], temos que a DCT-II resulta ser uma transformada de grande utilidade no contexto de processamento de imagens [15, 30, 64]. Contudo, existem diversas versões da DCT, as quais diferem nos pontos de amostragem da função cosseno utilizados na definição de suas matrizes de transformação. As versões da DCT mais usadas são a DCT tipo I (DCT-I) introduzida em [88], a DCT-II e a DCT tipo III (DCT-III) definidas em [2] e a DCT tipo IV (DCT-IV) dada em [46].

Um sinal $\mathbf{x} = [x_0 \ x_1 \ \cdots \ x_{N-1}]^\top$ pode ser estendido periodicamente de duas formas simétricas. Estas extensões simétricas são ilustradas na Figura 2.1 e são referenciadas na literatura de processamento de sinais como extensão W (whole-point symmetry) e extensão H (half-point symmetry). No caso da extensão W, a extensão é feita considerando $x_{-n} = x_n$ para $n = 0, 1, \dots, N-1$. Assim, a periodicidade do sinal estendido é $2N - 2$. Para o caso da extensão H, a extensão é da forma $x_{-n} = x_{n-1}$ para $n = 1, \dots, N$. Neste caso, a periodicidade do sinal estendido é igual a $2N$.

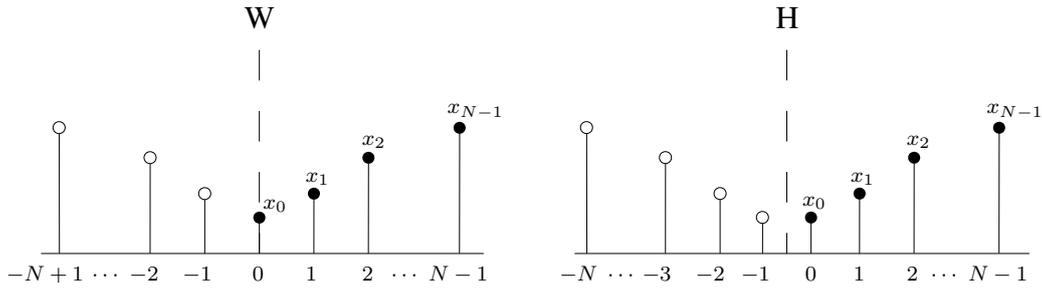


Figura 2.1: Extensões W e H para um sinal $\mathbf{x} \in \mathbb{R}^N$.

Usando a extensão H, considere-se a extensão simétrica para um sinal \mathbf{x} como apresentado na Figura 2.2 (a). O sinal resultante é denotado como \mathbf{x}^{II} , o qual é referenciado como o sinal estendido tipo II. A expansão de Fourier deste sinal é dado por [75]:

$$y_n^{\text{II}} = \sum_{k=0}^{N-1} e^{i\pi k/2N} x_n^{\text{II}} \cos\left(\frac{k(2n+1)\pi}{2N}\right).$$

O análogo discreto conduz diretamente aos coeficientes da DCT-II de comprimento N :

$$[\mathbf{C}_N^{\text{II}}]_{k,n} = \sqrt{\frac{2}{N}} c_k \cos\left(\frac{k(2n+1)\pi}{2N}\right), \quad k, n = 0, 1, \dots, N-1,$$

em que

$$c_k = \begin{cases} 1/\sqrt{2}, & \text{se } k = 0, \\ 1, & \text{se } k \neq 0. \end{cases}$$

O escalar $\sqrt{\frac{2}{N}}$ é introduzido com a finalidade de que as linhas da DCT-II sejam vetores

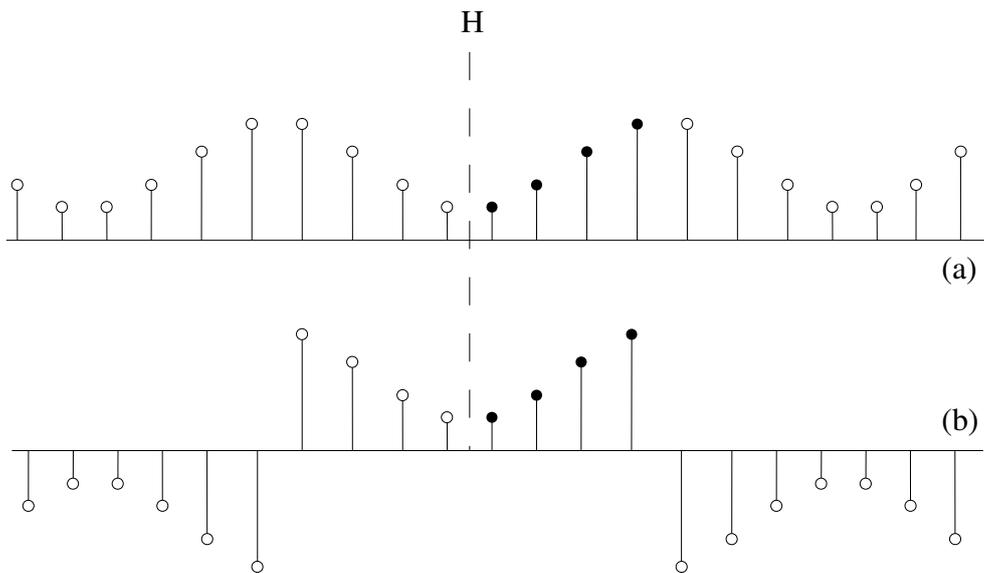


Figura 2.2: (a) Extensão tipo II e (b) extensão tipo IV.

unitários.

Diferentes extensões do sinal x considerado conduzem a diferentes versões da DCT. Por exemplo, a expansão de Fourier do sinal estendido apresentado na Figura 2.2 (b), que é referenciado como o sinal estendido tipo IV, conduz aos coeficientes da DCT-IV de comprimento N :

$$[C_N^{IV}]_{k,n} = \sqrt{\frac{2}{N}} \cos\left(\frac{(2k+1)(2n+1)\pi}{4N}\right), \quad k,n = 0,1,\dots,N-1.$$

Formalmente, os quatro tipos de DCT são definidas como as transformadas cujas

(k,n) -ésimas entradas são dadas por [15, 72, 84]:

$$\begin{aligned}
[\mathbf{C}_{N+1}^I]_{k,n} &= \sqrt{\frac{2}{N}} c_k c_n \cos\left(\frac{kn\pi}{N}\right), \quad k,n = 0,1,\dots,N, \\
[\mathbf{C}_N^{II}]_{k,n} &= \sqrt{\frac{2}{N}} c_k \cos\left(\frac{k(2n+1)\pi}{2N}\right), \quad k,n = 0,1,\dots,N-1, \\
[\mathbf{C}_N^{III}]_{k,n} &= \sqrt{\frac{2}{N}} c_n \cos\left(\frac{(2k+1)n\pi}{2N}\right), \quad k,n = 0,1,\dots,N-1, \\
[\mathbf{C}_N^{IV}]_{k,n} &= \sqrt{\frac{2}{N}} \cos\left(\frac{(2k+1)(2n+1)\pi}{4N}\right), \quad k,n = 0,1,\dots,N-1,
\end{aligned} \tag{2.9}$$

em que

$$c_j = \begin{cases} 1/\sqrt{2}, & \text{se } j = 0 \text{ ou } j = N, \\ 1, & \text{se } j \neq 0 \text{ e } j \neq N, \end{cases}$$

para $j = k$ ou $j = n$. Nas definições dos diferentes tipos de DCT, os subíndices indicam a ordem da matriz e o supraíndice o tipo de transformada.

Estas transformadas satisfazem as seguintes relações:

$$\begin{aligned}
(\mathbf{C}_{N+1}^I)^{-1} &= (\mathbf{C}_{N+1}^I)^\top = \mathbf{C}_{N+1}^I, \\
(\mathbf{C}_N^{II})^{-1} &= (\mathbf{C}_N^{II})^\top = \mathbf{C}_N^{III}, \\
(\mathbf{C}_N^{IV})^{-1} &= (\mathbf{C}_N^{IV})^\top = \mathbf{C}_N^{IV}.
\end{aligned}$$

Portanto, elas são ortogonais. Além disso, as transformadas DCT-I e DCT-IV são simétricas.

Como outra alternativa, Strang [72] obtém os diferentes tipos de DCT como soluções do oscilador harmônico simples, considerando diferentes condições de contorno. Com efeito, seja a equação diferencial:

$$-\mathbf{u}'' = \lambda \cdot \mathbf{u}, \tag{2.10}$$

em que \mathbf{u}'' representa a segunda derivada de \mathbf{u} com respeito de sua variável independente. Se $\mathbf{u} = [u_0 \ u_1 \ \dots \ u_{N-1}]^\top \in \mathbb{R}^N$, então a j -ésima componente de \mathbf{u}'' ,

Tabela 2.1: Tipos de DCT

Descrição	Condições de contorno	Autovetor \mathbf{u}_k , $k = 0, 1, \dots, N-1$
DCT-I	$f_1 = -f_2 = 2$ $-g_1 = g_2 = 2$	$\frac{1}{\sqrt{N}} \cos\left(\frac{kn\pi}{N-1}\right)$, $n = 0$ ou $n = N-1$, $\sqrt{\frac{2}{N}} \cos\left(\frac{kn\pi}{N-1}\right)$, $n = 1, \dots, N-2$.
DCT-II	$f_1 = -f_2 = 1$ $-g_1 = g_2 = 1$	$\frac{1}{\sqrt{N}} \cos\left(\frac{k(2n+1)\pi}{2N}\right)$, $n = 0$, $\sqrt{\frac{2}{N}} \cos\left(\frac{k(2n+1)\pi}{2N}\right)$, $n = 1, \dots, N-1$.
DCT-III	$f_1 = -f_2 = 2$ $g_1 = -1, g_2 = 2$	$\frac{1}{\sqrt{N}} \cos\left(\frac{(2k+1)n\pi}{2N}\right)$, $n = 0$, $\sqrt{\frac{2}{N}} \cos\left(\frac{(2k+1)n\pi}{2N}\right)$, $n = 1, \dots, N-1$.
DCT-IV	$f_1 = -f_2 = 1$ $g_1 = -1, g_2 = 3$	$\sqrt{\frac{2}{N}} \cos\left(\frac{(2k+1)(2n+1)\pi}{4N}\right)$, $n = 0, 1, \dots, N-1$.

2.5 Algoritmos rápidos para a DCT

Após a introdução da DCT por Ahmed *et al.* [2], o método convencional de implementá-la era através de uma transformada rápida de Fourier. Contudo, este método de implementação requer aritmética complexa. Dado que atualmente a DCT tem ampla utilização no contexto de processamento de imagens, algoritmos rápidos mais eficientes foram desenvolvidos para sua implementação. Alguns dos mais conhecidos são:

- Chen *et al.* (1977) [18];
- Lee (1984) [53];
- Wang (1984) [84];
- Vetterli e Nussbaumer (1984) [80];
- Suehiro e Hatori (1986) [76];
- Hou (1987) [40];
- Arai *et al.* (1988) [4];

- Loeffler *et al.* (1989) [56];
- Feig e Winograd (1992) [26];
- Yuan *et al.* (2006) [91].

Estes algoritmos usam aritmética real e baseiam-se em fatorações matriciais com a finalidade de reduzir o número de multiplicações necessárias para implementar a DCT.

Em particular, neste trabalho fazemos uso das fatorações de Chen *et al.* e Feig-Winograd e, portanto, as mesmas são introduzidas em detalhe nos Capítulos 4 e 5 respectivamente. Além de implementar a DCT eficientemente, estes algoritmos são usados em outros contextos. Por exemplo, a fatoração de Feig-Winograd é usada para obter um novo algoritmo DCT aproximado baseado em inteiros algébricos [22] e um *pruning method* para compressão de vídeo [70]. Além disso, a fatoração de Chen *et al.* tem aplicação na implementação de aproximações rápidas para a DCT baseadas em *lifting scheme* [55] e na obtenção de codecs para compressão de imagem/vídeo de baixo custo multiplicativo [50].

Na Tabela 2.2, é apresentado o custo aritmético para implementar a DCT de comprimento 8 de cada algoritmo considerado. Neste caso, a complexidade é dada essencialmente por o número de multiplicações e adições necessárias para implementar o algoritmo. Contudo, como é conhecido no contexto da aritmética de circuitos digitais, a operação de multiplicação tem maior custo de implementação que a operação de adição. Embora alguns algoritmos apresentem pequenas diferenças de multiplicações ou adições, o ganho em custo aritmético se faz evidente em aplicações que por sua natureza requerem uma grande quantidade de implementações desses algoritmos, por exemplo, no caso de compressão de imagens de satélites ou em compressão de vídeos que possuem um grande número de quadros a serem processados.

Como deduzido em [27], o limite inferior teórico do número de multiplicações necessárias para implementar a DCT de comprimento 8 é igual a 11. Assim, o algoritmo de Loeffler atinge o limite inferior teórico e, portanto, pode ser considerado como o algoritmo mais eficiente nestes termos, embora seja possível reduzir ainda mais o número de coeficientes da DCT a serem calculados, quantizados e codificados, usando uma técnica conhecida como *scaled DCT*, como acontece com o algoritmo de Arai *et al.* [4].

Tabela 2.2: *Custo aritmético dos algoritmos para a DCT*

	Multiplicações	Adições
<i>Arai et al. (scaled DCT)</i>	$5 + 8$	29
<i>Loeffler et al.</i>	11	29
Lee	12	29
Vetterli e Nussbaumer	12	29
Hou	12	29
Yuan <i>et al.</i>	12	29
Wang	13	29
Suehiro e Hatori	13	29
Chen <i>et al.</i>	16	26
Feig e Winograd	22	28

Capítulo 3

Aproximações da DCT

3.1 Introdução

No Capítulo 2, foram mencionados alguns dos algoritmos rápidos mais usados para implementar a DCT. Embora estes algoritmos permitam implementar a DCT com um menor custo aritmético, eles ainda requerem aritmética de ponto flutuante, o que demanda maiores exigências de hardware. Assim, uma alternativa possível é considerar aproximações que, embora não implementem a DCT exatamente, permitam obter aproximações que operem em aritmética inteira e não precisem de multiplicações para as suas implementações.

Neste capítulo, é feita uma revisão destes métodos aproximados. Na primeira seção é descrita, em forma geral, a representação numérica empregada nestes métodos. Nas seções posteriores são apresentadas as aproximações para a DCT livres de multiplicações mais conhecidas. Estas aproximações requerem de uma quantidade reduzida de adições e deslocamentos para as suas implementações e, portanto, podem ser empregadas em diversas aplicações que exigem baixa complexidade computacional. Finalmente, na última seção, é apresentado o custo aritmético requerido por estas aproximações.

3.2 Métodos Aproximados para implementar a DCT

Uma característica da DCT é que as entradas que conformam a matriz da transformada são quantidades irracionais. Portanto, na prática não é possível implementar a DCT com precisão numérica exata [81]. Assim, em aplicações práticas, os algoritmos rápidos mencionados na Seção 2.5 são implementados truncando os coeficientes da DCT com uma precisão satisfatória para a aplicação considerada.

Embora a aplicação destes algoritmos considerando entradas racionais permitam implementar a DCT com um menor custo aritmético, ainda precisam de aritmética de ponto flutuante, que demanda maiores exigências de hardware. Além disso, dado o avanço nesta área, alguns métodos de fatoração para a DCT exata atingem a complexidade mínima teórica [27]. Portanto, não é muito esperado que novos algoritmos proporcionem um ganho considerável no custo aritmético. Assim, métodos aproximados para a DCT oferecem uma alternativa viável para reduzir consideravelmente o custo aritmético. Embora estes métodos não calculem a DCT exatamente, eles proporcionam boas aproximações para a DCT com uma exigência computacional inferior.

Uma possibilidade para aproximar a DCT é considerar o conjunto de racionais diádicos [15, p. 221], que é definido como o conjunto de números racionais da forma:

$$p/2^k, \quad p, k \in \mathbb{N}, \quad p = 1, \dots, 2^k - 1.$$

Estes números podem ser representados na forma:

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

Como é conhecido em implementação de circuitos aritméticos digitais, multiplicar ou dividir uma quantidade por dois é equivalente à aplicação de um deslocamento de bits, o qual requer de um menor custo de implementação que adições e multiplicações. Assim, a multiplicação de uma quantidade x por um racional diádico pode ser implementada somente usando adições e deslocamentos de bits.

Embora a representação usando racionais diádicos possa oferecer boas aproxima-

ções, como as obtidas usando o *lifting scheme* [55], isto implica um incremento na complexidade aditiva e no número de deslocamentos de bits.

Uma representação alternativa para um racional diádico é considerar o sistema numérico conhecido como *canonical signed digital* (CSD). Esta representação é da forma (3.1), usando $c_i = 0$, $c_i = 1$ ou $c_i = -1$, e tem a vantagem de reduzir ainda mais o número de adições necessárias para a sua implementação. Por exemplo, a multiplicação de uma quantidade x pelo racional diádico $125/128$ na forma (3.1) produz:

$$(125/128)x = (1/2)x + (1/2^2)x + (1/2^3)x + (1/2^4)x + (1/2^5)x + (1/2^7)x,$$

enquanto que usando o sistema CSD pode ser representado como:

$$(125/128)x = x - (1/2^5)x + (1/2^7)x,$$

que requer somente de duas adições para a sua implementação.

Uma alternativa mais efetiva desde o ponto de vista computacional, é considerar quantidades que não introduzam adições extras na implementação. Isto é certo para potências de dois. Assim, a multiplicação de uma quantidade x por 2^r , com $r \in \mathbb{Z}$, requer somente r deslocamentos de bits para a sua implementação. Embora esta representação não introduz adições ou multiplicações a mais, pode requerer de uma quantidade significativa de deslocamentos.

Com o objetivo de minimizar o número de somas e deslocamentos, neste trabalho consideramos somente aproximações cujas fatorações envolvem elementos no conjunto $\mathcal{P} = \{0, \pm \frac{1}{2}, \pm 1, \pm 2\}$. Em termos de aritmética digital, tais elementos não introduzem adições a mais. Sobre o conjunto \mathcal{P} , o cenário mais desfavorável é considerar elementos em $\{\pm \frac{1}{2}, \pm 2\}$. Nesse caso, somente incrementa-se uma quantidade reduzida de deslocamentos de bits. Na Figura 3.1 é apresentado um esquema da representação numérica usada na implementação das aproximações da DCT.

Na literatura de processamento de imagens, são propostas diversas aproximações livres de multiplicações para o cômputo eficiente da DCT. Algumas das aproximações mais conhecidas envolvendo somente elementos no conjunto \mathcal{P} são apresentadas nas seguintes seções.

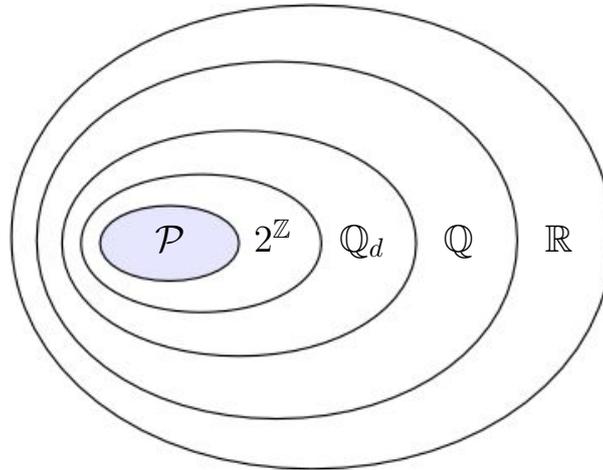


Figura 3.1: Esquema da representação numérica usada na implementação das aproximações da DCT (\mathbb{R} : números reais, \mathbb{Q} : números racionais, \mathbb{Q}_d : racionais diádicos, $2^{\mathbb{Z}}$: potências de 2, $\mathcal{P} = \{0, \pm \frac{1}{2}, \pm 1, \pm 2\}$).

3.2.1 A Transformada de Walsh-Hadamard

A Transformada de Walsh-Hadamard (WHT) de ordem N é dada por uma matriz binária \mathbf{T}_{WHT} de tamanho $N \times N$, com entradas em $\{\pm 1\}$, que satisfaz:

$$\mathbf{T}_{\text{WHT}} \cdot \mathbf{T}_{\text{WHT}}^{\top} = N \cdot \mathbf{I}_N, \quad (3.2)$$

em que \mathbf{I}_N é a matriz identidade $N \times N$.

A WHT de ordem 8 é dada por:

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

Para este caso particular, a transformada tem como funções base as funções de Walsh apresentadas na Figura 3.2 [25, p. 301]. As funções de Walsh formam um conjunto ortogonal definido sobre um intervalo unitário. Assim, de (3.2) vemos que a WHT é uma transformada ortogonal.

A WHT é muito usada em diversas aplicações, tais como codificação, compressão e análise espectral [25, 39, 77]. Embora a WHT de ordem 8 não é definida como

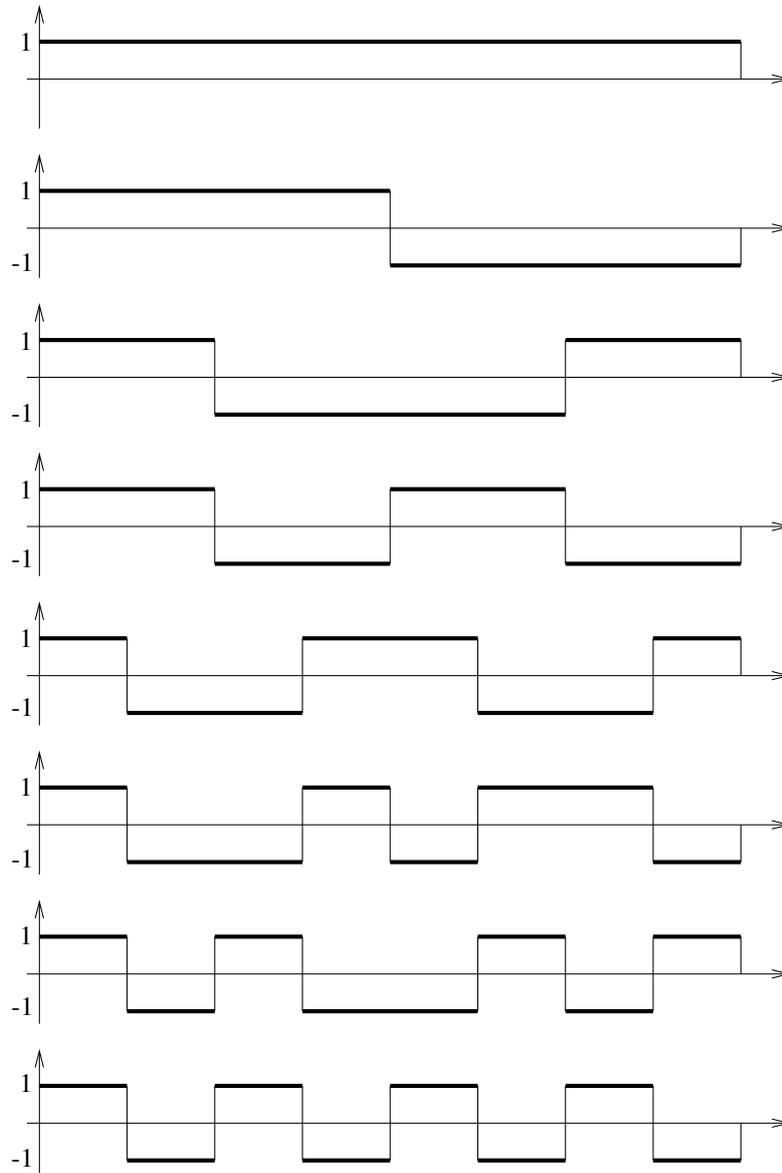


Figura 3.2: Funções de Walsh ($N = 8$).

uma aproximação para a DCT, é muito usada em processamento de imagens pela simplicidade na sua implementação e seu bom desempenho. Além disso, dado que tem entradas em $\{\pm 1\}$, somente requer de adições para a sua implementação, resultando assim em uma transformada de baixa complexidade computacional. Por ser uma transformada muito usada no contexto de compressão de imagens [3, 82], nos Capítulos 4 e 5 comparamos o desempenho da WHT com as novas aproximações para a DCT de comprimento 8 propostas neste trabalho, segundo medidas objetivas de qualidade.

3.2.2 A SDCT

A SDCT, introduzida por Haweel [32], foi a primeira transformada proposta como uma aproximação para a DCT com entradas no conjunto \mathcal{P} . A SDCT é dada por (1.1) e pode ser definida para qualquer comprimento N . Em particular, a SDCT de comprimento 8 é dada por:

$$\mathbf{T}_{\text{SDCT}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}. \quad (3.3)$$

Em [32], a SDCT é avaliada em termos de identificação de sistemas e compressão de imagens, mostrando assim sua similaridade respeito de decorrelação e compactação de energia com a DCT.

Dado que a SDCT é uma transformada binária com entradas em $\{\pm 1\}$, somente requer de adições para a sua implementação. No caso de $N = 8$, a SDCT inversa é dada por:

$$\mathbf{T}_{\text{SDCT}}^{-1} = \frac{1}{8} \cdot \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \\ 1 & 2 & 1 & 0 & -1 & -2 & -1 & 0 \\ 1 & 0 & -1 & -2 & -1 & 0 & 1 & 2 \\ 1 & 0 & -1 & 0 & 1 & 2 & -1 & -2 \\ 1 & 0 & -1 & 0 & 1 & -2 & -1 & 2 \\ 1 & 0 & -1 & 2 & -1 & 0 & 1 & -2 \\ 1 & -2 & 1 & 0 & -1 & 2 & -1 & 0 \\ 1 & -2 & 1 & -2 & 1 & 0 & 1 & 0 \end{bmatrix}. \quad (3.4)$$

Como mencionado nos Capítulos 4 e 5, no contexto de compressão de imagens, o fator $1/8$ na transformada inversa pode ser absorvido na etapa de quantização (cf. Apêndice B). Assim, a multiplicação por este fator não incrementa a complexidade

computacional no processo de compressão. Embora a SDCT não seja ortogonal, sua inversa possui (salvo por um fator de $1/8$) entradas no conjunto \mathcal{P} . Portanto, a transformada inversa mantém a propriedade de baixa complexidade atingida pela transformada direta.

Por ser uma referência clássica, a SDCT é comparada, no contexto de compressão de imagens, com as novas aproximações para a DCT apresentadas no Capítulo 5.

3.2.3 Aproximação de nível 1 de Lengwehasatit e Ortega

Em [54], Lengwehasatit e Ortega propuseram cinco níveis de aproximações para a DCT em função das características do sinal de entrada. Em particular, a aproximação de nível 1 é uma transformada ortogonal com entradas no conjunto \mathcal{P} e é dada por:

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

Como discutido no Capítulo 4, esta transformada pode ser obtida como um membro de uma classe particular de transformadas de baixa complexidade. De fato, esta transformada resulta ser ótima segundo medidas objetivas de qualidade, o qual permite explicar seu bom desempenho no contexto de compressão de imagens. Contudo, esta aproximação também é conhecida por seu relativamente alto custo aritmético quando comparada com outras aproximações para a DCT de baixa complexidade (cf. Seção 3.3).

3.2.4 Série de Aproximações BAS

A série de aproximações introduzidas por Bouguezel, Ahmad e Swamy [7, 8, 9, 10, 12, 13], denotada neste trabalho como série BAS, é um conjunto de aproximações para a DCT de baixa complexidade computacional. Como comentado na Seção 1.2, a maioria destas aproximações são obtidas como variações da SDCT. Dado que as entradas destas aproximações pertencem ao conjunto \mathcal{P} , somente requerem de adições e deslocamentos para suas implementações. Na Tabela 3.1, são apresentadas as diferentes

transformadas da série BAS.

Das aproximações BAS, a transformada introduzida em [9] é não-ortogonal. Sua transformada inversa é dada por:

$$\mathbf{T}^{-1} = \frac{1}{4} \cdot \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 & -\frac{1}{2} & -1 & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & \frac{1}{2} & 2 & -\frac{1}{2} & -2 \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 & \frac{1}{2} & -2 & -\frac{1}{2} & 2 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & 1 & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & -1 & \frac{1}{2} & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -1 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}.$$

Como acontece com a SDCT, a transformada inversa possui (salvo por um fator de $1/4$) entradas no conjunto \mathcal{P} , preservando assim a propriedade de baixa complexidade atingida pela transformada direta.

3.2.5 A DCT arredondada

Em [20], Cintra e Bayer introduzem uma aproximação para a DCT de comprimento 8, denotada por RDCT, usando a função de arredondamento (*round-off*), a qual é definida a seguir.

Dado $x \in \mathbb{R}$, seja $\lfloor x \rfloor$ o maior inteiro que não excede x . A função de arredondamento é definida por:

$$\text{round}(x) = \lfloor x + 0.5 \rfloor. \quad (3.5)$$

A função $\text{round}(\cdot)$, quando aplicada sobre matrizes, opera elemento a elemento.

A RDCT é definida por:

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

em que \mathbf{C} denota a DCT de comprimento 8. Assim, esta transformada é dada expli-

tamente por:

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

Dado que as entradas da transformada estão contidas no conjunto $\{0, \pm 1\}$, somente requer de adições para a sua implementação. Como apresentado no Capítulo 4, esta transformada pode ser obtida como um caso particular em uma classe de transformadas aproximadas, sendo considerada ótima segundo medidas de qualidade a serem discutidas.

3.2.6 A DCT arredondada modificada

Em [5], Bayer e Cintra introduzem uma aproximação para a DCT de comprimento 8, a qual possui a distinção de ter a menor complexidade de todas as aproximações para a DCT consideradas neste trabalho. Esta aproximação é obtida como uma modificação da RDCT e é referenciada como a MRDCT. Esta transformada é dada por:

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

Por ter entradas no conjunto $\{0, \pm 1\}$, somente requer de adições para a sua implementação. Conjuntamente com as aproximações de nível 1 de Lengwehasatit e Ortega e a RDCT discutidas nas Seções 3.2.3 e 3.2.5 respectivamente, esta transformada pertence a uma classe de aproximações de baixa complexidade introduzida no Capítulo 4.

3.2.7 Aproximação para *RF imaging*

Em [62], é proposta uma aproximação da DCT para uso em imagens de rádio-frequência (*RF imaging*). Esta aproximação é obtida como solução do seguinte problema de oti-

mização:

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

em que $\epsilon(\cdot)$ é o erro de energia total a ser discutido em (4.16). Neste caso, o mínimo é considerado sobre todas as transformadas ortogonais com formato dado por (4.1) e entradas no conjunto $\{0, \pm 1, \pm 2\}$.

Como resultado do problema de otimização, a seguinte transformada ortogonal de baixa complexidade para *RF imaging* é obtida:

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

Conjuntamente com as transformadas propostas nas Seções 3.2.3, 3.2.5 e 3.2.6, esta transformada pertence a uma classe particular de transformadas de baixa complexidade derivada no Capítulo 4.

3.3 Custo aritmético das Aproximações

Como comentado na Seção 3.2, as aproximações apresentadas nas seções anteriores requerem de uma baixíssima quantidade de adições e deslocamentos para as suas implementações.

Um fato importante para considerar é que, embora a DCT seja uma transformada ortogonal, algumas aproximações consideradas na literatura são não-ortogonais. De fato, das transformadas listadas anteriormente, a SDCT e a BAS introduzida em [9] são não-ortogonais. No contexto de compressão de imagens, isto não representa uma desvantagem se a transformada não-ortogonal tem inversa de baixa complexidade computacional. Na Tabela 3.2, é apresentado o custo aritmético das aproximações listadas (cf. Tabela 2.2). Como mostrado na tabela, a MRDCT possui o menor custo aritmético de todas as aproximações listadas.

Nos próximos dois capítulos são propostas novas aproximações para a DCT. Estas aproximações são obtidas considerando algumas das fatorações para a DCT listadas

Tabela 3.2: *Custo aritmético das aproximações consideradas*

Aproximação	Adições	Deslocamentos
MRDCT [5]	14	0
BAS [13]	16	0
BAS [13]	18	0
BAS [10]	18	0
BAS [8]	18	2
BAS [13]	18	2
BAS [9]	21	0
RDCT [20]	22	0
WHT [25, p. 301]	24	0
SDCT [32]	24	0
BAS [7]	24	0
Aproximação de nível 1 [54]	24	2
BAS [12]	24	4
Aproximação para <i>RF imaging</i> [62]	24	6

na Seção 2.5. Dado que os algoritmos rápidos para estas aproximações contêm entradas somente no conjunto \mathcal{P} , as mesmas são livres de multiplicações e precisam de uma quantidade reduzida de adições e deslocamentos, convertendo-as em boas aproximações para aplicações que precisam de uma baixa complexidade computacional nas suas implementações.

Uma Classe de Aproximações para a DCT Baseadas na Fatoração de Feig-Winograd

4.1 Introdução

O desenvolvimento deste capítulo pode ser dividido em duas etapas: primeiro, consideramos uma parametrização da fatoração de Feig-Winograd [26] para a DCT de comprimento 8, com o objetivo de introduzir uma nova classe de aproximações. Isto produz um subespaço matricial, onde cada transformada neste subespaço induz uma aproximação para a DCT e, além disso, o algoritmo de Feig-Winograd fornece, de fato, uma fatoração matricial para cada aproximação.

Esta nova classe de aproximações contém muitas transformadas e, portanto, nossa segunda etapa consiste em apresentar um problema de otimização que obtenha as aproximações que forneçam melhor desempenho em termos de compactação de imagens. Mais concisamente, apresentamos um problema de otimização multicritério [24, 59], em que as funções objetivos consideradas neste problema são, além do custo aritmético de cada transformada, medidas de avaliação amplamente conhecidas no contexto de compressão de imagens.

Finalmente, as aproximações obtidas são comparadas, no contexto de compressão de imagens, com outras transformadas conhecidas na literatura que não pertencem

ao formalismo de Feig-Winograd, em termos de PSNR [78] e o índice de similaridade estrutural (SSIM) [87]. As curvas destas medidas proporcionam, para um número dado de coeficientes retidos, valores que permitem comparar quantitativamente a qualidade de compressão das diferentes transformadas.

Os valores de PSNR e SSIM são calculados sobre um conjunto de 45 imagens padronizadas e considerada a média destes valores, proporcionando um resultado estatisticamente mais robusto quando comparadas com imagens individuais. Além disso, são avaliados, qualitativamente, os desempenhos das diferentes transformadas consideradas ao serem aplicadas sobre algumas imagens individuais.

4.2 Fatoração de Feig-Winograd

A DCT de comprimento 8, denotada por \mathbf{C} , é dada por:

$$\mathbf{C} = \frac{1}{2} \cdot \begin{bmatrix} \gamma_3 & \gamma_3 \\ \gamma_0 & \gamma_2 & \gamma_4 & \gamma_6 & -\gamma_6 & -\gamma_4 & -\gamma_2 & -\gamma_0 \\ \gamma_1 & \gamma_5 & -\gamma_5 & -\gamma_1 & -\gamma_1 & -\gamma_5 & \gamma_5 & \gamma_1 \\ \gamma_2 & -\gamma_6 & -\gamma_0 & -\gamma_4 & \gamma_4 & \gamma_0 & \gamma_6 & -\gamma_2 \\ \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 & \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 \\ \gamma_4 & -\gamma_0 & \gamma_6 & \gamma_2 & -\gamma_2 & -\gamma_6 & \gamma_0 & -\gamma_4 \\ \gamma_5 & -\gamma_1 & \gamma_1 & -\gamma_5 & -\gamma_5 & \gamma_1 & -\gamma_1 & \gamma_5 \\ \gamma_6 & -\gamma_4 & \gamma_2 & -\gamma_0 & \gamma_0 & -\gamma_2 & \gamma_4 & -\gamma_6 \end{bmatrix}, \quad (4.1)$$

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

Em [26], Feig e Winograd introduzem um algoritmo rápido para a DCT de comprimento 8 baseada na seguinte fatoração:

$$\mathbf{C} = \frac{1}{2} \cdot \mathbf{P} \cdot \tilde{\mathbf{K}} \cdot \tilde{\mathbf{B}}_1 \cdot \tilde{\mathbf{B}}_2 \cdot \tilde{\mathbf{B}}_3,$$

em que \mathbf{P} é uma matriz de permutação, $\tilde{\mathbf{K}}$ é uma matriz multiplicativa e $\tilde{\mathbf{B}}_1$, $\tilde{\mathbf{B}}_2$ e $\tilde{\mathbf{B}}_3$ são matrizes aditivas. Estas matrizes são dadas por:

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

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

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

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

e

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

Contudo, uma fatoração equivalente mais conveniente é dada por:

$$\mathbf{C} = \frac{1}{2} \cdot \mathbf{P} \cdot \mathbf{K} \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3, \quad (4.2)$$

em que

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

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

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

e

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

Uma das vantagens desta fatoração é que as matrizes \mathbf{B}_1 , \mathbf{B}_2 e \mathbf{B}_3 são simétricas. A complexidade multiplicativa da fatoração anterior é dada pela matriz multiplicativa \mathbf{K} , pois os elementos γ_k , $k = 0, 1, \dots, 6$, são as únicas quantidades não triviais no algoritmo de Feig-Winograd. Como se pode observar de (4.2), a implementação da DCT usando a fatoração de Feig-Winograd requer de 28 adições e 22 multiplicações.

4.3 O Subespaço de Feig-Winograd

A fatoração de Feig-Winograd induz naturalmente o seguinte mapeamento:

$$\begin{aligned} \text{FW} : \mathbb{R}^7 &\longrightarrow \mathcal{M}_8(\mathbb{R}) \\ \alpha &\longrightarrow \mathbf{P} \cdot \mathbf{K}(\alpha) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \end{aligned} \quad (4.3)$$

em que $\alpha = [\alpha_0 \ \alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ \alpha_5 \ \alpha_6]^\top \in \mathbb{R}^7$,

$$\mathbf{K}(\alpha) = \begin{bmatrix} \alpha_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_5 & \alpha_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha_1 & \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha_6 & -\alpha_4 & -\alpha_2 & -\alpha_0 \\ 0 & 0 & 0 & 0 & \alpha_4 & \alpha_0 & \alpha_6 & -\alpha_2 \\ 0 & 0 & 0 & 0 & -\alpha_0 & \alpha_2 & -\alpha_4 & \alpha_6 \\ 0 & 0 & 0 & 0 & -\alpha_2 & -\alpha_6 & \alpha_0 & -\alpha_4 \end{bmatrix}, \quad (4.4)$$

e $\mathcal{M}_8(\mathbb{R})$ denota o espaço das matrizes de tamanho 8×8 com elementos em \mathbb{R} . Usando as operações de adição e multiplicação escalar em $\mathcal{M}_8(\mathbb{R})$, é direto observar que o mapeamento $\text{FW}(\cdot)$ é uma transformação linear. Com efeito, de (4.3) e (4.4) temos que se satisfazem as seguintes propriedades:

1. *Aditividade*: Sejam $\alpha, \alpha' \in \mathbb{R}^7$. Então se satisfaz

$$\begin{aligned} \text{FW}(\alpha + \alpha') &= \mathbf{P} \cdot \mathbf{K}(\alpha + \alpha') \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= \mathbf{P} \cdot (\mathbf{K}(\alpha) + \mathbf{K}(\alpha')) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= \mathbf{P} \cdot \mathbf{K}(\alpha) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 + \mathbf{P} \cdot \mathbf{K}(\alpha') \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= \text{FW}(\alpha) + \text{FW}(\alpha'). \end{aligned}$$

2. *Multiplicação escalar*: Se $c \in \mathbb{R}$, então

$$\begin{aligned} \text{FW}(c \cdot \alpha) &= \mathbf{P} \cdot \mathbf{K}(c \cdot \alpha) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= \mathbf{P} \cdot (c \cdot \mathbf{K}(\alpha)) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= c \cdot \mathbf{P} \cdot \mathbf{K}(\alpha) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= c \cdot \text{FW}(\alpha). \end{aligned}$$

Como consequência, a imagem de $\text{FW}(\cdot)$, que denotamos por $\text{Im}(\text{FW})$, é um subespaço de $\mathcal{M}_8(\mathbb{R})$ que chamamos de subespaço de Feig-Winograd. Nosso objetivo é então identificar matrizes no subespaço de Feig-Winograd que proporcionem boas aproximações para a DCT exata segundo critérios a serem definidos mais adiante. Por brevidade, de aqui em diante será usada a notação $\mathbf{T} = \text{FW}(\alpha)$.

Considerando o mapeamento descrito em (4.3), vemos que as matrizes no subespaço de Feig-Winograd compartilham a mesma fatoração. Portanto, todas as matrizes

neste subespaço estão associadas a algoritmos rápidos com mesma estrutura. A Figura 4.1 exibe o diagrama de fluxo de sinal para o algoritmo rápido em questão.

4.4 Transformada Inversa

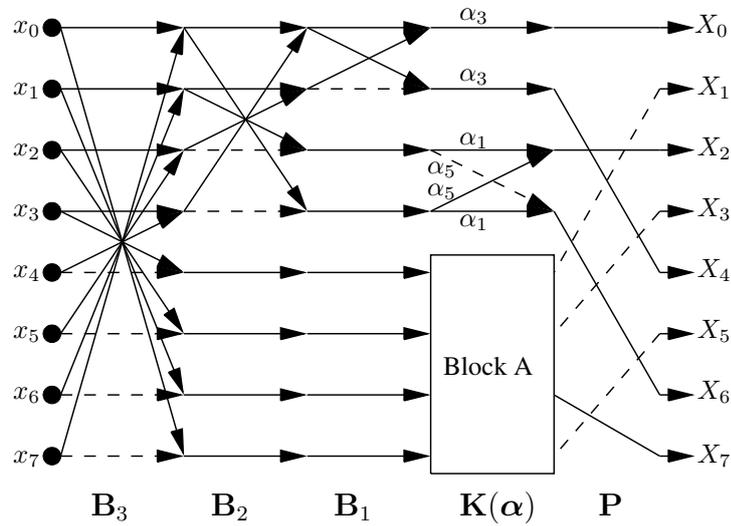
Nesta seção, deduzimos alguns resultados teóricos relacionando as estruturas das transformadas direta e inversa. Como resultado, obtemos que, se a transformada \mathbf{T} pertence à classe de Feig-Winograd, então a transposta da transformada inversa, $(\mathbf{T}^{-1})^\top$, também encontra-se nesta classe e, portanto, possui uma estrutura similar à transformada direta.

No diagrama de fluxo da Figura 4.1, as matrizes \mathbf{B}_1 , \mathbf{B}_2 e \mathbf{B}_3 representam estruturas “borboleta”. A matriz \mathbf{P}^\top é uma permutação simples, com a segunda, quarta e sexta colunas multiplicadas por -1 . Dado que $\mathbf{K}(\boldsymbol{\alpha})$ é uma matriz diagonal em blocos, sua inversa $\mathbf{K}^{-1}(\boldsymbol{\alpha})$ também resulta ser diagonal em blocos. De fato, usando software de computação simbólica [58], obtemos a seguinte relação:

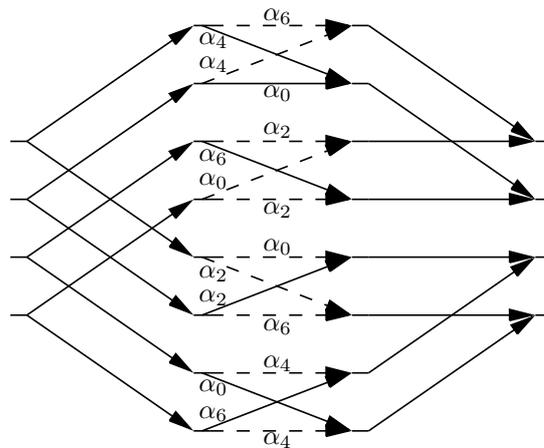
$$\mathbf{K}^{-1}(\boldsymbol{\alpha}) = \mathbf{K}^\top(\boldsymbol{\alpha}'), \quad (4.5)$$

em que o vetor $\boldsymbol{\alpha}' = [\alpha'_0 \ \alpha'_1 \ \alpha'_2 \ \alpha'_3 \ \alpha'_4 \ \alpha'_5 \ \alpha'_6]^\top \in \mathbb{R}^7$, relacionando as matrizes inversa e transposta, é dado por:

$$\begin{aligned} \alpha'_0 &= \frac{\alpha_0 \alpha_6^2 + (\alpha_2^2 - \alpha_4^2) \alpha_6 + 2 \alpha_0 \alpha_2 \alpha_4 + \alpha_0^3}{\lambda}, \\ \alpha'_1 &= \frac{\alpha_1}{\alpha_1^2 + \alpha_5^2}, \\ \alpha'_2 &= \frac{\alpha_2 \alpha_4^2 + (\alpha_0^2 - \alpha_6^2) \alpha_4 + 2 \alpha_0 \alpha_2 \alpha_6 + \alpha_2^3}{\lambda}, \\ \alpha'_3 &= \frac{1}{\alpha_3}, \\ \alpha'_4 &= \frac{\alpha_4 \alpha_2^2 + (\alpha_0^2 - \alpha_6^2) \alpha_2 - 2 \alpha_0 \alpha_4 \alpha_6 + \alpha_4^3}{\lambda}, \\ \alpha'_5 &= \frac{\alpha_5}{\alpha_1^2 + \alpha_5^2}, \\ \alpha'_6 &= \frac{\alpha_6 \alpha_0^2 + (\alpha_2^2 - \alpha_4^2) \alpha_0 - 2 \alpha_2 \alpha_4 \alpha_6 + \alpha_6^3}{\lambda}, \end{aligned} \quad (4.6)$$



(a) Diagrama de fluxo de sinal para T



(b) Block A

Figura 4.1: Diagrama de fluxo de sinal para as transformadas no subespaço de Feig-Winograd. Entrada de dados $x_n, n = 0,1, \dots, 7$, relacionadas à saída $X_m, m = 0,1, \dots, 7$, por $\mathbf{X} = \mathbf{T} \cdot \mathbf{x}$. As linhas tracejadas indicam multiplicação por -1 .

em que

$$\lambda = (\alpha_0^2 + \alpha_6^2)^2 + (\alpha_2^2 + \alpha_4^2)^2 + 4(\alpha_0 \alpha_2 - \alpha_4 \alpha_6)(\alpha_2 \alpha_6 + \alpha_0 \alpha_4).$$

O seguinte lema resulta útil na obtenção de um algoritmo rápido para a transformada inversa.

Lema 1 *Seja $\mathbf{D} = \text{diag}(d_0, d_1, d_2, d_1, d_0, d_1, d_2, d_1)$, para valores reais de d_k , $k = 0, 1, 2$. Então \mathbf{T} pertence ao subespaço de Feig-Winograd se e somente se $\mathbf{D} \cdot \mathbf{T}$ também pertence ao mesmo subespaço.*

Prova: Primeiro provamos a condição de suficiência. Por hipótese, \mathbf{T} admite a fatoração de Feig-Winograd (4.3). Portanto temos que:

$$\begin{aligned} \mathbf{D} \cdot \mathbf{T} &= \mathbf{D} \cdot \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\alpha}) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= \mathbf{P} \cdot \mathbf{D}' \cdot \mathbf{K}(\boldsymbol{\alpha}) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3, \end{aligned}$$

em que $\mathbf{D}' = \text{diag}(d_0, d_0, d_2, d_2, d_1, d_1, d_1, d_1)$. Contudo, se satisfaz que

$$\mathbf{D}' \cdot \mathbf{K}(\boldsymbol{\alpha}) = \begin{bmatrix} d_0 \cdot \alpha_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & d_0 \cdot \alpha_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_2 \cdot \alpha_1 & d_2 \cdot \alpha_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_2 \cdot \alpha_5 & -d_2 \cdot \alpha_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_1 \cdot \alpha_6 & d_1 \cdot \alpha_4 & d_1 \cdot \alpha_2 & -d_1 \cdot \alpha_0 \\ 0 & 0 & 0 & 0 & -d_1 \cdot \alpha_4 & -d_1 \cdot \alpha_0 & -d_1 \cdot \alpha_6 & -d_1 \cdot \alpha_2 \\ 0 & 0 & 0 & 0 & d_1 \cdot \alpha_0 & -d_1 \cdot \alpha_2 & d_1 \cdot \alpha_4 & d_1 \cdot \alpha_6 \\ 0 & 0 & 0 & 0 & d_1 \cdot \alpha_2 & d_1 \cdot \alpha_6 & -d_1 \cdot \alpha_0 & -d_1 \cdot \alpha_4 \end{bmatrix}$$

Dado que $\mathbf{D}' \cdot \mathbf{K}(\boldsymbol{\alpha})$ possui o formato (4.4), concluímos que $\mathbf{D} \cdot \mathbf{T}$ também pertence ao subespaço de Feig-Winograd.

Reciprocamente, sejam \mathbf{T} e \mathbf{T}' matrizes no subespaço de Feig-Winograd associadas aos vetores de parâmetros $\boldsymbol{\alpha}$ e $\boldsymbol{\beta}$, respectivamente. Desse modo, obtemos:

$$\begin{aligned} \mathbf{T} &= \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\alpha}) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3, \\ \mathbf{T}' &= \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\beta}) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3. \end{aligned}$$

Portanto, se $\mathbf{T}' = \mathbf{D} \cdot \mathbf{T}$ para uma matriz \mathbf{D} diagonal, considerando (4.5) temos que:

$$\begin{aligned} \mathbf{D} &= \mathbf{T}' \cdot \mathbf{T}^{-1} \\ &= \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\beta}) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \cdot \mathbf{B}_3^{-1} \cdot \mathbf{B}_2^{-1} \cdot \mathbf{B}_1^{-1} \cdot \mathbf{K}^{-1}(\boldsymbol{\alpha}) \cdot \mathbf{P}^{-1} \\ &= \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\beta}) \cdot \mathbf{K}^{-1}(\boldsymbol{\alpha}) \cdot \mathbf{P}^\top \\ &= \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\beta}) \cdot \mathbf{K}^\top(\boldsymbol{\alpha}') \cdot \mathbf{P}^\top, \end{aligned}$$

em que $\boldsymbol{\alpha}'$ é dado por (4.6).

Calculando explicitamente a matriz diagonal \mathbf{D} , se verifica diretamente que:

$$\mathbf{D} = \text{diag}(d_0, d_1, d_2, d_1, d_0, d_1, d_2, d_1), \quad d_k \in \mathbb{R}, k = 0,1,2.$$

■

Como uma consequência, o algoritmo rápido para a transformada inversa é obtido simplesmente substituindo o vetor $\boldsymbol{\alpha}$ na fatoração (4.3) por $\boldsymbol{\alpha}'$ dado por (4.6) e então aplicar a operação de transposição. Este resultado é enunciado na seguinte proposição.

Proposição 1 *Se \mathbf{T} pertence ao subespaço de Feig-Winograd, então $(\mathbf{T}^{-1})^\top$ também pertence ao mesmo subespaço.*

Prova: Assumindo a existência da inversa \mathbf{T}^{-1} , invertendo (4.3) obtemos a fatoração:

$$\mathbf{T}^{-1} = \mathbf{B}_3^{-1} \cdot \mathbf{B}_2^{-1} \cdot \mathbf{B}_1^{-1} \cdot \mathbf{K}^{-1}(\boldsymbol{\alpha}) \cdot \mathbf{P}^{-1}.$$

Contudo, as seguintes relações são satisfeitas:

$$\begin{aligned} \mathbf{P}^{-1} &= \mathbf{P}^\top, \quad \mathbf{B}_1^{-1} = \text{diag}\left(\frac{1}{2} \cdot \mathbf{I}_2, \mathbf{I}_6\right) \cdot \mathbf{B}_1, \\ \mathbf{B}_2^{-1} &= \text{diag}\left(\frac{1}{2} \cdot \mathbf{I}_4, \mathbf{I}_4\right) \cdot \mathbf{B}_2, \quad \mathbf{B}_3^{-1} = \frac{1}{2} \cdot \mathbf{B}_3, \end{aligned}$$

em que $\text{diag}(\cdot)$ retorna uma matriz diagonal. Portanto, a fatoração para a inversa resulta em:

$$\mathbf{T}^{-1} = \mathbf{B}_3 \cdot \mathbf{B}_2 \cdot \mathbf{B}_1 \cdot \mathbf{K}^{-1}(\boldsymbol{\alpha}) \cdot \mathbf{P}^\top \cdot \mathbf{D}_0, \quad (4.7)$$

em que $\mathbf{D}_0 = \text{diag} \left(\frac{1}{8}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2}, \frac{1}{8}, \frac{1}{2}, \frac{1}{4}, \frac{1}{2} \right)$. Aplicando (4.5) em (4.7), obtemos diretamente que:

$$\begin{aligned} (\mathbf{T}^{-1})^\top &= \mathbf{D}_0 \cdot \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\alpha}') \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3 \\ &= \mathbf{D}_0 \cdot \mathbf{T}', \end{aligned}$$

em que

$$\mathbf{T}' = \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\alpha}') \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3$$

pertence ao subespaço de Feig-Winograd. Portanto, sob as condições do Lema 1, a proposição é provada. ■

O seguinte corolário enuncia condições para a existência da inversa $\mathbf{K}^{-1}(\boldsymbol{\alpha})$, que implica diretamente a existência de \mathbf{T}^{-1} .

Corolário 1 *Se \mathbf{T} é uma transformada no subespaço de Feig-Winograd, as seguintes condições são necessárias para que \mathbf{T} tenha inversa:*

- (i) $\alpha_3 \neq 0$,
- (ii) $\alpha_1^2 + \alpha_5^2 \neq 0$,
- (iii) $\alpha_0^2 + \alpha_2^2 + \alpha_4^2 + \alpha_6^2 \neq 0$.

Prova: De (4.6), é direto observar que as condições necessárias e suficientes para a existência de $\boldsymbol{\alpha}'$ são $\alpha_3 \neq 0$, $\alpha_1^2 + \alpha_5^2 \neq 0$ e $\lambda \neq 0$. Além disso, uma condição necessária para que $\lambda \neq 0$ é que $|\alpha_i| \neq 0$ para algum índice i ($i = 0, 2, 4, 6$), o qual é equivalente à condição $\alpha_0^2 + \alpha_2^2 + \alpha_4^2 + \alpha_6^2 \neq 0$. ■

Diagrama de fluxo de sinal da transformada inversa

Da seção anterior temos que, se $\mathbf{T} \in \mathcal{I}m(\text{FW})$, então sua transformada inversa possui uma estrutura simples, a qual é obtida ao considerar a transposta da transformada $(\mathbf{T}^{-1})^\top \in \mathcal{I}m(\text{FW})$. Embora este resultado seja evidente para transformadas ortogonais, não é evidente para transformadas que são não-ortogonais. Portanto, os resultados obtidos na seção anterior proporcionam uma generalização importante que é útil em implementações de hardware.

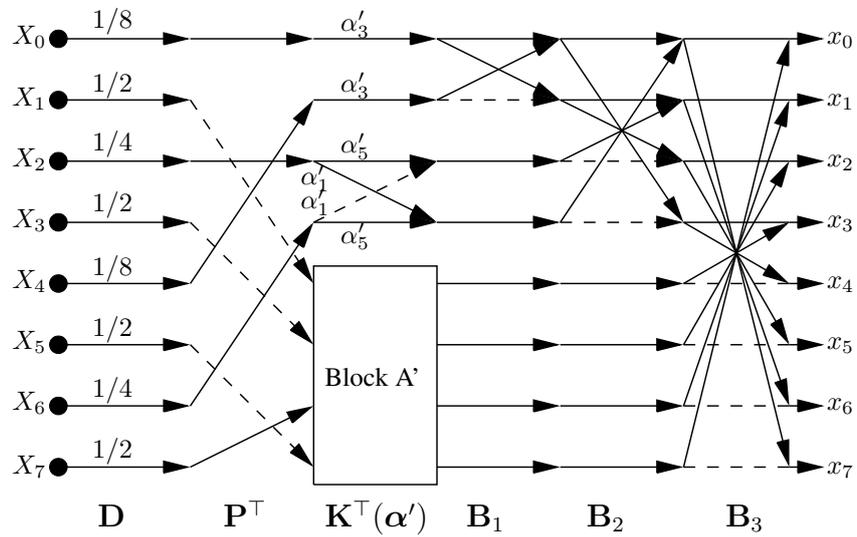
Na Figura 4.2 é mostrado o algoritmo rápido para a transformada inversa (cf. Figura 4.1). Como pode-se observar, em termos de desenho de circuitos, os blocos \mathbf{B}_1 , \mathbf{B}_2 e \mathbf{B}_3 podem ser reutilizados na implementação da transformada inversa. O bloco \mathbf{P}^\top representa uma permutação simples. Além disso, no contexto de compressão de imagens, o bloco \mathbf{D} pode ser absorvido na etapa de quantização (cf. Apêndice B). Portanto, a adaptação do diagrama de fluxo de sinal da transformada direta para a transformada inversa se reduz principalmente à adaptação de $\mathbf{K}^\top(\alpha')$ considerando o vetor de parâmetros α' dado por (4.6). Assim, em termos de circuitos digitais, é possível implementar as transformadas direta e inversa com modificações mínimas de hardware.

4.5 Aproximações Conhecidas

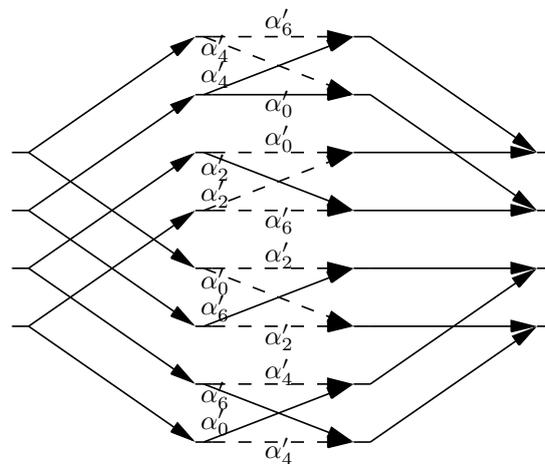
Como comentado anteriormente, cada vetor $\alpha \in \mathbb{R}^7$ fornece uma aproximação para a DCT exata dada pela fatoração (4.3). Nesta seção, apresentamos algumas transformadas já conhecidas na literatura que pertencem ao subespaço de Feig-Winograd:

1. *SDCT*: Para $\alpha_1 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^\top$, temos que $\text{FW}(\alpha_1)$ fornece a transformada SDCT definida em [32].
2. *Aproximação de nível 1*: Para $\alpha_2 = [1 \ 1 \ 1 \ 1 \ 1 \ \frac{1}{2} \ 0]^\top$, temos que $\text{FW}(\alpha_2)$ é a aproximação de nível 1 definida em [54].
3. *DCT arredondada*: Para $\alpha_3 = [1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]^\top$, a transformada $\text{FW}(\alpha_3)$ resulta na aproximação RDCT introduzida em [20].
4. *DCT arredondada modificada*: Para $\alpha_4 = [1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]^\top$, $\text{FW}(\alpha_4)$ é a aproximação MRDCT dada em [5].
5. *Aproximação para RF imaging*: Para $\alpha_5 = [2 \ 2 \ 1 \ 1 \ 1 \ 1 \ 0]^\top$, a transformada $\text{FW}(\alpha_5)$ resulta na aproximação dada em [62].
6. *Matriz de Haar*: Seja \mathbf{H} a matriz de Haar não normalizada de comprimento 8 [47, p. 159]. Para $\alpha_6 = [0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0]^\top$, temos que

$$\mathbf{H} = \mathbf{P}_1 \cdot \text{FW}(\alpha_6) \cdot \mathbf{P}_2,$$



(a) Diagrama de fluxo de sinal para T^{-1}



(b) Block A'

Figura 4.2: Diagrama de fluxo de sinal para as transformadas inversas. Entrada de dados X_n , $n = 0, 1, \dots, 7$, relacionados à saída x_m , $m = 0, 1, \dots, 7$, por $\mathbf{x} = \mathbf{T}^{-1} \cdot \mathbf{X}$. As linhas tracejadas indicam multiplicação por -1 .

em que \mathbf{P}_1 e \mathbf{P}_2 são as seguintes matrizes de permutação:

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

4.6 Critérios de Aproximação

O critério de busca das transformadas \mathbf{T} no subespaço de Feig-Winograd será baseada em condições que permitam obter aproximações adequadas para a DCT. Portanto, neste trabalho adotamos os seguintes critérios:

1. \mathbf{T} deve ter baixa complexidade computacional;
2. \mathbf{T} deve fornecer uma aproximação para a DCT que seja ortogonal ou quasi-ortogonal [28];
3. \mathbf{T}^{-1} deve ter baixa complexidade computacional;
4. \mathbf{T} deve fornecer uma aproximação para a DCT que possua bom desempenho em termos de medidas de avaliação usadas no contexto de compressão de imagens.

Os quatro critérios descritos são detalhados nas próximas seções.

4.6.1 Complexidade Computacional

A complexidade computacional da estrutura Feig-Winograd é essencialmente quantificada por sua complexidade aritmética, dada pelo número de multiplicações, adições e deslocamentos necessários para sua implementação [14, 33]. A quantidade de multiplicações requeridas é dada pelos elementos α_k , $k = 0, \dots, 6$, presentes apenas na matriz $\mathbf{K}(\alpha)$. Contudo, selecionando criteriosamente os valores α_k , é possível obter aproximações que sejam livres de multiplicações. Como mencionado na Seção 3.2, com o objetivo de minimizar o custo aritmético, consideramos $\alpha_k \in \mathcal{P} = \{0, \pm \frac{1}{2}, \pm 1, \pm 2\}$. Em termos de circuitos aritméticos digitais, tais elementos introduzem somente uma

quantidade reduzida de deslocamentos. Isto proporciona uma complexidade consideravelmente menor que implementando diretamente a DCT exata usando a fatoração de Feig-Winograd ou aproximando a DCT usando racionais diádicos.

Sobre o conjunto \mathcal{P} , o cenário mais desfavorável em termos de complexidade computacional é selecionar parâmetros no conjunto $\{\pm\frac{1}{2}, \pm 2\}$. Isso implicaria considerar, no máximo, 28 adições e 22 deslocamentos. Considerando as transformadas no formalismo Feig-Winograd já obtidas, a complexidade esperada para boas aproximações pode ser inferior. Por exemplo, a SDCT e a RDCT —ambas no subespaço de Feig-Winograd— requerem 24/0 e 22/0 adições/deslocamentos, respectivamente [20, 32]. Na Tabela 4.1, é apresentada a complexidade aritmética do algoritmo Feig-Winograd segundo a representação numérica usada.

Tabela 4.1: Complexidade aritmética do algoritmo Feig-Winograd segundo a representação numérica usada

Representação numérica	Multiplicações	Adições	Deslocamentos
Ponto flutuante	22	28	0
Racionais diádicos	0	$28 + 14 = 42$	21
Elementos no conjunto \mathcal{P}	0	no máximo 28	no máximo 22

Inspecionando a fatoração (4.3), é possível obter expressões matemáticas para o cálculo do número de adições e deslocamentos necessários para implementar as aproximações. Dado o vetor $\boldsymbol{\alpha} = [\alpha_0 \ \alpha_1 \ \alpha_2 \ \alpha_3 \ \alpha_4 \ \alpha_5 \ \alpha_6]^\top \in \mathcal{P}^7$, sejam as funções:

$$\theta(\alpha_i) = \begin{cases} 1, & \text{se } \alpha_i = 0, \\ 0, & \text{c.c.} \end{cases} \quad \text{e} \quad \phi(\alpha_i) = \begin{cases} 1, & \text{se } |\alpha_i| = \frac{1}{2} \text{ ou } |\alpha_i| = 2, \\ 0, & \text{c.c.} \end{cases}$$

para $i = 0, 1, \dots, 6$. Então, o número de adições $\mathcal{A}(\boldsymbol{\alpha})$ e deslocamentos $\mathcal{S}(\boldsymbol{\alpha})$ necessários para implementar a transformada \mathbf{T} é dada pelas equações:

$$\mathcal{A}(\boldsymbol{\alpha}) = 28 - \sum_i k_i \cdot \theta(\alpha_i), \quad (4.8)$$

$$\mathcal{S}(\boldsymbol{\alpha}) = \sum_i k_i \cdot \phi(\alpha_i), \quad (4.9)$$

em que

$$k_i = \begin{cases} 4, & \text{se } i \text{ for par,} \\ 2, & \text{se } i \text{ for ímpar.} \end{cases}$$

Das expressões anteriores temos que $14 \leq \mathcal{A}(\alpha) \leq 28$ e $0 \leq \mathcal{S}(\alpha) \leq 22$. Portanto, o limite inferior para a complexidade aritmética das matrizes no subespaço de Feig-Winograd é de 14 adições. Observar da Tabela 4.1 que o custo aritmético das aproximações considerando entradas somente no conjunto \mathcal{P} é muito menor que implementando diretamente a DCT exata usando o algoritmo de Feig-Winograd ou considerando racionais diádicos. Em consequência, estas aproximações são boas alternativas em implementações que requerem de uma complexidade computacional mínima.

4.6.2 Ortogonalidade e Quasi-ortogonalidade

Neste trabalho, adotamos a terminologia definida a continuação respeito de ortogonalidade.

Definição 1 Dizemos que uma matriz \mathbf{A} é ortogonal se $\mathbf{A} \cdot \mathbf{A}^\top$ é uma matriz diagonal. Em particular, se $\mathbf{A} \cdot \mathbf{A}^\top$ é a matriz identidade, então \mathbf{A} é dita ser ortonormal.

Ortogonalidade é frequentemente uma propriedade desejável em aproximações para a DCT [15, 54]. Entre vários métodos de ortogonalização existentes [36, 89], selecionamos o método baseado na decomposição polar [34, 38]. Por este método, o processo para ortogonalizar a transformada \mathbf{T} requer somente da matriz dada em [19]:

$$\mathbf{S} = \sqrt{(\mathbf{T} \cdot \mathbf{T}^\top)^{-1}}, \quad (4.10)$$

em que $\sqrt{\cdot}$ denota a raiz quadrada matricial [35, 58]. A aproximação ortonormal para a DCT é dada por [5, 19, 20, 62]:

$$\hat{\mathbf{C}} = \mathbf{S} \cdot \mathbf{T}. \quad (4.11)$$

Uma característica fundamental é que este método de ortogonalização preserva a estrutura e a baixa complexidade da transformada \mathbf{T} [37].

No contexto da compressão de imagens, se \mathbf{S} é uma matriz diagonal, então não introduz nenhum custo computacional. Neste caso, a matriz \mathbf{S} pode ser absorvida na etapa de quantização no esquema de compressão tipo JPEG [5, 6, 8, 10, 13, 20, 54]. No Apêndice B, são dados os detalhes matemáticos da operação de quantização.

Para que a matriz \mathbf{S} seja diagonal, é suficiente que a transformada \mathbf{T} seja ortogonal [19]. Se a transformada \mathbf{T} pertence ao subespaço de Feig-Winograd, calculando explicitamente $\mathbf{T} \cdot \mathbf{T}^\top$ obtemos:

$$\mathbf{T} \cdot \mathbf{T}^\top = \begin{bmatrix} \tau_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \tau_1 & 0 & -\tau_3 & 0 & \tau_3 & 0 & 0 \\ 0 & 0 & \tau_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\tau_3 & 0 & \tau_1 & 0 & 0 & 0 & \tau_3 \\ 0 & 0 & 0 & 0 & \tau_0 & 0 & 0 & 0 \\ 0 & \tau_3 & 0 & 0 & 0 & \tau_1 & 0 & \tau_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & \tau_2 & 0 \\ 0 & 0 & 0 & \tau_3 & 0 & \tau_3 & 0 & \tau_1 \end{bmatrix},$$

em que

$$\begin{aligned} \tau_0 &= 8 \alpha_3^2, \\ \tau_1 &= 2 (\alpha_0^2 + \alpha_2^2 + \alpha_4^2 + \alpha_6^2), \\ \tau_2 &= 4 (\alpha_1^2 + \alpha_5^2), \\ \tau_3 &= 2 (\alpha_4 \alpha_6 + \alpha_2 \alpha_6 + \alpha_0 \alpha_4 - \alpha_0 \alpha_2). \end{aligned}$$

Portanto, se $\tau_3 = 0$, então \mathbf{T} é uma matriz ortogonal. Isto é equivalente a provar que $\alpha_4 \alpha_6 + \alpha_2 \alpha_6 + \alpha_0 \alpha_4 - \alpha_0 \alpha_2 = 0$, ou que

$$\alpha_0 (\alpha_2 - \alpha_4) = \alpha_6 (\alpha_2 + \alpha_4). \quad (4.12)$$

Assim, (4.12) é uma condição suficiente para que seja satisfeita a condição de ortogonalidade.

Se a transformada \mathbf{T} satisfaz (4.12), então temos que

$$\mathbf{S} = \text{diag}(s_0, s_1, s_2, s_1, s_0, s_1, s_2, s_1),$$

em que $s_0 = 1/(2^{3/2} \alpha_3)$, $s_1 = 1/\sqrt{2 (\alpha_6^2 + \alpha_4^2 + \alpha_2^2 + \alpha_0^2)}$, $s_2 = 1/(2\sqrt{\alpha_5^2 + \alpha_1^2})$ e $\text{diag}(\cdot)$ retorna uma matriz diagonal com entradas desde seu argumento. Como provado na seguinte proposição, se a condição (4.12) é satisfeita, então a aproximação $\hat{\mathbf{C}}$ também pertence ao subespaço de Feig-Winograd.

Proposição 2 *Seja \mathbf{T} no subespaço de Feig-Winograd. Se \mathbf{T} satisfaz (4.12), então $\mathbf{S} \cdot \mathbf{T}$ pertence ao mesmo subespaço.*

Prova: O calculo explícito da matriz de ortogonalização \mathbf{S} (cf. (4.10)) resulta em

$$\mathbf{S} = \text{diag} \left(\frac{1}{\sqrt{\tau_0}}, \frac{1}{\sqrt{\tau_1}}, \frac{1}{\sqrt{\tau_2}}, \frac{1}{\sqrt{\tau_1}}, \frac{1}{\sqrt{\tau_0}}, \frac{1}{\sqrt{\tau_1}}, \frac{1}{\sqrt{\tau_2}}, \frac{1}{\sqrt{\tau_1}} \right).$$

Observar que \mathbf{S} tem o formato da matriz \mathbf{D} do Lema 1. Portanto, se satisfaz que $\mathbf{S} \cdot \mathbf{T}$ pertence ao subespaço de Feig-Winograd. ■

Se a transformada \mathbf{T} não é ortogonal, porém $\mathbf{T} \cdot \mathbf{T}^\top$ é uma matriz quasi-diagonal [28], então \mathbf{S} também é quasi-diagonal. Portanto, neste caso, a aproximação para \mathbf{S} definida na seguinte definição pode ser considerada como matriz de ortogonalização.

Definição 2 *Se $\mathbf{T} \cdot \mathbf{T}^\top$ é uma matriz quasi-diagonal, então uma aproximação para a matriz de ortogonalização \mathbf{S} é dada por*

$$\hat{\mathbf{S}} = \sqrt{[\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)]^{-1}}, \quad (4.13)$$

em que, neste caso, o operador $\text{diag}(\cdot)$ retorna uma matriz diagonal com os mesmos elementos diagonais de sua matriz argumento [69, p. 2].

Em consequência, se a transformada \mathbf{T} não é ortogonal, então a aproximação quasi-ortogonal para a DCT provista por \mathbf{T} é obtida por:

$$\hat{\mathbf{C}} = \hat{\mathbf{S}} \cdot \mathbf{T}. \quad (4.14)$$

Para quantificar quanto próxima uma matriz está na forma diagonal, consideramos a medida de desvio da diagonalidade [28], que é definida como segue.

Definição 3 *Seja \mathbf{M} uma matriz quadrada. A medida de desvio da diagonalidade de \mathbf{M} é dada por:*

$$\delta(\mathbf{M}) = 1 - \frac{\|\text{diag}(\mathbf{M})\|_F^2}{\|\mathbf{M}\|_F^2}, \quad (4.15)$$

em que $\|\cdot\|_F$ denota a norma matricial de Frobenius [89, p. 115].

Observar que para matrizes diagonais, a função $\delta(\cdot)$ retorna zero. Tanto a SDCT como a aproximação BAS proposta em [9] são não-ortogonais e boas aproximações para a DCT. Estas transformadas tem desvio da ortogonalidade $\delta(\mathbf{T} \cdot \mathbf{T}^\top)$ iguais a 0.20 e 0.1774 respectivamente, em que \mathbf{T} denota a transformada não-ortogonal considerada. Portanto, adotamos essas medidas particulares como valores de referência para identificar matrizes de ortogonalização quasi-diagonais no contexto de aproximações para a DCT.

4.6.3 Estrutura e Complexidade da Transformada Inversa

Além de identificar transformadas no subespaço de Feig-Winograd que tenham baixa complexidade computacional, também é importante identificar aquelas cujas transformadas inversas sejam de baixa complexidade.

Para transformadas ortogonais, temos que se satisfaz a relação:

$$\hat{\mathbf{C}}^{-1} = \hat{\mathbf{C}}^\top = \mathbf{T}^\top \cdot \mathbf{S}.$$

Portanto, neste caso, a aproximação inversa herda as propriedades de baixa complexidade de \mathbf{T} . Ainda, a matriz \mathbf{S} pode ser “absorvida” na etapa de quantização, como detalhado no Apêndice B.

Para o caso não-ortogonal, supor que \mathbf{T} é uma transformada de baixa complexidade. O conjunto de equações (4.6) proporciona expressões em forma fechada para os elementos multiplicativos α'_k , $k = 0, 1, \dots, 6$, ligados à transformada \mathbf{T}^{-1} . Portanto, uma condição suficiente para que a transformada inversa tenha baixa complexidade é que $\alpha'_k \in \mathcal{P}$, para $k = 0, 1, \dots, 6$.

Como ilustração, considerar a transformada SDCT introduzida explicitamente em (3.3), que é não-ortogonal. Neste caso, a matriz SDCT, definida no subespaço de Feig-Winograd, admite a fatoração dada por:

$$\text{FW}(\boldsymbol{\alpha}) = \mathbf{P} \cdot \mathbf{K}(\boldsymbol{\alpha}) \cdot \mathbf{B}_1 \cdot \mathbf{B}_2 \cdot \mathbf{B}_3,$$

em que $\boldsymbol{\alpha} = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^\top$. De (4.6), obtemos que

$$\boldsymbol{\alpha}' = [1/2 \ 1 \ 1/2 \ 1 \ 0 \ 1 \ 0]^\top.$$

Observar que os elementos de α' pertencem ao conjunto \mathcal{P} . Este vetor de parâmetros proporciona a relação:

$$\text{FW}(\alpha) \cdot [\text{FW}(\alpha')]^\top = 8 \cdot \mathbf{I}_8.$$

Portanto, a SDCT inversa, introduzida explicitamente em (3.4), é dada por:

$$[\text{FW}(\alpha)]^{-1} = \frac{1}{8} \cdot [\text{FW}(\alpha')]^\top.$$

Assim, no contexto de compressão de imagens, a SDCT inversa mantém a propriedade de baixa complexidade atingida pela SDCT.

4.6.4 Medidas de Proximidade

Com o objetivo de avaliar as aproximações para a DCT candidatas, consideramos as seguintes figuras de mérito:

- i) Erro de energia total [20];
- ii) Erro quadrático médio (MSE) [15, 85];
- iii) Ganho em codificação unificado [15, 31, 48];
- iv) Eficiência da transformada [15].

As medidas de erro de energia total e MSE são usadas para mensurar o quão próxima está uma dada aproximação $\hat{\mathbf{C}}$ à DCT exata \mathbf{C} . As medidas de ganho em codificação e eficiência da transformada capturam o desempenho em codificação e a capacidade de decorrelação de uma transformada, respectivamente [15].

Para avaliar o desempenho em codificação, o sinal de entrada é modelado como um processo de Markov de primeira ordem estacionário com média zero, variância unitária e coeficiente de correlação $\rho \in (0,1)$. Então, o sinal de entrada \mathbf{x} é definido pela matriz de covariância \mathbf{R}_x , cujo (i,j) -ésimo elemento é dado por:

$$(\mathbf{R}_x)_{ij} = \rho^{|i-j|}.$$

A matriz \mathbf{R}_x é simétrica e Toeplitz [69, p. 158].

Se $\mathbf{X} = \hat{\mathbf{C}} \cdot \mathbf{x}$ é o sinal de saída, em que $\hat{\mathbf{C}}$ é uma aproximação para a DCT, então sua matriz de covariância é dada por:

$$\mathbf{R}_{\mathbf{X}} = \hat{\mathbf{C}} \cdot \mathbf{R}_{\mathbf{x}} \cdot \hat{\mathbf{C}}^{\top}.$$

Geralmente, é considerado que imagens naturais satisfazem as suposições teóricas anteriores, com $\rho \approx 0.95$ [15, 48, 55].

A seguir, são descritas matematicamente as figuras de mérito previamente consideradas.

Erro de energia total

Cada linha de uma matriz de transformação pode ser considerada como os coeficientes de um filtro de resposta ao impulso finito (FIR) [19, 72]. Baseado em [32], a magnitude da diferença entre as funções de transferência associadas aos filtros FIR induzidos pela DCT e a SDCT foi utilizada em [20] para obter uma medida de similaridade. Tal medida, chamada de erro de energia, foi posteriormente usada como uma medida de proximidade para outras aproximações [6, 20].

Para $m, n = 0, 1, \dots, 7$, sejam os sinais discretos $h_m[n]$ e $\hat{h}_m[n]$, formados pelas $(m + 1, n + 1)$ -ésimas entradas da DCT exata \mathbf{C} e de uma dada aproximação $\hat{\mathbf{C}}$, respectivamente. Tomando a transformada de Fourier de tempo discreto destes sinais, obtemos

$$H_m(\omega) = \sum_{n=0}^7 h_m[n] e^{-jn\omega},$$

$$\hat{H}_m(\omega) = \sum_{n=0}^7 \hat{h}_m[n] e^{-jn\omega},$$

em que $j = \sqrt{-1}$ e $\omega \in [0, 2\pi]$. As funções $H_m(\omega)$ e $\hat{H}_m(\omega)$, $m = 0, 1, \dots, 7$, representam as funções de transferência das $(m + 1)$ -ésimas linhas das matrizes \mathbf{C} e $\hat{\mathbf{C}}$, respectivamente. Assim, o erro de energia da $(m + 1)$ -ésima linha matricial é

dado por

$$\epsilon_m = \int_0^\pi |H_m(\omega) - \hat{H}_m(\omega)|^2 d\omega.$$

O erro de energia total associado a $\hat{\mathbf{C}}$ é simplesmente fornecido por

$$\epsilon(\hat{\mathbf{C}}) = \sum_{m=0}^7 \epsilon_m. \quad (4.16)$$

Como descrito em [20], $\epsilon(\hat{\mathbf{C}})$ é uma medida de similaridade entre \mathbf{C} e a aproximação $\hat{\mathbf{C}}$ considerada.

Embora definida originalmente no domínio espectral, a equação (4.16) pode ser expressada usando a norma matricial de Frobenius. Com efeito, usando o teorema de Parseval [47, p. 18] temos que:

$$\frac{1}{\pi} \int_0^\pi |H_m(\omega) - \hat{H}_m(\omega)|^2 d\omega = \sum_{n=0}^7 |h_m[n] - \hat{h}_m[n]|^2.$$

Portanto, o erro de energia total é dado por

$$\begin{aligned} \epsilon(\hat{\mathbf{C}}) &= \sum_{m=0}^7 \int_0^\pi |H_m(\omega) - \hat{H}_m(\omega)|^2 d\omega \\ &= \pi \sum_{m=0}^7 \sum_{n=0}^7 |h_m[n] - \hat{h}_m[n]|^2 \\ &= \pi \|\mathbf{C} - \hat{\mathbf{C}}\|_F^2. \end{aligned} \quad (4.17)$$

Assim, o erro de energia total é uma medida de proximidade entre \mathbf{C} e $\hat{\mathbf{C}}$ com respeito à norma de Frobenius.

Erro quadrático médio

O MSE para a aproximação $\hat{\mathbf{C}}$ é definido por [15, 55]:

$$\text{MSE}(\hat{\mathbf{C}}) = \frac{1}{8} \cdot \text{tr} \left((\mathbf{C} - \hat{\mathbf{C}}) \cdot \mathbf{R}_x \cdot (\mathbf{C} - \hat{\mathbf{C}})^\top \right). \quad (4.18)$$

Para minimizar o erro na saída entre a aproximação \hat{C} e a DCT exata C , o MSE entre C e \hat{C} deve ser minimizado [15, 55].

Ganho em codificação unificado

Dada uma aproximação \hat{C} , o ganho em codificação usual é definido por [15, p. 163]:

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

Contudo, como apontado em [48], esta medida é definida considerando transformadas ortogonais. Notar que para transformadas unitárias, $\|f_i\| = 1$. Portanto, nesse caso, o termo entre chaves é a razão entre a média aritmética e a média geométrica das variâncias.

Dado que neste trabalho também consideramos transformadas não-ortogonais, usamos como medida de avaliação o ganho em codificação unificado introduzido em [48]. Para $k = 0, 1, \dots, N-1$, sejam \mathbf{h}_k e \mathbf{g}_k os vetores linhas de \hat{C} e \hat{C}^{-1} , respectivamente. O ganho em codificação unificado associado a \hat{C} é dado por:

$$C_g^*(\hat{C}) = 10 \log_{10} \left\{ \prod_{k=0}^{N-1} \frac{1}{\sqrt[8]{A_k \cdot B_k}} \right\},$$

em que $A_k = \text{su} [(\mathbf{h}_k^\top \cdot \mathbf{h}_k) \odot \mathbf{R}_x]$, $\text{su}(\cdot)$ retorna a soma dos elementos do seu argumento matricial, o operador \odot denota o produto matricial de Hadamard [69] e $B_k = \|\mathbf{g}_k\|^2$. Para o caso de transformadas ortogonais, os valores de $C_g(\hat{C})$ e $C_g^*(\hat{C})$ coincidem.

As aproximações que exibem altos valores de ganho em codificação fornecem melhor compactação de energia em poucos coeficientes [55]. Para $\rho = 0.95$, os valores correspondentes às transformadas KLT e DCT são de 8.8462 dB e 8.8259 dB, respectivamente [15].

Eficiência da transformada

A medida de eficiência da transformada relacionada à aproximação $\hat{\mathbf{C}}$ é definida por [15, 16]:

$$\eta(\hat{\mathbf{C}}) = \frac{\sum_{i=1}^8 |(\mathbf{R}_{\mathbf{x}})_{ii}|}{\sum_{i=1}^8 \sum_{j=1}^8 |(\mathbf{R}_{\mathbf{x}})_{ij}|} \cdot 100, \quad (4.19)$$

em que $(\mathbf{R}_{\mathbf{x}})_{ij}$ e $(\mathbf{R}_{\mathbf{X}})_{ij}$ são os (i,j) -ésimos elementos das matrizes $\mathbf{R}_{\mathbf{x}}$ e $\mathbf{R}_{\mathbf{X}}$ respectivamente.

A eficiência da transformada $\eta(\hat{\mathbf{C}})$ é uma medida da capacidade de descorrelação da aproximada $\hat{\mathbf{C}}$ [15]. A transformada KLT converte sinais em coeficientes completamente descorrelacionados e seu valor de eficiência é de 100, para qualquer valor de ρ .

4.7 Otimização sobre o Subespaço de Feig-Winograd

4.7.1 Otimização Multicritério

Nesta seção, é proposto um problema de otimização considerando todos os critérios discutidos na seção anterior. No contexto de compressão de imagens, estes critérios geralmente encontram-se em conflito, no sentido que transformadas que apresentam bons valores em termos de algumas medidas particulares, podem não apresentar bons valores para outras medidas e vice-versa. Portanto, neste trabalho propomos um problema de otimização considerando todas estas medidas, em vez de considerá-las separadamente. Mais formalmente, temos o seguinte problema de otimização multicritério [24, 59]:

$$\min_{\alpha} \left(\epsilon(\hat{\mathbf{C}}), \text{MSE}(\hat{\mathbf{C}}), -C_g^*(\hat{\mathbf{C}}), -\eta(\hat{\mathbf{C}}), \mathcal{A}(\alpha), \mathcal{S}(\alpha) \right). \quad (4.20)$$

Dado que $C_g^*(\hat{\mathbf{C}})$ e $\eta(\hat{\mathbf{C}})$ precisam ser maximizados, consideramos os valores negativos destas quantidades.

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

Observar que, no caso de um problema de otimização simples, o significado de *mínimo* está relacionado diretamente com a ordem canônica \leq em \mathbb{R} . Isto não acontece com um problema de otimização multicritério. Dado que não existe uma ordem canônica em \mathbb{R}^p para $p \geq 2$, então a definição de *mínimo* em (4.20) está relacionada com uma determinada ordem em \mathbb{R}^p previamente estabelecida.

A seguinte definição será de utilidade para a definição de uma ordem usada na solução de (4.20).

Definição 4 Um conjunto $\mathcal{C} \subset \mathbb{R}^p$ ($p \geq 2$) é um cone se para todo elemento $\mathbf{d} \in \mathcal{C}$ e todo número real $\alpha > 0$, se satisfaz que $\alpha \mathbf{d} \in \mathcal{C}$.

Em geral, ordens são definidas em relação a um cone. Em particular, consideremos o cone:

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

Assim, definimos uma nova relação binária “ \leq ” em \mathbb{R}^p ($p \geq 2$) por:

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

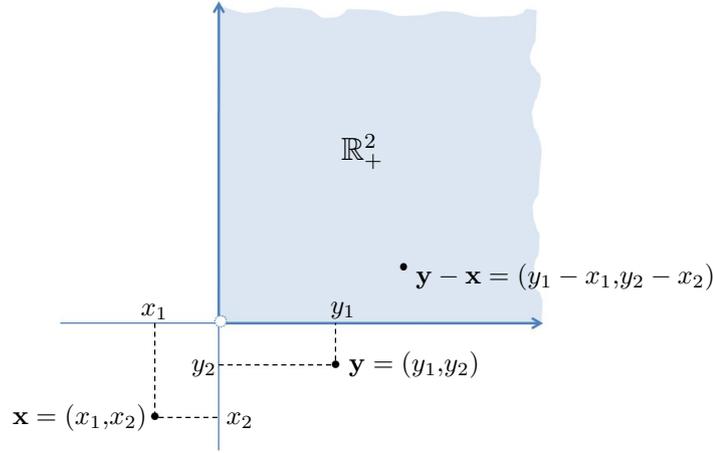
a qual pode ser escrita na forma alternativa:

$$\mathbf{x} \leq \mathbf{y} \iff x_i \leq y_i \forall i \text{ e } x_j < y_j \text{ para algum } j. \quad (4.21)$$

Esta ordem parcial respeito do cone \mathbb{R}_+^p é conhecida como a *ordem de Pareto* em \mathbb{R}^p e o conceito de otimização multicritério com respeito a esta ordem é conhecido como *otimalidade de Pareto* [24]. Na Figura 4.3, ilustra-se a ordem de Pareto em \mathbb{R}^2 .

Resolução do problema de otimização

Para resolver o problema (4.20), neste trabalho consideramos a ordem de Pareto como definida em (4.21). Além disso, precisamos identificar o espaço de busca, o conjunto de restrições e a técnica de resolução.


 Figura 4.3: Ilustração da ordem de Pareto em \mathbb{R}^2 ($\mathbf{x} \leq \mathbf{y}$).

Uma condição é que as soluções candidatas α gerem transformadas \mathbf{T} de baixa complexidade. Portanto, temos que $\alpha_k \in \mathcal{P}$, $k = 0, 1, \dots, 6$. Assim, o espaço de busca para (4.20) é o conjunto \mathcal{P}^7 .

Observar que (4.20) é um problema de otimização restrito. Com efeito, somente consideramos soluções candidatas cuja transformada inversa possua baixa complexidade. Resumindo, requeremos que tanto as transformadas obtidas como as suas inversas possuam entradas no conjunto $\mathcal{P} = \{0, \pm \frac{1}{2}, \pm 1, \pm 2\}$.

Como consequência da restrição anterior, reconhecemos que (4.20) é um problema analiticamente intratável. Portanto, é feita uma busca computacional exaustiva sobre o conjunto \mathcal{P}^7 , que contém $7^7 = 823543$ elementos, com o objetivo de identificar as soluções *eficientes* de (4.20) [24, 59].

Seja

$$\mathcal{F} = \{\epsilon(\cdot), \text{MSE}(\cdot), -C_g^*(\cdot), -\eta(\cdot), \mathcal{A}(\cdot), \mathcal{S}(\cdot)\}$$

o conjunto de funções objetivos considerado em (4.20). Então, o conjunto de soluções eficientes no sentido da otimalidade de Pareto [24, p. 24] é definido, segundo (4.21), por:

$$\mathcal{S}_E = \{\alpha^* \in \mathcal{P}^7 : \text{não existe } \alpha \in \mathcal{P}^7 \text{ tal que } f(\alpha) \leq f(\alpha^*) \text{ para todo } f \in \mathcal{F} \\ \text{e } f_0(\alpha) < f_0(\alpha^*) \text{ para algum } f_0 \in \mathcal{F}\}.$$

Suporte computacional

Com o objetivo de implementar a busca computacional do conjunto de soluções eficientes de (4.20), foi usado a linguagem de programação C conjuntamente com a biblioteca GNU Scientific Library (GSL) na sua versão 1.15 [29]. O diagrama da estrutura do laço de busca é apresentado na Figura 4.4. Os tempos de execução sobre um computador com processador Intel Pentium Dual-Core T4300 @ 2.10 Hz, 2 GB de RAM e sistema operacional Linux 3.2.0-48-generic são proporcionados na Tabela 4.2.

Tabela 4.2: Tempos de execução

Tipo de transformada	Tempo de execução (em segundos)
Ortogonal	34.98
Não-ortogonal	12.08

4.7.2 Soluções Eficientes e Transformadas Ótimas

Como resultado da busca computacional, obtemos 16 soluções eficientes distintas denotadas por α_i^* , $i = 1, \dots, 16$, mostradas na Tabela 4.3. Cada solução eficiente implica uma transformada ótima dada por

$$\mathbf{T}_i = \text{FW}(\alpha_i^*).$$

A forma explícita de cada transformada \mathbf{T}_i é obtida diretamente de (4.3).

Observar que, de todas as soluções eficientes obtidas, algumas resultam em transformadas ortogonais já conhecidas na literatura. Em particular, temos que:

- i) \mathbf{T}_1 é a aproximação de nível 1 para a DCT introduzida por Lengwehasatit e Ortega [54];
- ii) \mathbf{T}_2 é a RDCT introduzida em [20];
- iii) \mathbf{T}_3 é a MRDCT proposta em [5].

Além disso, as matrizes $\mathbf{T}_4, \mathbf{T}_5, \dots, \mathbf{T}_{16}$ são novas transformadas. Excetuando \mathbf{T}_{16} , todas as matrizes obtidas satisfazem (4.12) e, portanto, conduzem a aproximações ortonormais.

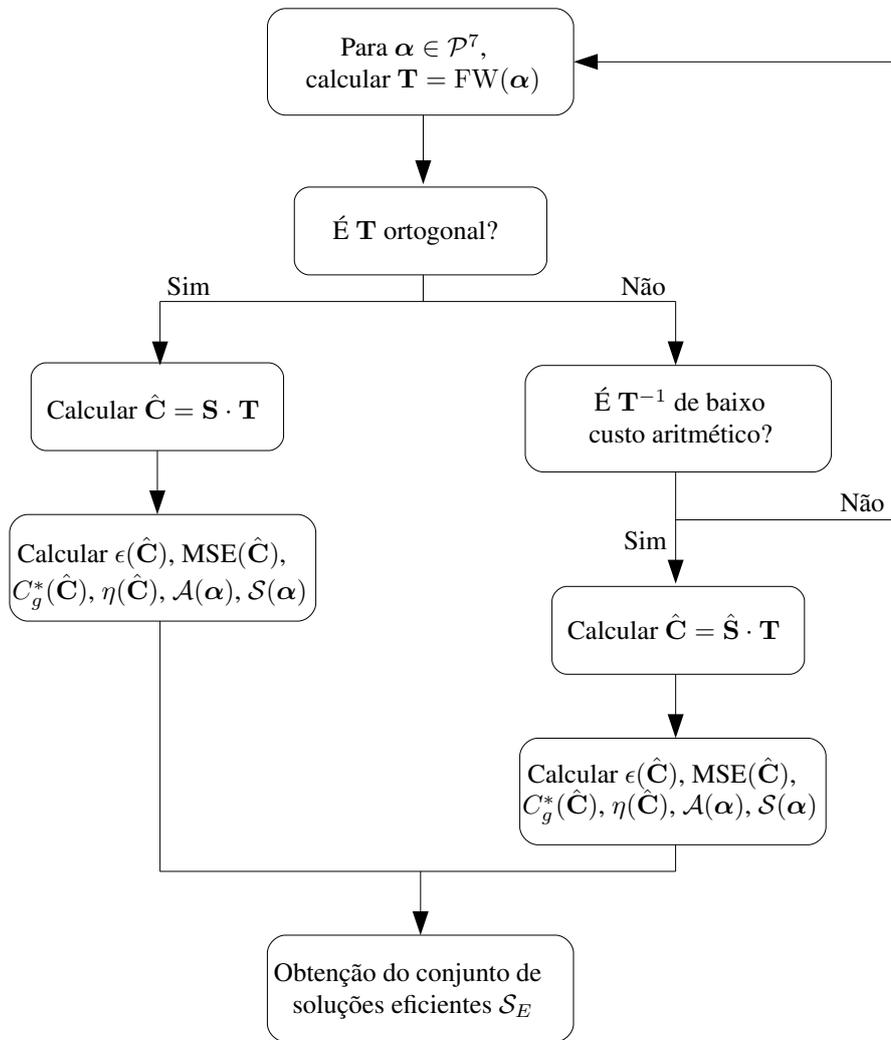


Figura 4.4: Estrutura do laço de busca em \mathcal{P}^7 .

Tabela 4.3: Soluções eficientes no subespaço de Feig-Winograd

i	Solução eficiente (α_i^*)
1	$[1 \ 1 \ 1 \ 1 \ 1 \ 1/2 \ 0]^T$
2	$[1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]^T$
3	$[1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]^T$
4	$[1 \ 2 \ 0 \ 1 \ 0 \ 1 \ 0]^T$
5	$[0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0]^T$
6	$[0 \ 2 \ 1 \ 1 \ 1 \ 1 \ 0]^T$
7	$[0 \ 2 \ 2 \ 1 \ 1 \ 1 \ 0]^T$
8	$[2 \ 2 \ 0 \ 1 \ 0 \ 1 \ 1/2]^T$
9	$[1 \ 2 \ 1 \ 1 \ 1 \ 1 \ 0]^T$
10	$[1 \ 1 \ 0 \ 1 \ 0 \ 1/2 \ 0]^T$
11	$[0 \ 1 \ 1 \ 1 \ 1 \ 1/2 \ 0]^T$
12	$[0 \ 1 \ 2 \ 1 \ 1 \ 1/2 \ 0]^T$
13	$[0 \ 2 \ 1 \ 1 \ 1/2 \ 1 \ 0]^T$
14	$[0 \ 1 \ 1 \ 1 \ 1/2 \ 1/2 \ 0]^T$
15	$[2 \ 1 \ 0 \ 1 \ 0 \ 1/2 \ 1/2]^T$
16	$[1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0]^T$

Aproximações Ortonormais

Um exame cuidadoso mostra que se satisfazem as seguintes relações entre as transformadas ortogonais:

$$\begin{aligned} \mathbf{T}_9 &= \mathbf{D}_1 \cdot \mathbf{T}_1, & \mathbf{T}_{10} &= \mathbf{D}_2 \cdot \mathbf{T}_4, & \mathbf{T}_{11} &= \mathbf{D}_2 \cdot \mathbf{T}_6, & \mathbf{T}_{12} &= \mathbf{D}_2 \cdot \mathbf{T}_7, \\ \mathbf{T}_{13} &= \mathbf{D}_3 \cdot \mathbf{T}_7, & \mathbf{T}_{14} &= \mathbf{D}_4 \cdot \mathbf{T}_7, & \mathbf{T}_{15} &= \mathbf{D}_2 \cdot \mathbf{T}_8, \end{aligned}$$

em que

$$\begin{aligned} \mathbf{D}_1 &= \text{diag}(1, 1, 2, 1, 1, 1, 2, 1), & \mathbf{D}_2 &= \text{diag}\left(1, 1, \frac{1}{2}, 1, 1, 1, \frac{1}{2}, 1\right), \\ \mathbf{D}_3 &= \text{diag}\left(1, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}\right), & \mathbf{D}_4 &= \text{diag}\left(1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right). \end{aligned}$$

Este fato motiva a seguinte definição, que resulta útil no formalismo matemático para a obtenção das aproximações para a DCT.

Definição 5 Dizemos que duas transformadas são equivalentes se existe uma matriz diagonal \mathbf{D}_+ , com entradas reais positivas, tal que

$$\mathbf{T}' = \mathbf{D}_+ \cdot \mathbf{T}. \quad (4.22)$$

É direto observar que a relação (4.22) define uma relação de equivalência, que resulta ser um caso particular da equivalência de matrizes definida em [42, p. 332]. Para uma transformada $\mathbf{T} \in \mathcal{Im}(\text{FW})$, denotamos sua classe de equivalência como $[\mathbf{T}]$.

O seguinte teorema de unicidade enuncia que duas transformadas equivalentes fornecem, de fato, a mesma aproximação para a DCT.

Teorema 1 Se \mathbf{T} e \mathbf{T}' são transformadas equivalentes no subespaço de Feig-Winograd, então \mathbf{T} e \mathbf{T}' fornecem a mesma aproximação para a DCT.

Prova: Supomos primeiro que as transformadas \mathbf{T} e \mathbf{T}' satisfazem (4.12), e sejam \mathbf{S} e \mathbf{S}' suas matrizes de ortogonalização segundo definidas em (4.10). Da equação (4.11), é suficiente provar que $\mathbf{S} = \mathbf{S}' \cdot \mathbf{D}_+$.

Observar que

$$\begin{aligned} \mathbf{S}' \cdot \mathbf{D}_+ &= \sqrt{(\mathbf{T}' \cdot \mathbf{T}'^\top)^{-1}} \cdot \mathbf{D}_+ \\ &= \sqrt{(\mathbf{D}_+ \cdot \mathbf{T} \cdot \mathbf{T}^\top \cdot \mathbf{D}_+)^{-1}} \cdot \mathbf{D}_+ \\ &= \sqrt{\mathbf{D}_+^{-1} \cdot (\mathbf{T} \cdot \mathbf{T}^\top)^{-1} \cdot \mathbf{D}_+^{-1}} \cdot \mathbf{D}_+. \end{aligned}$$

Usando o fato que \mathbf{D}_+ e $\mathbf{T} \cdot \mathbf{T}^\top$ são matrizes diagonais, obtemos que:

$$\begin{aligned} \mathbf{S}' \cdot \mathbf{D}_+ &= \sqrt{(\mathbf{T} \cdot \mathbf{T}^\top)^{-1} \cdot (\mathbf{D}_+^{-1})^2 \cdot \mathbf{D}_+} \\ &= \sqrt{(\mathbf{T} \cdot \mathbf{T}^\top)^{-1}} \\ &= \mathbf{S}. \end{aligned}$$

Supomos agora que \mathbf{T} e \mathbf{T}' não satisfazem (4.12), e sejam $\hat{\mathbf{S}}$ e $\hat{\mathbf{S}}'$ suas matrizes de ortogonalização segundo definidas em (4.13). Da equação (4.14), é suficiente provar que $\hat{\mathbf{S}} = \hat{\mathbf{S}}' \cdot \mathbf{D}_+$. Novamente, usando o fato que \mathbf{D}_+ e $\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)$ são matrizes diagonais, obtemos que:

$$\begin{aligned} \hat{\mathbf{S}}' \cdot \mathbf{D}_+ &= \sqrt{[\text{diag}(\mathbf{T}' \cdot \mathbf{T}'^\top)]^{-1} \cdot \mathbf{D}_+} \\ &= \sqrt{[\text{diag}(\mathbf{D}_+ \cdot \mathbf{T} \cdot \mathbf{T}^\top \cdot \mathbf{D}_+)]^{-1} \cdot \mathbf{D}_+} \\ &= \sqrt{[\mathbf{D}_+ \cdot \text{diag}(\mathbf{T} \cdot \mathbf{T}^\top) \cdot \mathbf{D}_+]^{-1} \cdot \mathbf{D}_+} \\ &= \sqrt{[\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)]^{-1} \cdot \mathbf{D}_+^{-2} \cdot \mathbf{D}_+} \\ &= \sqrt{[\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)]^{-1}} \\ &= \hat{\mathbf{S}} \end{aligned}$$

e a prova conclui. ■

Pelo Lema 1, é possível caracterizar as classes de equivalência no subespaço de Feig-Winograd. Com efeito, temos que:

$$\begin{aligned} \llbracket \mathbf{T} \rrbracket &= \{ \mathbf{T}' \in \mathcal{I}m(\text{FW}) : \mathbf{T}' = \mathbf{D}_+ \cdot \mathbf{T}, \\ &\quad \mathbf{D}_+ = \text{diag}(d_0, d_1, d_2, d_1, d_0, d_1, d_2, d_1), d_k \in \mathbb{R}_+, k = 0, 1, 2. \}, \end{aligned}$$

em que \mathbb{R}_+ denota o conjunto dos reais positivos.

No subespaço de Feig-Winograd, obtemos as seguintes relações de equivalência sobre as transformadas ótimas consideradas:

$$\begin{aligned} \{ \mathbf{T}_1, \mathbf{T}_9 \} &\subset \llbracket \mathbf{T}_1 \rrbracket, \quad \{ \mathbf{T}_4, \mathbf{T}_{10} \} \subset \llbracket \mathbf{T}_4 \rrbracket, \quad \{ \mathbf{T}_6, \mathbf{T}_{11} \} \subset \llbracket \mathbf{T}_6 \rrbracket, \\ \{ \mathbf{T}_8, \mathbf{T}_{15} \} &\subset \llbracket \mathbf{T}_8 \rrbracket, \quad \{ \mathbf{T}_7, \mathbf{T}_{12}, \mathbf{T}_{13}, \mathbf{T}_{14} \} \subset \llbracket \mathbf{T}_7 \rrbracket. \end{aligned}$$

Como consequência, embora estas transformadas são diferentes em pares, as aproximações ortonormais que são derivadas das mesmas (cf. (4.11)) são iguais, pois como foi provado no Teorema 1, matrizes na mesma classe de equivalência fornecem a mesma aproximação para a DCT.

As transformadas ótimas restantes não apresentam relação de equivalência com nenhuma outra transformada conhecida, sendo portanto estruturalmente novas. Assim, apresentamos \mathbf{T}_4 , \mathbf{T}_5 , \mathbf{T}_6 , \mathbf{T}_7 e \mathbf{T}_8 como novas transformadas ortogonais. As mesmas são apresentadas explicitamente a seguir:

$$\mathbf{T}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}, \mathbf{T}_5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 1 & -1 & 0 & 0 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 \end{bmatrix},$$

$$\mathbf{T}_6 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & -1 & -1 & 0 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 1 & 0 & 0 & -1 & 1 & 0 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 1 & -1 & 0 & 0 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 1 & 0 & 0 & -1 & 1 & 0 \end{bmatrix}, \mathbf{T}_7 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 & 0 & -1 & -2 & 0 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 0 & 0 & -1 & 1 & 0 & 0 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 0 & 0 & 2 & -2 & 0 & 0 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & 0 & 0 & -2 & 1 & 0 \end{bmatrix},$$

$$\mathbf{T}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 0 & 0 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 0 & -\frac{1}{2} & -2 & 0 & 0 & 2 & \frac{1}{2} & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -2 & \frac{1}{2} & 0 & 0 & -\frac{1}{2} & 2 & 0 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ \frac{1}{2} & 0 & 0 & -2 & 2 & 0 & 0 & -\frac{1}{2} \end{bmatrix}.$$

Aproximação Quasi-ortogonal

De todas as soluções eficientes, obtemos somente uma transformação que conduz a uma aproximação não-ortogonal para a DCT. Esta solução particular, denotada por \mathbf{T}_{16} , é uma nova transformada dada por:

$$\mathbf{T}_{16} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 0 & 0 \end{bmatrix}.$$

Observar que \mathbf{T}_{16} não satisfaz (4.12). Considerando a medida de desvio da diagonalidade discutida em [28] (cf. (4.15)) temos que $\delta(\mathbf{T}_{16} \cdot \mathbf{T}_{16}^\top) = 0.125$. Para

comparar, a SDCT também fornece uma aproximação quasi-ortogonal, cujo desvio da ortogonalidade é de 0.20. Neste sentido, a nova matriz \mathbf{T}_{16} é “mais ortogonal” que a SDCT. Portanto, aceitamos \mathbf{T}_{16} como uma matriz que fornece uma aproximação quasi-ortogonal para a DCT.

Assim, a aproximação quasi-ortogonal, derivada de (4.14), é dada por:

$$\hat{\mathbf{C}} = \hat{\mathbf{S}}_{16} \cdot \mathbf{T}_{16},$$

em que $\hat{\mathbf{S}}_{16} = \text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{\sqrt{8}}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2} \right)$ é obtida de (4.13).

A Tabela 4.4 resume a discussão anterior sobre as novas transformadas e suas relações.

Tabela 4.4: Transformadas ótimas no subespaço de Feig-Winograd

Transformada	Ortogonal?	Descrição
\mathbf{T}_1	Sim	Proposta em [54]
\mathbf{T}_2	Sim	Proposta em [20]
\mathbf{T}_3	Sim	Proposta em [5]
\mathbf{T}_4	Sim	Nova transformada
\mathbf{T}_5	Sim	Nova transformada
\mathbf{T}_6	Sim	Nova transformada
\mathbf{T}_7	Sim	Nova transformada
\mathbf{T}_8	Sim	Nova transformada
\mathbf{T}_9	Sim	Equivalente a \mathbf{T}_1
\mathbf{T}_{10}	Sim	Equivalente a \mathbf{T}_4
\mathbf{T}_{11}	Sim	Equivalente a \mathbf{T}_6
\mathbf{T}_{12}	Sim	Equivalente a \mathbf{T}_7
\mathbf{T}_{13}	Sim	Equivalente a \mathbf{T}_7
\mathbf{T}_{14}	Sim	Equivalente a \mathbf{T}_7
\mathbf{T}_{15}	Sim	Equivalente a \mathbf{T}_8
\mathbf{T}_{16}	Não	Nova transformada

4.7.3 Avaliação das Novas Transformadas

Submetemos todas as soluções eficientes ao procedimento de aproximação descrito em (4.11) e (4.14), a depender da solução considerada satisfazer (4.12) ou não. As

aproximações obtidas foram avaliadas segundo: (i) complexidade computacional e (ii) medidas de proximidade respeito da DCT.

Somente transformadas não equivalentes foram consideradas, como discutido na seção anterior. Portanto, somente consideramos as seguintes transformadas: \mathbf{T}_1 , \mathbf{T}_2 , \mathbf{T}_3 , \mathbf{T}_4 , \mathbf{T}_5 , \mathbf{T}_6 , \mathbf{T}_7 , \mathbf{T}_8 , e \mathbf{T}_{16} .

A Tabela 4.5 apresenta os valores obtidos de custo aritmético e medidas de proximidade para as transformadas ótimas consideradas.

Tabela 4.5: Medidas de avaliação das transformadas ótimas consideradas

Transformada	$\epsilon(\hat{\mathbf{C}})$	$\text{MSE}(\hat{\mathbf{C}})$	$C_g^*(\hat{\mathbf{C}})$	$\eta(\hat{\mathbf{C}})$	$\mathcal{A}(\alpha)$	$\mathcal{S}(\alpha)$
\mathbf{T}_1	0.87	0.006	8.39	88.70	24	2
\mathbf{T}_2	1.79	0.010	8.18	87.43	22	0
\mathbf{T}_3	8.66	0.059	7.33	80.90	14	0
\mathbf{T}_4	7.73	0.056	7.54	81.99	16	2
\mathbf{T}_5	8.66	0.059	7.37	81.18	18	0
\mathbf{T}_6	7.73	0.055	7.58	82.27	20	2
\mathbf{T}_7	7.53	0.054	7.56	82.70	20	6
\mathbf{T}_8	7.41	0.053	7.58	83.08	20	10
\mathbf{T}_{16}	3.32	0.021	6.05	83.08	18	0
Média	5.97	0.041	7.51	83.48	19	2

4.7.4 Discussão e Comparação

Os valores apresentados na Tabela 4.5 mostram que a transformada \mathbf{T}_1 tem desempenho superior às outras transformadas consideradas para todas as medidas de avaliação. Como mostrado na Tabela 4.4, a transformada \mathbf{T}_1 corresponde à aproximação de nível 1 introduzida por Lengwehasatit e Ortega [54], conhecida por ser uma muito boa aproximação para a DCT. Contudo, também é conhecida por sua comparativamente alta complexidade computacional, portanto, não resulta muito usada no contexto de aproximações para a DCT. A aproximação \mathbf{T}_2 , conhecida como a RDCT [20], também apresenta uma boa proximidade com a DCT, enquanto que requer menos esforço computacional que \mathbf{T}_1 . Além disso, a transformada \mathbf{T}_3 introduzida em [5] tem a distinção de ter um custo aritmético de somente 14 adições, conseguindo assim a menor

Tabela 4.6: Medidas de avaliação para as transformadas que não pertencem ao formalismo Feig-Winograd

Transformada	Ortogonal?	$\epsilon(\hat{\mathbf{C}})$	$\text{MSE}(\hat{\mathbf{C}})$	$C_g^*(\hat{\mathbf{C}})$	$\eta(\hat{\mathbf{C}})$	$\mathcal{A}(\boldsymbol{\alpha})$	$\mathcal{S}(\boldsymbol{\alpha})$
BAS ₁	Não	4.19	0.019	6.27	83.17	21	0
BAS ₂	Sim	5.93	0.024	8.12	86.86	18	2
BAS ₃	Sim	6.85	0.028	7.91	85.38	18	0
BAS ₄	Sim	4.09	0.021	8.33	88.22	24	4
BAS ₅	Sim	26.86	0.071	7.91	85.38	18	0
BAS ₆	Sim	26.86	0.071	7.91	85.64	16	0
BAS ₇	Sim	26.40	0.068	8.12	86.86	18	2
BAS ₈	Sim	35.06	0.102	7.95	85.31	24	0
WHT	Sim	5.05	0.025	7.95	85.31	24	0
Média		15.70	0.048	7.83	85.79	20	1

complexidade dentre todas as transformadas ótimas no subespaço de Feig-Winograd (cf. (4.8)–(4.9)).

Considerando as novas transformadas propostas, temos que a matriz \mathbf{T}_4 possui bom desempenho em termos de medidas de proximidade, embora exija somente 16 adições. As matrizes \mathbf{T}_5 e \mathbf{T}_{16} , por requerer ambas somente 18 adições, podem serem comparadas entre elas. Enquanto \mathbf{T}_5 fornece uma aproximação ortonormal, \mathbf{T}_{16} resulta em uma aproximação não-ortogonal. A matriz \mathbf{T}_{16} apresenta o menor valor de erro de energia total e MSE de entre todas as novas transformadas. Em termos de ganho em codificação unificado, a transformada \mathbf{T}_5 supera à transformada \mathbf{T}_{16} ; além disso, em termos de eficiência da transformada, \mathbf{T}_{16} supera à transformada \mathbf{T}_5 .

As novas transformadas \mathbf{T}_6 , \mathbf{T}_7 e \mathbf{T}_8 exibem bons desempenhos, no entanto, requerem maior custo aritmético. Contudo, sua complexidade aditiva é menor que a requerida pela transformada \mathbf{T}_1 .

Para uma análise posterior, comparamos as novas transformadas com outras aproximações que não pertencem ao subespaço de Feig-Winograd: a WHT [39] e a série de aproximações introduzida por Bouguezzel, Ahmad e Swamy denotadas por BAS₁ [9], BAS₂ [8], BAS₃ [10], BAS₄ [12], BAS₅ [13], BAS₆ [13], BAS₇ [13] e BAS₈ [7]. A Tabela 4.6 apresenta as medidas de avaliação para estas aproximações.

A Figura 4.5 apresenta os gráficos relacionando o custo aritmético (número de

adições) com as medidas de avaliação discutidas. Nestes gráficos, cada transformada corresponde a um ponto etiquetado. Transformadas ortogonais são etiquetadas por círculos (\circ), ao passo que transformadas não-ortogonais são denotadas pelo símbolo (\times). As aproximações BAS_5 , BAS_6 , BAS_7 e BAS_8 não foram incluídas na Figura 4.5 (a) por apresentar valores excessivamente grandes, não comportando na escala adotada. Igualmente, a aproximação BAS_8 foi excluída da Figura 4.5 (b).

As Figuras 4.5 (a)-(b) indicam que as transformadas ótimas no subespaço de Feig-Winograd destacam-se em termos de erro de energia total e MSE. Além disso, as transformadas no formalismo Feig-Winograd apresentam melhores figuras de ganho em codificação, exceto para as transformadas que requerem 16 e 18 adições.

Em particular, considerando as transformadas que requerem somente 16 adições, observamos que T_4 supera à transformada BAS_6 em termos de erro de energia total e MSE. Contudo, BAS_6 apresenta melhor valor de ganho em codificação unificado que T_4 . Agora, considerando transformadas que requerem 18 adições, temos que as aproximações BAS_2 , BAS_3 , BAS_5 e BAS_7 apresentam melhores valores de ganho em codificação unificado que as transformadas propostas T_5 e T_6 . Como esperado, T_1 tem o melhor valor de ganho em codificação de entre todas as transformadas consideradas na Figura 4.5. Também, T_2 supera as transformadas não-Feig-Winograd em todas as métricas, excetuando BAS_4 que resulta melhor em termos de ganho em codificação unificado e eficiência da transformada. Contudo, BAS_4 precisa de duas adições a mais que T_2 .

Em termos de transformadas não-ortogonais, temos que T_{16} apresenta um valor de erro de energia total menor que BAS_1 , entanto que apresenta um desempenho similar nas outras medidas de avaliação. Contudo, a transformada T_{16} é 14.3% menos complexa em termos de custo aritmético.

Além da análise anterior, pode-se observar das Figuras 4.5 (a) e (b) que, dentre as transformadas ótimas, T_{16} apresenta um melhor balanço entre medida de avaliação e custo aritmético respeito de erro de energia total e MSE, enquanto que em termos de ganho em codificação, destacam-se as transformadas T_3 , BAS_2 , BAS_6 e BAS_7 , como apresentado na Figura 4.5 (c). Já em termos de eficiência da transformada, a Figura 4.5 (d) permite observar que as transformadas BAS_2 , BAS_6 e BAS_7 apresentam um melhor balanço entre medida de avaliação e custo aritmético.

Observar que a Figura 4.5 pode ser interpretada como resultado de quatro proble-

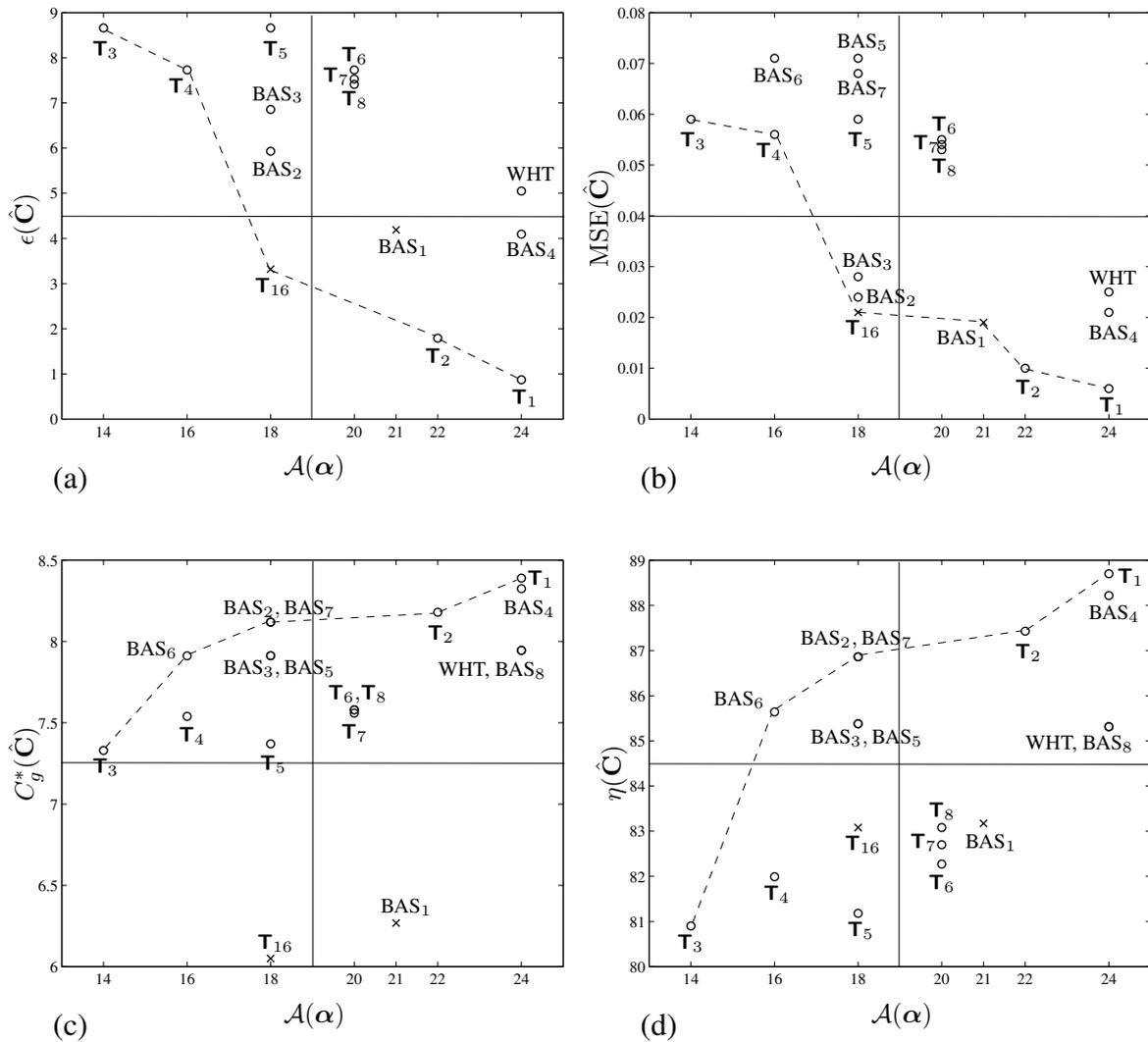


Figura 4.5: Gráficos de avaliação para as aproximações ótimas propostas; outras transformadas foram incluídas para comparação.

mas de otimização bi-objetivos diferentes [59, p. 245], em que são enfatizados um balanço entre custo aritmético e as medidas de avaliação consideradas. As soluções ótimas dos problemas de otimização bi-objetivos encontram-se sobre a fronteira do conjunto de soluções factíveis [24, p. 28]. Tais fronteiras são mostradas na Figura 4.5 e representam a fronteira de Pareto para cada problema de otimização [59, p. 11]. Portanto, neste sentido, identificamos as transformadas que se encontram sobre cada fronteira de Pareto como transformadas ótimas. Tais transformadas foram $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \mathbf{T}_4, \mathbf{T}_{16}, \text{BAS}_1, \text{BAS}_2, \text{BAS}_6$ e BAS_7 . Este conjunto reduzido de transformadas foi submetido a uma análise subsequente no contexto de compressão de imagens.

4.8 Compressão de Imagens

4.8.1 Metodologia

Com o objetivo de avaliar o desempenho das aproximações selecionadas, adotamos a compressão de imagens tipo JPEG (cf. Apêndice A) baseada em 45 imagens de 8 bits de tamanho 512×512 em escala de cinzas, obtidas de um banco de imagens público [79]. Cada imagem foi dividida em blocos de tamanho 8×8 e posteriormente a cada bloco foi aplicado a transformação 2-D, dependendo da aproximação DCT considerada. No Apêndice B é descrita algebricamente a operação de transformação 2-D, que depende se a aproximação para a DCT considerada é ortogonal ou não. Esta operação fornece 64 coeficientes no domínio da transformada para cada bloco.

Empregando a sequência zig-zag padrão [81], somente r coeficientes iniciais em cada bloco foram retidos e usados para reconstruir a imagem [20]. Os demais coeficientes são definidos como zero. Neste trabalho, adotamos $1 \leq r \leq 45$. Após a compressão, é aplicada a transformação 2-D inversa para reconstruir os dados processados. Posteriormente, a qualidade da imagem compactada foi avaliada.

A metodologia apresentada anteriormente é também descrita em [32] e suportada em [8, 9, 10, 12, 13]. Contudo, a diferença dos experimentos de compressão tipo JPEG descritos em [8, 9, 10, 12, 13, 32], adotamos a média das medidas de avaliação de qualidade de imagens considerando o conjunto de 45 imagens [5, 6, 20]. Esta tendência é, portanto, menos afetada por efeitos de variância e entradas fortuitas de dados, proporcionando assim uma metodologia mais robusta [49].

A degradação da imagem compactada foi avaliada usando as medidas de PSNR [78] e SSIM [87]. O PSNR é uma medida de qualidade muito usada em processamento de imagens [78] e o SSIM é considerado como um método complementar para avaliação da qualidade de imagens [87]. De fato, o SSIM considera luminância, contraste e estrutura da imagem para quantificar a degradação da imagem, resultando assim numa medida de qualidade subjetiva [86].

As Figuras 4.6(a)-(b) apresentam as medidas de qualidade obtidas para as transformadas identificadas como ótimas segundo a discussão apresentada no final da seção precedente. Este tipo de curvas de desempenho é geralmente empregado como uma ferramenta de comparação na literatura das aproximações para a DCT [5, 9, 10, 13, 20]. Para melhorar a visualização das curvas de desempenho, consideramos o *erro absoluto percentual* (APE) relativo à DCT das medidas de PSNR e SSIM. Por exemplo, considerando o PSNR, o APE é calculado segundo a seguinte expressão:

$$\text{APE (PSNR)} = \left| \frac{\text{PSNR}_C - \text{PSNR}_T}{\text{PSNR}_C} \right|,$$

em que PSNR_C e PSNR_T são os valores de PSNR considerando a DCT exata e uma dada aproximação T , respectivamente. O valor de APE (SSIM) é calculado de forma similar. Os valores de APE (PSNR) e APE (SSIM) são apresentados nas figuras 4.6(c)-(d).

4.8.2 Resultados e discussão

Como esperado, através de todos os métodos considerados, T_1 apresenta o melhor desempenho em termos de qualidade de imagem, mas também tem a maior complexidade computacional [9, 10, 12]. A transformada T_2 supera em desempenho às transformadas T_3 , T_4 , T_{16} e BAS_6 , entanto exibe um desempenho similar a BAS_2 e BAS_7 . As transformadas T_3 , T_4 e T_{16} apresentam um melhor desempenho que BAS_1 em termos de PSNR, para $r > 25$, e de SSIM, para todos os valores de r . A transformada não-ortogonal T_{16} supera às transformadas T_3 e T_4 . Além disso, T_{16} apresenta uma complexidade aritmética inferior que a transformada não-ortogonal BAS_1 . A transformada T_4 apresenta melhor desempenho que T_3 em termos de PSNR e SSIM para todo r , embora requer duas adições a mais para sua implementação.

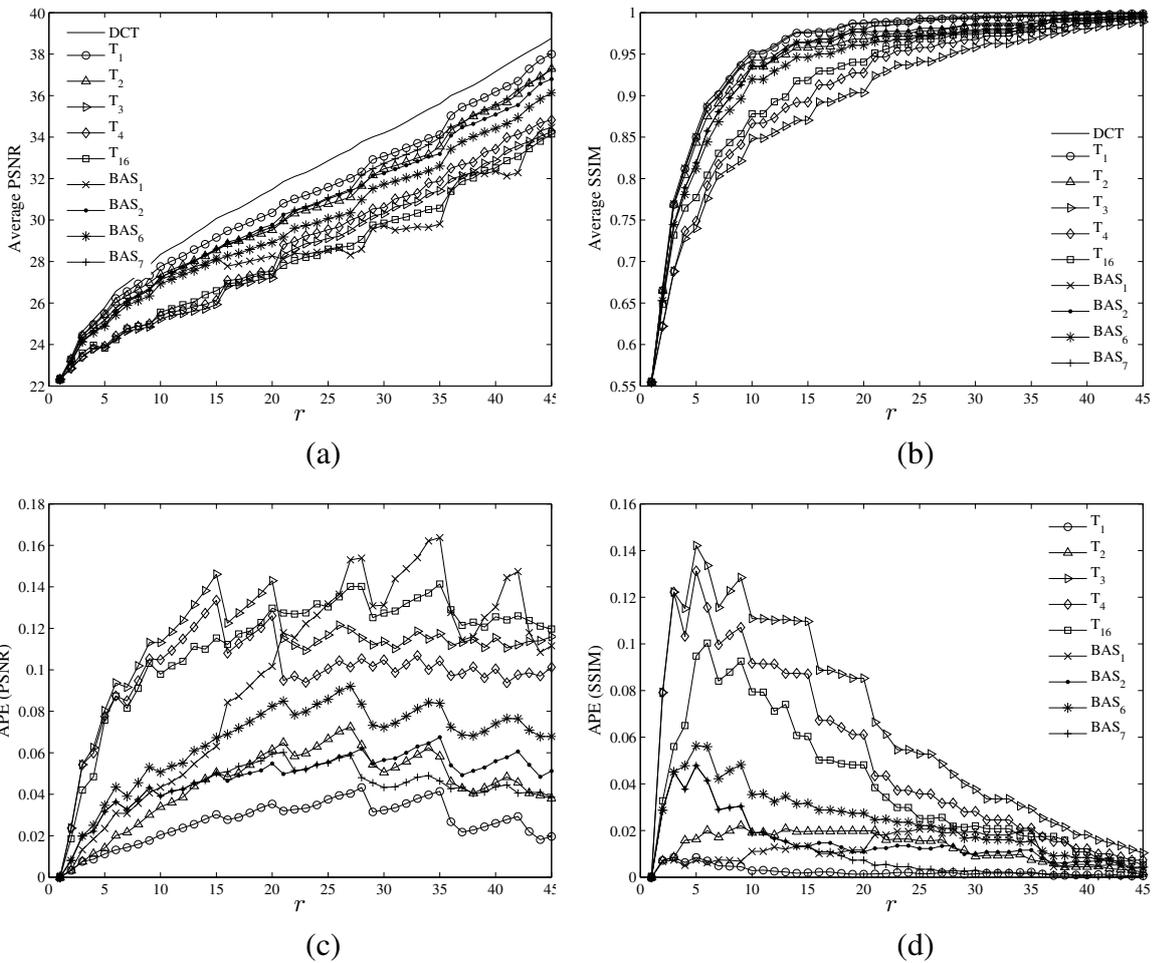


Figura 4.6: Medidas de qualidade das transformadas ótimas consideradas para vários valores de r segundo as seguintes figuras de mérito: (a) PSNR, (b) SSIM, (c) APE (PSNR) relativo à DCT, e (d) APE (SSIM) relativo à DCT.

Para uma análise qualitativa, as Figuras 4.7 e 4.8 apresentam imagens de tamanho 512×512 depois de serem submetidas à compressão tipo JPEG, como descrita no Apêndice A, para valores de $r = 10$ e $r = 25$, respectivamente. Estes valores de r correspondem a um truncamento de aproximadamente 60% e 85% dos coeficientes, respectivamente. As medidas de PSNR e SSIM correspondentes são apresentadas na Tabela 4.7.

Tabela 4.7: Medidas de PSNR e SSIM das imagens consideradas

Transformada	Imagem <i>Boat</i> ($r = 10$)		Imagem <i>Lena</i> ($r = 25$)	
	PSNR	SSIM	PSNR	SSIM
\mathbf{T}_1	28.360	0.967	35.176	0.995
\mathbf{T}_2	27.862	0.955	34.138	0.989
\mathbf{T}_3	25.190	0.882	31.838	0.970
\mathbf{T}_4	25.418	0.895	32.299	0.977
\mathbf{T}_{16}	25.805	0.903	31.602	0.985
BAS_1	27.567	0.960	31.452	0.980
BAS_2	27.774	0.956	34.794	0.991
BAS_6	27.352	0.943	33.819	0.986
BAS_7	27.774	0.956	35.433	0.995
DCT	28.972	0.969	37.886	0.997

4.9 Conclusão

Neste capítulo, introduzimos uma nova classe de matrizes baseadas numa parametrização da fatoração de Feig-Winograd para a DCT. Resolvendo um problema de otimização multicritério, várias aproximações para a DCT de baixa complexidade foram obtidas. De todas as aproximações obtidas, identificamos algumas como aproximações já citadas na literatura. Portanto, o procedimento utilizado proporciona um método matemático geral para obter estas transformadas. Além disso, são derivados aproximações para a DCT de bom desempenho. As novas transformadas foram avaliadas em termos de complexidade computacional, proximidade com a DCT, ganho em codificação e desempenho em compressão tipo JPEG. O custo aritmético das transformadas ótimas consideradas no subespaço de Feig-Winograd variam de 14 a 24 adições e de

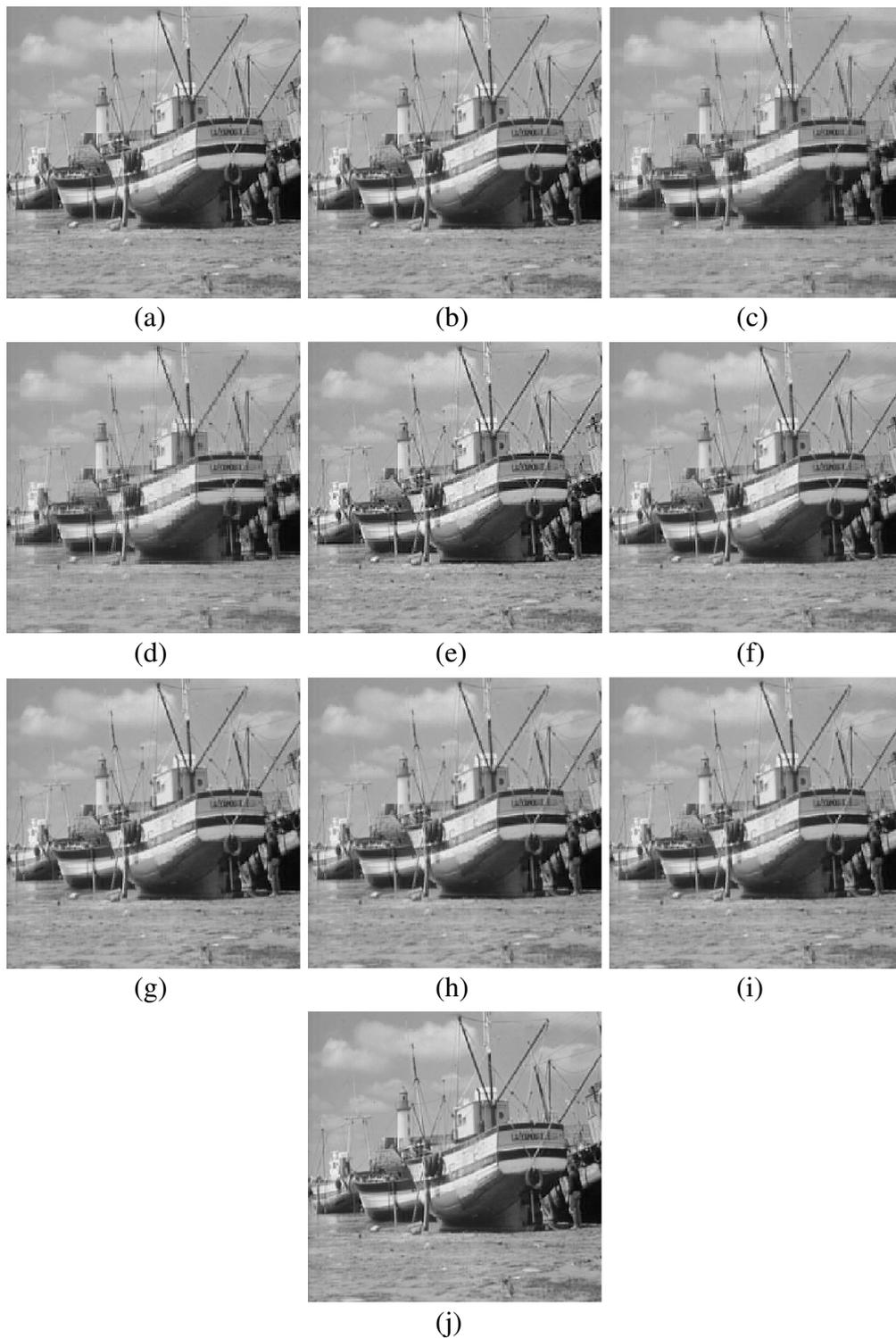


Figura 4.7: Imagem compactada *Boat* usando: (a) T_1 , (b) T_2 , (c) T_3 , (d) T_4 , (e) T_{16} , (f) BAS_1 , (g) BAS_2 , (h) BAS_6 , (i) BAS_7 , e (j) DCT, para $r = 10$.



Figura 4.8: Imagem compactada *Lena* usando: (a) \mathbf{T}_1 , (b) \mathbf{T}_2 , (c) \mathbf{T}_3 , (d) \mathbf{T}_4 , (e) \mathbf{T}_{16} , (f) BAS_1 , (g) BAS_2 , (h) BAS_6 , (i) BAS_7 , e (j) DCT, para $r = 25$.

0 a 10 deslocamentos. Um fato importante para destacar é que todas as transformadas na classe Feig-Winograd possuem a mesma estrutura do algoritmo rápido. Além disso, as transformadas inversas associadas têm um formalismo matemático similar e possuem um algoritmo rápido simples. Portanto, em termos de desenho de circuitos, é possível trocar transformadas com modificações mínimas de hardware. Em sistemas reconfiguráveis emergentes, é possível mudar o *modus operandi* baseados sobre a qualidade de quadro demandada e o consumo de energia requerido. Portanto, a classe proposta de aproximações podem ser candidatas de algoritmos rápidos neste contexto. Além do contexto de compressão de imagens, a classe Feig-Winograd de aproximações para a DCT pode ser aplicada à criptografia de dados seguindo o esquema introduzido em [11, 17]. Dado que as transformadas no subespaço de Feig-Winograd não requerem de multiplicações para sua implementação, os métodos propostos reduzem a complexidade computacional e estrutural dos esquemas de criptografia de imagens apresentados em [7].

Duas Aproximações para a DCT Baseadas na Fatoração de Chen

5.1 Introdução

Neste capítulo, primeiramente obtém-se um algoritmo rápido para a DCT-II de comprimento 8 baseado na fatoração de Chen *et al.* [18]. Este algoritmo baseia-se numa fatoração para a DCT-IV, que difere da DCT-II nos pontos de amostragem da função cosseno utilizado em suas definições. A relação entre ambos tipos de DCT é apresentado em [18] e [83]. Posteriormente, fazendo uso das funções $\text{sign}(\cdot)$ e $\text{round}(\cdot)$ como definidas em (1.2) e (3.5), respectivamente, são obtidas duas aproximações para a DCT livres de multiplicação. As transformadas obtidas são comparadas em termos de erro de energia total (cf. (4.17)) com as transformadas HT e WHT [39], a aproximação BAS introduzida em [9] e a SDCT introduzida em [32]. Finalmente, todas as transformadas consideradas são avaliadas no contexto de compressão de imagens em termos de PSNR [78] e SSIM [87]. Estas medidas fornecem valores que permitem avaliar quantitativamente a qualidade de compressão destas aproximações. Seguindo a metodologia usada na Seção 4.8, calculamos a média destes valores sobre um conjunto de 45 imagens padronizadas. Além disso, comparamos qualitativamente o desempenho destas aproximações ao serem aplicadas sobre uma imagem particular.

Neste capítulo, fazemos as seguintes contribuições:

- i) Propomos duas novas aproximações para a DCT de comprimento 8, baseadas

no algoritmo de Chen *et al.* [18] para a DCT exata;

- ii) Derivamos algoritmos rápidos para as transformadas propostas;
- iii) Avaliamos as aproximações propostas por meio de figuras de mérito;
- iv) Comparamos as técnicas propostas com as aproximações SDCT e BAS, além das transformadas HT e WHT, por serem livres de multiplicação.

5.2 Fatoração de Chen para a DCT

Em [18], Chen *et al.* propuseram uma fatoração para DCT-II baseada numa fatoração para DCT-IV. Estas duas versões da DCT diferem entre si nos pontos de amostragem da função cosseno utilizados na definição de suas matrizes de transformação [15, 84]. Tais matrizes de transformação, denotadas por C_N^{II} e C_N^{IV} , respectivamente, têm suas (k,n) -ésimas entradas definidas por (2.9).

Seja $\bar{\mathbf{I}}_N$ a matriz contra-identidade de ordem N definida por:

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

Em [83], Wang demonstrou que a DCT-II de comprimento 8 admite a seguinte fatoração:

$$\mathbf{C}_8^{\text{II}} = \frac{1}{2} \mathbf{P}_8 \cdot \left[\begin{array}{c|c} \mathbf{C}_4^{\text{II}} & \\ \hline \bar{\mathbf{I}}_4 \cdot \mathbf{C}_4^{\text{IV}} \cdot \bar{\mathbf{I}}_4 & \end{array} \right] \cdot \mathbf{B}_8, \quad (5.1)$$

em que \mathbf{P}_8 e \mathbf{B}_8 são matrizes de permutação e pré-adição, respectivamente, dadas por:

$$\mathbf{P}_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{B}_8 = \left[\begin{array}{c|c} \mathbf{I}_4 & \bar{\mathbf{I}}_4 \\ \hline \bar{\mathbf{I}}_4 & -\mathbf{I}_4 \end{array} \right].$$

Em [18], Chen *et al.* sugere que a matriz C_4^{IV} admite a seguinte fatoração:

$$C_4^{IV} = Q \cdot A_1 \cdot A_2 \cdot A_3, \quad (5.2)$$

em que

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} \beta_0 & 0 & 0 & \beta_3 \\ 0 & \beta_2 & \beta_1 & 0 \\ 0 & \beta_1 & -\beta_2 & 0 \\ \beta_3 & 0 & 0 & -\beta_0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \quad A_3 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & \alpha & \alpha & 0 \\ 0 & -\alpha & \alpha & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

$$\text{e } \alpha = \cos\left(\frac{\pi}{4}\right), \beta_n = \cos\left(\frac{(2n+1)\pi}{16}\right).$$

Substituindo (5.2) em (5.1) e expandindo a fatoração, obtemos

$$C_8^{II} = \frac{1}{2} \cdot P_8 \cdot M_1 \cdot M_2 \cdot M_3 \cdot M_4 \cdot B_8, \quad (5.3)$$

em que

$$M_1 = \left[\begin{array}{c|c} I_4 & \\ \hline & \bar{I}_4 \cdot Q \end{array} \right], \quad M_2 = \left[\begin{array}{c|c} P_4 & \\ \hline & A_1 \end{array} \right], \quad M_3 = \left[\begin{array}{c|c} \tilde{C} & \\ \hline & A_2 \end{array} \right],$$

$$M_4 = \left[\begin{array}{c|c} B_4 & \\ \hline & A_3 \end{array} \right], \quad P_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad B_4 = \left[\begin{array}{c|c} I_2 & \bar{I}_2 \\ \hline \bar{I}_2 & -I_2 \end{array} \right],$$

$$\tilde{C} = \left[\begin{array}{c|c} C_2^{II} & \\ \hline \bar{I}_2 \cdot C_2^{IV} \cdot \bar{I}_2 \end{array} \right] = \begin{bmatrix} \alpha & \alpha & 0 & 0 \\ \alpha & -\alpha & 0 & 0 \\ 0 & 0 & -\gamma_0 & \gamma_1 \\ 0 & 0 & \gamma_1 & \gamma_0 \end{bmatrix}$$

e $\gamma_n = \cos\left(\frac{(2n+1)\pi}{8}\right)$. A expressão (5.3) é chamada de fatoração de Chen.

As quantidades α , β_n e γ_n são irracionais presentes apenas nas matrizes M_2 , M_3 e M_4 . Tais quantidades exigem operações de multiplicação usualmente sobre aritmética de ponto flutuante.

5.3 Aproximações Propostas

Nesta seção, novas aproximações para a DCT são introduzidas. Seguindo o esquema do Capítulo 4, observamos que a fatoração (5.3) induz naturalmente o seguinte mapeamento:

$$\begin{aligned} T_C : \mathbb{R} \times \mathbb{R}^4 \times \mathbb{R}^2 &\longrightarrow \mathcal{M}_8(\mathbb{R}) \\ (\alpha, \boldsymbol{\beta}, \boldsymbol{\gamma}) &\longrightarrow \mathbf{P}_8 \cdot \mathbf{M}_1 \cdot \mathbf{M}_2(\boldsymbol{\beta}) \cdot \mathbf{M}_3(\alpha, \boldsymbol{\gamma}) \cdot \mathbf{M}_4(\alpha) \cdot \mathbf{B}_8 \end{aligned} \quad (5.4)$$

em que $\alpha \in \mathbb{R}$, $\boldsymbol{\beta} = [\beta_0 \ \beta_1 \ \beta_2 \ \beta_3]^\top \in \mathbb{R}^4$, $\boldsymbol{\gamma} = [\gamma_0 \ \gamma_1]^\top \in \mathbb{R}^2$,

$$\mathbf{M}_2(\boldsymbol{\beta}) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \beta_0 & 0 & 0 & \beta_3 \\ 0 & 0 & 0 & 0 & 0 & \beta_2 & \beta_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \beta_1 & -\beta_2 & 0 \\ 0 & 0 & 0 & 0 & \beta_3 & 0 & 0 & -\beta_0 \end{bmatrix},$$

$$\mathbf{M}_3(\alpha, \boldsymbol{\gamma}) = \begin{bmatrix} \alpha & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ \alpha & -\alpha & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\gamma_0 & \gamma_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & \gamma_1 & \gamma_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

e

$$\mathbf{M}_4(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & \alpha & \alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & -\alpha & \alpha & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Em particular, para os valores

$$\begin{aligned} \alpha &= \cos\left(\frac{\pi}{4}\right), \\ \beta_n &= \cos\left(\frac{(2n+1)\pi}{16}\right), \quad n = 0, 1, 2, 3, \\ \gamma_n &= \cos\left(\frac{(2n+1)\pi}{8}\right), \quad n = 0, 1, \end{aligned} \quad (5.5)$$

temos que $T_C(\alpha, \beta, \gamma) = 2 \cdot C_8^{\text{II}}$.

A seguir, manipularemos os valores em (5.5), visando derivar novas matrizes cujos elementos estejam definidos sobre o conjunto $\mathcal{P} = \{0, \pm \frac{1}{2}, \pm 1, \pm 2\}$. Inicialmente, consideremos as funções $\text{sign}(\cdot)$ e $\text{round}(\cdot)$ como definidas em (1.2) e (3.5), respectivamente. As funções $\text{sign}(\cdot)$ e $\text{round}(\cdot)$, quando operadas sobre matrizes, operam elemento a elemento.

Desse modo, considerando os valores em (5.5) e aplicando diretamente as funções acima, obtemos os vetores:

$$\begin{aligned} (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}) &= \text{sign}[(\alpha, \beta, \gamma)] = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]^\top, \\ (\hat{\alpha}, \hat{\beta}, \hat{\gamma}) &= \text{round}[(\alpha, \beta, \gamma)] = [1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0]^\top. \end{aligned}$$

Assim, definimos as seguintes matrizes aproximadas:

$$\begin{aligned} \tilde{\mathbf{M}}_2 &= \mathbf{M}_2(\tilde{\beta}), & \hat{\mathbf{M}}_2 &= \mathbf{M}_2(\hat{\beta}), \\ \tilde{\mathbf{M}}_3 &= \mathbf{M}_3(\tilde{\alpha}, \tilde{\gamma}), & \hat{\mathbf{M}}_3 &= \mathbf{M}_3(\hat{\alpha}, \hat{\gamma}), \\ \tilde{\mathbf{M}}_4 &= \mathbf{M}_4(\tilde{\alpha}), & \hat{\mathbf{M}}_4 &= \mathbf{M}_4(\hat{\alpha}). \end{aligned}$$

Considerando as definições destas matrizes, obtemos:

$$\begin{aligned} \widetilde{\mathbf{M}}_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}, & \widehat{\mathbf{M}}_2 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \\ \widetilde{\mathbf{M}}_3 &= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, & \widehat{\mathbf{M}}_3 &= \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, \\ \widetilde{\mathbf{M}}_4 &= \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \widehat{\mathbf{M}}_4. \end{aligned}$$

Consequentemente, invocando a fatoração (5.4), definimos as transformadas:

$$\widetilde{\mathbf{T}}_8 = \mathbf{P}_8 \cdot \mathbf{M}_1 \cdot \widetilde{\mathbf{M}}_2 \cdot \widetilde{\mathbf{M}}_3 \cdot \widetilde{\mathbf{M}}_4 \cdot \mathbf{B}_8, \quad (5.6)$$

$$\widehat{\mathbf{T}}_8 = \mathbf{P}_8 \cdot \mathbf{M}_1 \cdot \widehat{\mathbf{M}}_2 \cdot \widehat{\mathbf{M}}_3 \cdot \widehat{\mathbf{M}}_4 \cdot \mathbf{B}_8. \quad (5.7)$$

As matrizes nas fatorações (5.6) e (5.7) têm inversas simples dadas por:

$$\begin{aligned} \mathbf{P}_8^{-1} &= \mathbf{P}_8^\top, \quad \mathbf{M}_1^{-1} = \mathbf{M}_1^\top, \quad \widetilde{\mathbf{M}}_3^{-1} = \frac{1}{2} \cdot \widetilde{\mathbf{M}}_3, \quad \mathbf{B}_8^{-1} = \frac{1}{2} \cdot \mathbf{B}_8^\top, \\ \widetilde{\mathbf{M}}_2^{-1} &= \frac{1}{2} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}, & \widehat{\mathbf{M}}_2^{-1} &= \frac{1}{2} \cdot \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix}, \\ \widetilde{\mathbf{M}}_3^{-1} &= \frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}, & \widehat{\mathbf{M}}_4^{-1} &= \frac{1}{2} \cdot \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \end{bmatrix} = \widehat{\mathbf{M}}_4^{-1}. \end{aligned}$$

Assim, as transformadas dadas por (5.6) e (5.7), têm inversas:

$$\tilde{\mathbf{T}}_8^{-1} = \frac{1}{4} \cdot \mathbf{B}_8^\top \cdot \tilde{\mathbf{M}}_4^{-1} \cdot \tilde{\mathbf{M}}_3 \cdot \tilde{\mathbf{M}}_2^{-1} \cdot \mathbf{M}_1^\top \cdot \mathbf{P}_8^\top, \quad (5.8)$$

$$\hat{\mathbf{T}}_8^{-1} = \frac{1}{2} \cdot \mathbf{B}_8^\top \cdot \hat{\mathbf{M}}_4^{-1} \cdot \hat{\mathbf{M}}_3^{-1} \cdot \hat{\mathbf{M}}_2^{-1} \cdot \mathbf{M}_1^\top \cdot \mathbf{P}_8^\top. \quad (5.9)$$

Temos que, por construção, os elementos das matrizes envolvidas nas fatorações (5.6), (5.7), (5.8) e (5.9) pertencem ao conjunto \mathcal{P} . Dado que as transformadas $\tilde{\mathbf{T}}_8$ e $\hat{\mathbf{T}}_8$, bem como suas inversas, são implementadas através das respectivas fatorações, suas implementações são livres de multiplicações.

Observar que as transformadas propostas são não-ortogonais, tais como a SDCT [32] e a BAS [9]. Considerando (4.15), os valores de desvio da diagonalidade das novas transformadas são

$$\delta(\tilde{\mathbf{T}}_8 \cdot \tilde{\mathbf{T}}_8^\top) = 0.0714,$$

$$\delta(\hat{\mathbf{T}}_8 \cdot \hat{\mathbf{T}}_8^\top) = 0.0299.$$

Dado que o desvio da diagonalidade da SDCT é de 0.20, aceitamos $\tilde{\mathbf{T}}_8$ e $\hat{\mathbf{T}}_8$ como transformadas que fornecem aproximações quasi-ortogonais para a DCT dadas por (4.14):

$$\tilde{\mathbf{C}}_8 = \tilde{\mathbf{S}} \cdot \tilde{\mathbf{T}}_8,$$

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

em que

$$\tilde{\mathbf{S}} = \text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}} \right),$$

$$\hat{\mathbf{S}} = \text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{12}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{12}}, \frac{1}{2}, \frac{1}{\sqrt{6}} \right),$$

são dadas por (4.13). Um fato importante no contexto de compressão de imagens é que as matrizes $\tilde{\mathbf{S}}$ e $\hat{\mathbf{S}}$, assim como os escalares nas equações (5.8) e (5.9), podem ser absorvidas na etapa de quantização [5, 20] (cf. apêndice B). Assim, nesse sentido, não representam aumento na complexidade computacional.

Calculando as fatorações (5.6) e (5.7), obtemos explicitamente as transformadas:

$$\tilde{\mathbf{T}}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 0 & 1 & -1 & 0 & -2 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 2 & -1 & 1 & -2 & 0 & -1 \end{bmatrix}, \quad \hat{\mathbf{T}}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -2 & -1 & 1 & 2 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}.$$

A Figura 5.1 apresenta o diagrama de fluxo de sinal do algoritmo rápido proposto para $\hat{\mathbf{T}}_8$ e sua inversa. O diagrama de fluxo de sinal para $\tilde{\mathbf{T}}_8$ é similar e portanto não é apresentado.

5.4 Medida de Proximidade e Custo Computacional

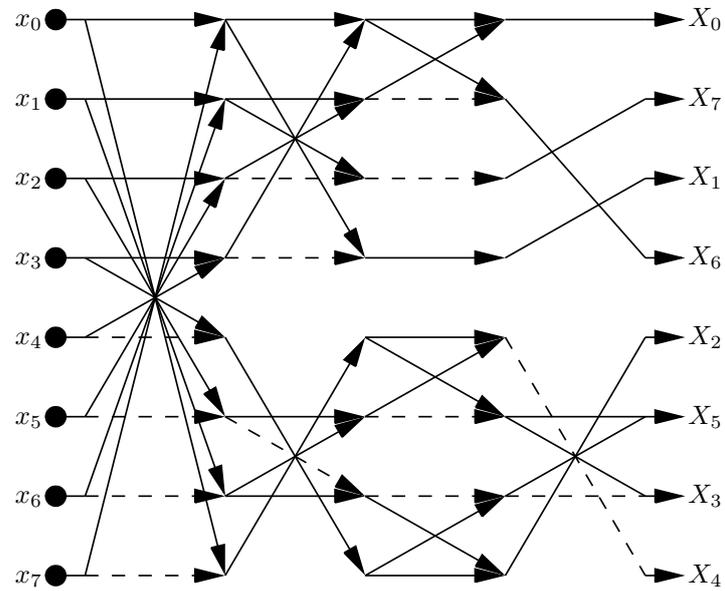
Com o objetivo de mensurar a proximidade das novas aproximações propostas com relação a DCT exata, utilizamos o erro de energia total como critério (cf. (4.17)). Como descrito em [20], esta é uma medida de similaridade entre a DCT exata e a aproximação considerada. Para comparação com as transformadas propostas, consideramos as aproximações SDCT e BAS—que são não-ortogonais, assim como as transformadas HT e WHT, ambas ortogonais.

Os resultados para a medida de erro de energia total para as transformadas discutidas estão listados na Tabela 5.1. Adicionalmente, na Tabela 5.2, são fornecidas as complexidades computacionais de cada método considerado neste capítulo.

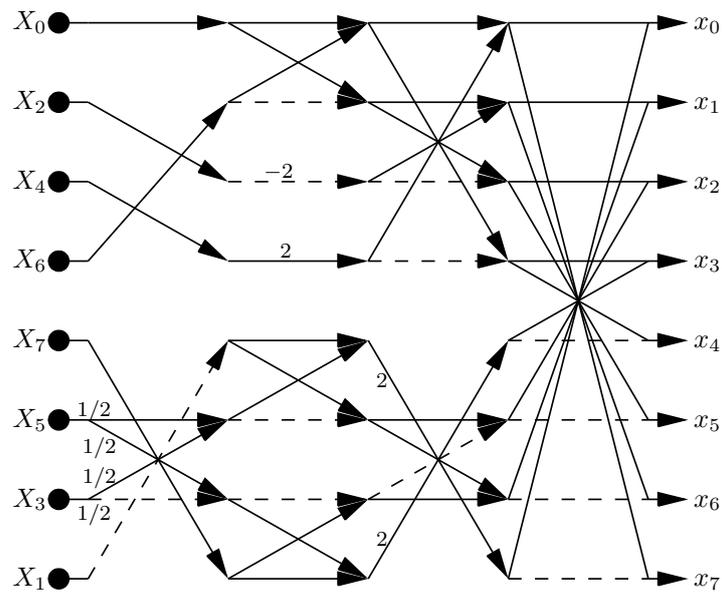
Tabela 5.1: *Erro de energia total das transformadas consideradas*

	$\hat{\mathbf{T}}_8$	$\tilde{\mathbf{T}}_8$	SDCT	BAS	WHT	HT
Erro de energia total	1.79	3.64	3.32	4.12	5.05	47.61

A transformada proposta $\hat{\mathbf{T}}_8$ apresenta o menor valor de erro de energia total entre as transformadas consideradas e requer apenas 22 adições, i.e., duas operações a menos que a SDCT e apenas uma a mais que a BAS. A Tabela 5.2 também apresenta o custo computacional da DCT exata utilizando a fatoração de Chen descrita em (5.3), necessitando 26 adições e 16 multiplicações para sua implementação.



(a) \hat{T}_8



(b) \hat{T}_8^{-1}

Figura 5.1: Diagramas de fluxo de sinal. As linhas em traços indicam multiplicação por -1 .

Tabela 5.2: Custo computacional

	Adições	Mult.	Deslocamentos	Total
DCT	26	16	0	42
$\hat{\mathbf{T}}_8$	22	0	0	22
$\tilde{\mathbf{T}}_8$	26	0	0	26
SDCT	24	0	0	24
BAS	21	0	0	21
WHT	24	0	0	24
HT	24	0	0	24

O custo computacional da transformada $\tilde{\mathbf{T}}_8$ é superior ao custo computacional das outras transformadas analisadas. Com isso, no restante do capítulo damos ênfase à transformada proposta $\hat{\mathbf{T}}_8$.

5.5 Aplicações em Compressão de Imagens

5.5.1 Compressão tipo JPEG e Medidas de Erro

As transformadas $\tilde{\mathbf{T}}_8$ e $\hat{\mathbf{T}}_8$ propostas foram avaliadas no contexto de compressão de imagens, conjuntamente com as aproximações SDCT, BAS, HT e WHT. Para tanto, foi usado um conjunto de 45 imagens padronizadas de 8-bits em escala de cinzas de tamanho 512×512 obtidas de [79]. Cada imagem foi subdividida em blocos de tamanho 8×8 . Cada bloco foi submetido à transformada 2-D como discutido no Apêndice B. Os 64 coeficientes resultantes para cada bloco foram ordenados segundo a sequência zig-zag padrão [81] e apenas r coeficientes foram retidos pelo processo de compressão. Como sugerido em [20], adotamos $1 \leq r \leq 45$. Finalmente, a transformada 2-D inversa é aplicada para reconstruir a imagem compactada. A imagem resultante é então comparada com a imagem original segundo medidas objetivas de qualidade.

Como medidas de avaliação foram consideradas o PSNR [78] e o SSIM [87]. A medida de PSNR é comumente a mais utilizada para comparar técnicas de compressão. Contudo, como apontado em [87], o SSIM considera características do sistema visual humano não consideradas no PSNR. Para melhorar a visualização das curvas

de desempenho, também foi calculado o APE relativo à DCT exata destas medidas, como descrito na Seção 4.8.1. Seguindo a metodologia de [20], a média das medidas de avaliação sob as 45 imagens padronizadas foram calculadas, o que resulta em resultados estatisticamente mais robustos, quando comparados com análises feitas em imagens particulares.

5.5.2 Resultados

Os valores resultantes dos experimentos computacionais são apresentados nas Figuras 5.2(a)-(b). Na Figura 5.2(d), as curvas correspondentes à HT e SDCT foram suprimidas, pois apresentam valores excessivamente altos em comparação às outras. Os resultados mostram que, em termos de PSNR, a aproximação \hat{T}_8 supera a SDCT e a BAS, e proporciona resultados similares à WHT, mas com um menor custo computacional. Em termos de SSIM, temos que as duas aproximações propostas \tilde{T}_8 e \hat{T}_8 tem desempenhos similares.

A aproximação proposta \hat{T}_8 tem desempenho bastante próximo ao da WHT. Conforme as Figuras 5.2(a) e (c), o desempenho da transformada \hat{T}_8 é superior ao desempenho da WHT para r pequeno ($r \leq 15$). Assim, na Figura 5.3, é apresentada a imagem padrão *Elaine* após o processo de compressão considerando todas as transformadas apresentadas. Todas as imagens foram comprimidas utilizando $r = 6$, valor que representa um truncamento de aproximadamente 90.5% dos coeficientes. A análise visual dessas imagens obtidas mostra a superioridade da transformada \hat{T}_8 proposta sobre a SDCT em compressão de imagens. Ainda, a Tabela 5.3 evidencia quantitativamente o melhor desempenho de \hat{T}_8 sobre as demais transformadas consideradas.

5.5.3 Análise de artefatos de blocagens (*blocking artifact*)

Um efeito visual indesejável em compressão de imagens é o surgimento de blocos na imagem. Esse efeito é conhecido como artefatos de blocagens [30, p. 573].

A Figura 5.4 mostra qualitativamente essa comparação em termos de artefatos de blocagens derivados da \hat{T}_8 e WHT. A utilização de \hat{T}_8 na compressão de imagens minimiza o aparecimento de artefatos de blocagens comparativamente à WHT.

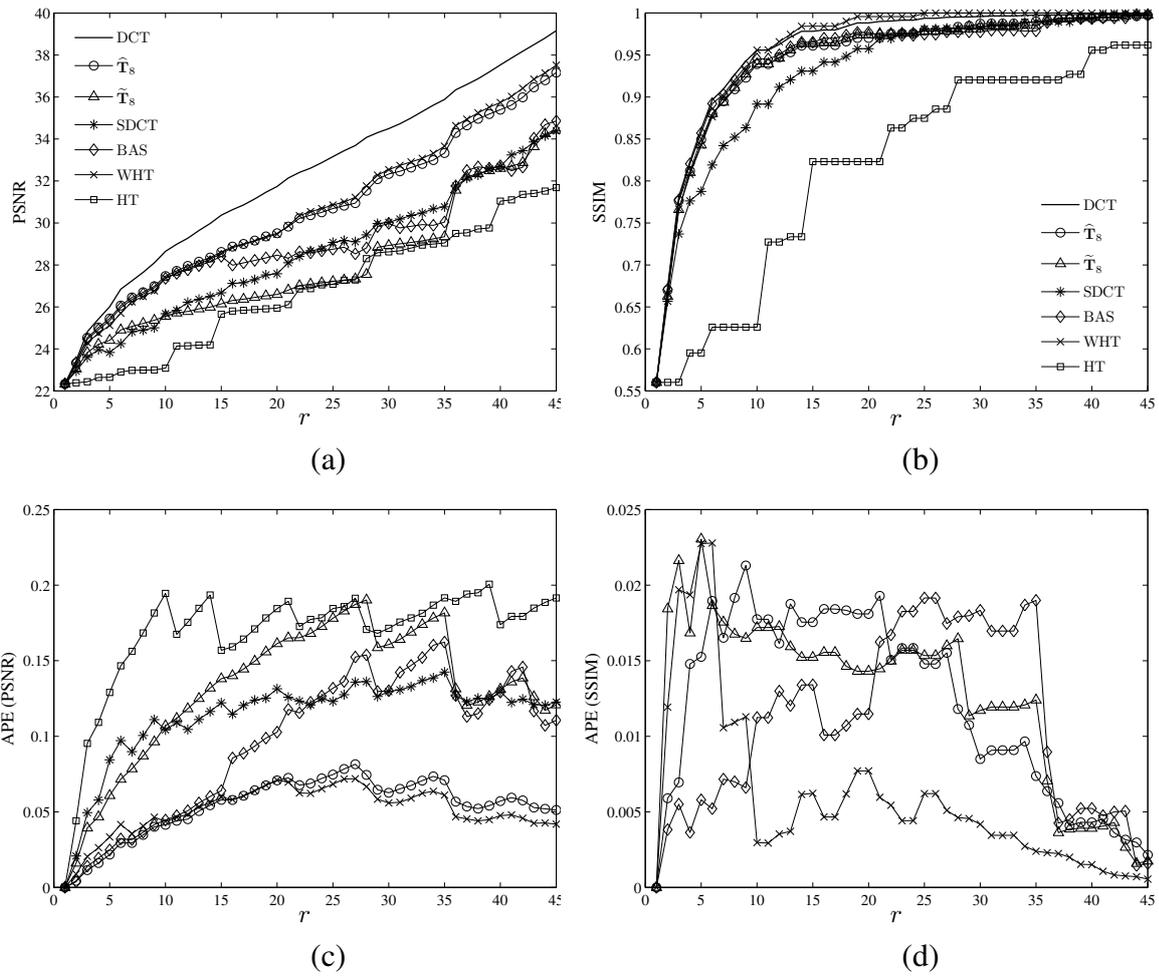


Figura 5.2: Curvas para a média correspondente a 45 imagens. (a) PSNR, (b) SSIM, (c) APE (PSNR), e (d) APE (SSIM).

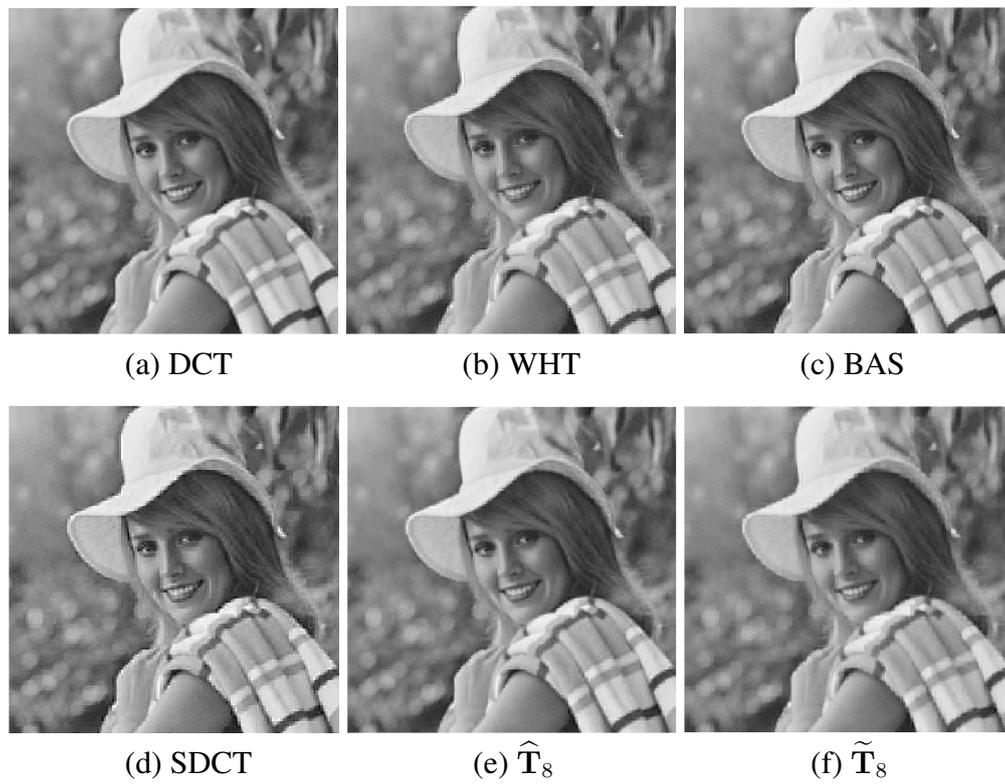
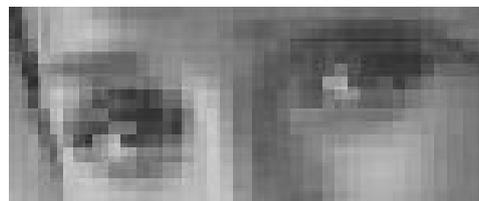


Figura 5.3: Imagem compactada *Elaine* para $r = 6$.



(a) WHT



(b) \hat{T}_8

Figura 5.4: Efeito de artefatos de blocagens na imagem *Elaine* para $r = 6$.

Tabela 5.3: Medidas de avaliação de compressão da imagem *Elaine*, para $r = 6$

Transformada	PSNR	SSIM
DCT	31.03	0.95
$\hat{\mathbf{T}}_8$	30.00	0.94
$\tilde{\mathbf{T}}_8$	30.00	0.94
SDCT	27.59	0.88
WHT	28.91	0.92
HT	28.91	0.92

5.6 Conclusão

O presente capítulo apresenta duas aproximações para a DCT baseadas na fatoração de Chen. Seus desempenhos foram comparados com outras aproximações conhecidas na literatura de processamento de sinais. A aproximação $\hat{\mathbf{T}}_8$ apresentou o menor erro de energia total dentre todas as aproximações analisadas, evidenciando uma maior similaridade espectral com a DCT exata. A aproximação $\hat{\mathbf{T}}_8$ também demonstrou superioridade sobre a SDCT, a BAS e a HT em termos de PSNR das imagens comprimidas, tendo um desempenho similar à WHT, porém com um custo computacional 8% inferior.

Capítulo 6

Conclusões finais

Neste trabalho, foram obtidas diferentes aproximações de baixa complexidade para a DCT de comprimento 8 para serem aplicadas principalmente no contexto de compressão de imagens. A obtenção destas transformadas basou-se na parametrização de fatorações para a DCT já existentes. Este método proporciona uma generalização dos métodos existentes para a obtenção de aproximações da DCT. As aplicações destas transformadas de baixa complexidade é variada e podem ser consideradas em implementações que requerem de baixo custo de hardware e baixa demanda de potência, tais como transmissão e processamento de vídeo em tempo real e aplicações de cômputo portátil, entre outros usos.

Nos Capítulos 1, 2 e 3, foram apresentados uma introdução e as considerações teóricas da DCT, assim como a apresentação das aproximações de baixa complexidade para a DCT já listadas na literatura. Estes capítulos apresentam uma revisão geral da literatura e motivam a inclusão e o uso de aproximações de baixa complexidade no contexto de processamento de imagens.

As contribuições deste trabalho são apresentadas nos Capítulos 4 e 5. No Capítulo 4, uma nova classe de transformadas de baixa complexidade computacional é proposta. Estas aproximações são obtidas ao considerar uma parametrização da fatoração de Feig-Winograd para a DCT de comprimento 8. Este método induz um subespaço matricial, que referenciamos como o subespaço de Feig-Winograd. Dado que este subespaço contém centenas de transformadas, foi proposto um problema de otimização multicritério com o objetivo de vislumbrar aquelas transformadas que apresentam um

desempenho ótimo em termos de medidas de avaliação tais como erro de energia, erro quadrático médio, ganho em codificação e eficiência da transformada. Os critérios usados para a obtenção destas aproximações ótimas foram: (i) baixa complexidade computacional, (ii) ortogonalidade ou quase-ortogonalidade, (iii) baixa complexidade da transformada inversa e (iv) proximidade com a DCT exata. As transformadas ótimas obtidas foram comparadas com outras aproximações já listadas na literatura que não pertencem ao subespaço de Feig-Winograd. Um fato importante para destacar é que todas as transformadas na classe Feig-Winograd possuem a mesma estrutura do algoritmo rápido e podem ser implementadas com modificações mínimas de hardware.

No Capítulo 5, foram propostas duas novas aproximações de baixa complexidade computacional para a DCT de comprimento 8 baseadas na fatoração de Chen. Estas novas transformadas foram avaliadas em termos de erro de energia com outras aproximações para a DCT já conhecidas na literatura. Um fato para destacar é que, embora estas transformadas são não-ortogonais, suas transformadas inversas mantêm a propriedade de baixa complexidade computacional atingidas pelas transformadas diretas.

Todas as transformadas obtidas foram avaliadas e comparadas, no contexto de compressão de imagens, com outras transformadas em termos de PSNR e SSIM. As curvas destas medidas proporcionam valores que permitem comparar quantitativamente a qualidade de compressão das transformadas propostas. Em todos os casos, os resultados obtidos evidenciam que as transformadas obtidas são adequadas para aplicações em compressão de imagens.

A seguir, são listadas as principais contribuições deste trabalho:

- Nos anos recentes, várias aproximações de baixa complexidade para a DCT de comprimento 8 foram propostas [5, 7, 8, 9, 10, 12, 13, 20, 32, 54, 62]. Este trabalho fornece uma revisão, análise e comparação de todas estas aproximações, o qual resulta útil para o leitor interessado;
- Como foi comentado na Seção 1.3, as aproximações de baixo custo computacional listadas na literatura são obtidas principalmente através de: (i) aproximações brutas [20, 32], (ii) inspeção [12], (iii) variações de aproximações prévias [5, 7, 8, 9, 10, 13] e (iv) procedimentos de otimização simples baseados na estrutura da DCT [62]. Este trabalho propõe um formalismo matemático unificador para derivar aproximações de baixa complexidade computacional, gene-

realizando os métodos já existentes na literatura. Assim, são identificadas várias aproximações já conhecidas como casos particulares da estrutura geral introduzida no Capítulo 4 (cf. Seção 4.5);

- No Capítulo 4, é proposto um problema de otimização multicritério, com o objetivo de identificar aproximações ótimas para a DCT de comprimento 8. Assim, foram usadas medidas de avaliação tais como erro de energia total, erro quadrático médio, ganho em codificação unificado e eficiência da transformada, as quais foram consideradas como funções objetivos do problema de otimização. Portanto, remarcamos a importância de considerar várias medidas como critério de avaliação. A diferença de outros trabalhos clássicos no tema, este trabalho considera uma tendência multicritério para obter as novas aproximações;
- Este trabalho propõe o conceito de subespaço matricial gerado pela parametrização de um algoritmo rápido para a DCT, como uma ferramenta para a obtenção de aproximações de baixa complexidade computacional. Esta tendência é nova no contexto de aproximações para a DCT. Assim, é possível introduzir uma família completa de aproximações de baixíssima complexidade computacional. Esta tendência também permite obter automaticamente os algoritmos rápidos para as aproximações contidas nestes subespaços;
- Este trabalho introduz 8 novas aproximações de baixa complexidade computacional para a DCT: 6 transformadas estruturalmente novas no Capítulo 4 e 2 novas transformadas no Capítulo 5. Destas, 5 são ortogonais e 3 são não-ortogonais. Estas transformadas foram avaliadas e comparadas com outras transformadas listadas na literatura, resultando ser boas aproximações para serem usadas no contexto de compressão de imagens.

Estas contribuições foram tema de pesquisa nos seguintes trabalhos:

- i) C. J. Tablada, F. M. Bayer e R. J. Cintra. Duas Aproximações para a DCT Baseadas na Fatoração de Chen. *XXXI Simpósio Brasileiro de Telecomunicações (SBrT)*, 2013.
- ii) R. S. Oliveira, R. J. Cintra, F. M. Bayer e C. J. Tablada. Uma Aproximação Ortogonal para a DCT. *XXXI Simpósio Brasileiro de Telecomunicações (SBrT)*, 2013.

- iii) F. M. Bayer, C. J. Tablada and R. J. Cintra. A Class of DCT Approximations Based on the Feig-Winograd Algorithm. Artigo submetido em *IEEE Transactions on Image Processing*, 2014.
- iv) R. J. Cintra, F. M. Bayer and C. J. Tablada. Low-complexity 8-point DCT Approximations based on Integer Functions. *Signal Processing*, 2014.

Como contribuições futuras e extensões deste trabalho, podemos listar as seguintes:

1. Seguindo a tendência do Capítulo 4, é possível considerar uma ampliação do método apresentado no Capítulo 5 ao considerar uma parametrização da fatoração de Chen, com o objetivo de obter um subespaço matricial (subespaço de Chen). Assim, pode-se considerar um problema de otimização multicritério que permita vislumbrar, neste subespaço, as aproximações ótimas segundo as medidas de avaliação consideradas;
2. Observar que, além das fatorações de Feig-Winograd e Chen, é possível estender o método apresentado neste trabalho ao considerar outras fatorações para a DCT de comprimento 8, como aquelas apresentadas na Seção 2.5. Assim, cada fatoração pode induzir um subespaço matricial que permita obter boas aproximações para a DCT segundo medidas de avaliação a ser consideradas;
3. Além da DCT de comprimento 8, transformadas de outras ordens são importantes no contexto de compressão de imagens. Por exemplo, a DCT de comprimento 16 é usada no recente padrão de compressão HEVC [63]. Este padrão faz uso da fatoração de Chen para a implementação da DCT de comprimento 16. Como é mostrado em Chen *et al.* [18], a fatoração para a DCT de ordem 16 envolve a DCT de comprimento 8. Portanto, é plausível considerar uma aproximação para a DCT de ordem 16 ao considerar uma aproximação para a DCT de ordem 8. Outra tendência é considerar um subespaço matricial ao considerar uma parametrização da fatoração de Chen para a DCT de ordem 16, similar à tendência apresentada no Capítulo 4. Assim, a substituição da DCT de ordem 16 por uma aproximação no padrão HEVC pode conduzir a um padrão “HEVC aproximado” que proporcione um bom balanço entre desempenho e custo computacional.

Naturalmente, a lista apresentada pode ser estendida ao considerar: (i) aproximações para a DCT de outros comprimentos, (ii) fatorações para a DCT que não são apresentadas neste trabalho e (iii) outras aplicações além de compressão de imagens, como por exemplo criptografia [11, 17].

Referências

- [1] Digital compression and coding of continuous-tone still images. Part 1, requirements and guidelines. Technical report, ISO/IEC JTC1 Committee Draft 10918-2, 1991.
- [2] AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transform. *IEEE Transactions on Computers*, C-23(1):90–93, 1974.
- [3] AMIRA, A.; BOURIDANE, A.; MILLIGAN, P. An FPGA based Walsh-Hadamard transforms. In *IEEE International Symposium on Circuits and Systems, 2001. ISCAS 2001*, volume 2, pages 569–572, 2001.
- [4] ARAI, Y.; AGUI, T.; NAKAJIMA, M. A fast DCT-SQ scheme for images. *Transactions of the IEICE*, E-71(11):1095–1097, 1988.
- [5] BAYER, F. M.; CINTRA, R. J. DCT-like transform for image compression requires 14 additions only. *Electronics Letters*, 48(15):919–921, 2012.
- [6] BAYER, F. M. et al. A digital hardware fast algorithm and FPGA-based prototype for a novel 16-point approximate DCT for image compression applications. *Measurement Science and Technology*, 23(8):114010, 2012.
- [7] BOUGHEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. Binary discrete cosine and Hartley transforms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 60(4):989–1002, 2013.

-
- [8] BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. Low-complexity 8×8 transform for image compression. *Electronics Letters*, 44(21):1249–1250, 2008.
- [9] BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. A multiplication-free transform for image compression. In *2nd International Conference on Signals, Circuits and Systems (SCS)*, pages 1–4, 2008.
- [10] BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. A fast 8×8 transform for image compression. In *2009 International Conference on Microelectronics (ICM)*, pages 74–77, 2009.
- [11] BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. Image encryption using the reciprocal-orthogonal parametric transform. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2542–2545, 2010.
- [12] BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. A novel transform for image compression. In *53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 509–512, 2010.
- [13] BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. N. S. A low-complexity parametric transform for image compression. In *Proceedings of the 2011 IEEE International Symposium on Circuits and Systems*, 2011.
- [14] BRIGGS, W. L.; HENSON, V. E. *The DFT: An Owners' Manual for the Discrete Fourier Transform*. Miscellaneous Bks. Society for Industrial and Applied Mathematics, 1995.
- [15] BRITANAK, V.; YIP, P.; RAO, K. R. *Discrete Cosine and Sine Transforms*. Academic Press, 2007.
- [16] CHAM, W. K. Development of integer cosine transforms by the principle of dyadic symmetry. In *IEE Proceedings I Communications, Speech and Vision*, volume 136, pages 276–282, 1989.
- [17] CHEN, L.; ZHAO, D. Optical image encryption with Hartley transforms. *Optics Letters*, 31(23):3438–3440, 2006.

-
- [18] CHEN, W. H.; SMITH, C.; FRALICK, S. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on Communications*, 25(9):1004–1009, 1977.
- [19] CINTRA, R. J. An integer approximation method for discrete sinusoidal transforms. *Journal of Circuits, Systems, and Signal Processing*, 30(6):1481–1501, 2011.
- [20] CINTRA, R. J.; BAYER, F. M. A DCT approximation for image compression. *IEEE Signal Processing Letters*, 18(10):579–582, 2011.
- [21] CLARKE, R. J. Relation between the Karhunen-Loève and cosine transforms. *IEEE Proceedings F Communications, Radar and Signal Processing*, 128(6):359–360, 1981.
- [22] DIMITROV, V.S.; JULLIEN, G.A.; MILLER, W.C. A new DCT algorithm based on encoding algebraic integers. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, 1998.*, volume 3, pages 1377–1380 vol.3, 1998.
- [23] EDIRISURIYA, A. et al. 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*, 2(2):127–142, 2012.
- [24] EHRGOTT, M. *Multicriteria Optimization*. Lecture Notes in Economics and Mathematical Systems. Springer-Verlag GmbH, 2000.
- [25] ELLIOT, D. F.; RAO, K. R. *Fast Transforms: Algorithms, Analyses, Applications*. Computer Science and Applied Mathematics Series. Academic Press, 1982.
- [26] FEIG, E.; WINOGRAD, S. Fast algorithms for the discrete cosine transform. *IEEE Transactions on Signal Processing*, 40(9):2174–2193, 1992.
- [27] FEIG, E.; WINOGRAD, S. On the multiplicative complexity of discrete cosine transforms. *IEEE Transactions on Information Theory*, 38(4):1387–1391, 1992.

-
- [28] FLURY, B.; GAUTSCHI, W. An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. *SIAM Journal on Scientific and Statistical Computing*, 7(1):169–184, 1986.
- [29] GALASSI, M. *GNU Scientific Library : reference manual for GSL version 1.12*. Network Theory, 2009.
- [30] GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 3rd edition, 2006.
- [31] GOYAL, V. K. Theoretical foundations of transform coding. *IEEE Signal Processing Magazine*, 18(5):9–21, 2001.
- [32] HAWHEEL, T. I. A new square wave transform based on the DCT. *Signal Processing*, 82:2309–2319, 2001.
- [33] HEIDEMAN, M. T.; BURRUS, C. S. *Multiplicative complexity, convolution, and the DFT*. Signal Processing and Digital Filtering. Springer-Verlag, 1988.
- [34] HIGHAM, N. J. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986.
- [35] HIGHAM, N. J. Computing real square roots of a real matrix. *Linear Algebra and its Applications*, 88–89:405–430, 1987.
- [36] HIGHAM, N. J. *Functions of Matrices: Theory and Computation*. SIAM e-books. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2008.
- [37] HIGHAM, N. J. et al. Computing the polar decomposition and the matrix sign decomposition in matrix groups. *SIAM J. Matrix Analysis Applications*, 25(4):1178–1192, 2004.
- [38] HIGHAM, N. J.; SCHREIBER, R. S. Fast polar decomposition of an arbitrary matrix. Technical report, Ithaca, NY, USA, 1988.
- [39] HORADAM, K. J. *Hadamard Matrices and Their Applications*. Princeton University Press, 2007.

-
- [40] HOU, H. S. A fast recursive algorithm for computing the discrete cosine transform. *IEEE Transactions on Acoustic, Signal, and Speech Processing*, 6(10):1455–1461, 1987.
- [41] HUFFMAN, D. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.
- [42] HUNGERFORD, T. W. *Algebra*. Graduate Texts in Mathematics. Springer, 1974.
- [43] International Organisation for Standardisation. Generic coding of moving pictures and associated audio information – Part 2: Video. ISO/IEC JTC1/SC29/WG11 - coding of moving pictures and audio, ISO, 1994.
- [44] International Telecommunication Union. ITU-T recommendation H.261 version 1: Video codec for audiovisual services at $p \times 64$ kbits. Technical report, ITU-T, 1990.
- [45] International Telecommunication Union. ITU-T recommendation H.263 version 1: Video coding for low bit rate communication. Technical report, ITU-T, 1995.
- [46] JAIN, A. K. A sinusoidal family of unitary transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(4):356–365, 1979.
- [47] JAIN, A. K. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, N. J., USA, 1989.
- [48] KATTO, J.; YASUDA, Y. Performance evaluation of subband coding and optimization of its filter coefficients. *Journal of Visual Communication and Image Representation*, 2(4):303–313, 1991.
- [49] KAY, S. M. *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, volume 1 of *Prentice Hall Signal Processing Series*. Prentice Hall, Upper Saddle River, NJ, 1993.
- [50] KIM, B.; ZIAVRAS, S. G. Low-power multiplierless DCT for image/video coders. In *IEEE 13th International Symposium on Consumer Electronics, 2009. ISCE '09*, pages 133–136, 2009.

-
- [51] KUO, W. K.; WU, K. W. Traffic prediction and QoS transmission of real-time live VBR videos in WLANs. *ACM Transactions on Multimedia Computing, Communications and Applications*, 7(4):36:1–36:21, 2011.
- [52] LECUIRE, V.; MAKKAOUI, L.; MOUREAUX, J. M. Fast zonal DCT for energy conservation in wireless image sensor networks. *Electronics Letters*, 48(2):125–127, 2012.
- [53] LEE, B. G. A new algorithm for computing the discrete cosine transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-32:1243–1245, 1984.
- [54] LENGWEHASATIT, K.; ORTEGA, A. Scalable variable complexity approximate forward DCT. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(11):1236–1248, 2004.
- [55] LIANG, J.; TRAN, T. D. Fast multiplierless approximation of the DCT with the lifting scheme. *IEEE Transactions on Signal Processing*, 49:3032–3044, 2001.
- [56] LOEFFLER, C.; LIGTENBERG, A.; MOSCHYTZ, G. Practical fast 1D DCT algorithms with 11 multiplications. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, pages 988–991, 1989.
- [57] MADANAYAKE, A. et al. A single-channel architecture for algebraic integer based 8×8 2-D DCT computation. *IEEE Transactions on Circuits and Systems for Video Technology*, PP(99):1–1, 2013.
- [58] MATLAB. *version 8.1 (R2013a) Documentation*. The MathWorks Inc., Natick, Massachusetts, 2013.
- [59] MIETTINEN, K. *Nonlinear Multiobjective Optimization*. International series in operations research and management science. Kluwer Academic Publishers, 1999.
- [60] OLIVEIRA, R. S.; CINTRA, R. J. Uma fatoração para SDCT inversa. Trabalho de IC da UFPE, 2013.

-
- [61] PENNEBAKER, W. B.; MITCHELL, J. L. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, NY, 1992.
- [62] POTLURI, U. S. et al. Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing. *Measurement Science and Technology*, 23(11):114003, 2012.
- [63] POURAZAD, M. T. et al. HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC? *IEEE Consumer Electronics Magazine*, 1(3):36–46, 2012.
- [64] PRATT, W. K. *Digital Image Processing: PIKS Inside*. John Wiley & Sons, Inc., New York, NY, USA, 3rd edition, 2001.
- [65] RAO, K. R.; YIP, P. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, San Diego, CA, 1990.
- [66] RAY, W.; DRIVER, R. Further decomposition of the Karhunen-Loève series representation of a stationary random process. *IEEE Transactions on Information Theory*, 16(6):663–668, 1970.
- [67] ROMA, N.; SOUSA, L. Efficient hybrid DCT-domain algorithm for video spatial downscaling. *EURASIP Journal on Advances in Signal Processing*, 2007(2):30–30, 2007.
- [68] SAPONARA, S. Real-time and low-power processing of 3D direct/inverse discrete cosine transform for low-complexity video codec. *Journal of Real-Time Image Processing*, 7:43–53, 2012.
- [69] SEBER, G. A. F. *A Matrix Handbook for Statisticians*. Wiley Series in Probability and Mathematical Statistics. Wiley, 2008.
- [70] SILVA, A.; NAVARRO, A. Fast 8×8 DCT pruning algorithm. In *IEEE International Conference on Image Processing, 2005. ICIP 2005.*, volume 2, pages II–317–20, 2005.
- [71] STRANG, G. *Linear Algebra and Its Applications*. Brooks Cole, 1988.

-
- [72] STRANG, G. The discrete cosine transform. *SIAM Rev.*, 41(1):135–147, 1999.
- [73] STRANG, G. *Computational Science and Engineering*. Wellesley-Cambridge Press, 2007.
- [74] STRANG, G. *Calculus*. Wellesley-Cambridge Press, 2010.
- [75] STRANG, G.; NGUYEN, T. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
- [76] SUEHIRO, N.; HATORI, M. Fast algorithms for the DFT and other sinusoidal transforms. *IEEE Transactions on Acoustic, Signal, and Speech Processing*, 34(6):642–644, 1986.
- [77] TALLIA, R.; MORELLO, P.; CASTELLANO, G. The Walsh-Hadamard transform: an alternative means of obtaining phase and amplitude maps. *J Nucl Med*, 25(5):608–12, 1984.
- [78] THU, Q. H.; GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.
- [79] The USC-SIPI image database. <http://sipi.usc.edu/database/>, 2011. University of Southern California, Signal and Image Processing Institute.
- [80] VETTERLI, M.; NUSSBAUMER, H. Simple FFT and DCT algorithms with reduced number of operations. *Signal Processing*, 6:267–278, 1984.
- [81] WALLACE, G. K. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.
- [82] WANG, C.; HOU, Z.; YANG, A. Image coding algorithm based on all phase Walsh biorthogonal transform. In *International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, 2007*, pages 1118–1123, 2007.
- [83] WANG, Z. Reconsideration of: A fast computational algorithm for the Discrete Cosine Transform. *IEEE Transactions on Communications*, 31(1):121–123, 1983.

-
- [84] WANG, Z. Fast algorithms for the discrete W transform and for the discrete Fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-32:803–816, 1984.
- [85] WANG, Z.; BOVIK, A. C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.
- [86] WANG, Z.; BOVIK, A. C. Reduced- and no-reference image quality assessment. *IEEE Signal Processing Magazine*, 28(6):29–40, 2011.
- [87] WANG, Z. et al. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- [88] WANG, Z.; HUNT, B. The discrete cosine transform—A new version. In *Proc. IEEE International Conference on ICASSP '83. Acoustics, Speech, and Signal Processing*, volume 8, pages 1256–1259, 1983.
- [89] WATKINS, D. S. *Fundamentals of Matrix Computations*. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2004.
- [90] WIEGAND, T. et al. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(7):560–576, 2003.
- [91] YUAN, W.; HAO, P.; XU, C. Matrix factorization for fast DCT algorithms. In *IEEE International Conference on Acoustic, Speech, Signal Processing (ICASSP)*, volume 3, pages 948–951, 2006.

Compressão de Imagens tipo JPEG

O formato JPEG (Joint Photographic Expert Group) é um método de compressão muito usado em imagens. Este formato usa uma forma de compressão com perda de informação baseado na DCT, a qual é aplicada numa transformação 2-D (cf. Apêndice B). Esta operação matemática transforma cada imagem de entrada no domínio espacial ao domínio frequência. O modo de compressão tipo JPEG baseia-se no fato de que o sistema psicovisual humano descarta a informação de alta frequência, percebendo principalmente a informação de baixa frequência.

Na Figura A.1 são apresentadas as diferentes etapas da compressão tipo JPEG. Suas etapas podem ser dividida em duas partes: *codificação* e *decodificação*, as quais são resumidas a seguir [61, 81]:

- *Subdivisão da imagem original*: Nesta primeira etapa, a imagem original é dividida em sub-blocos de tamanho 8×8 ;
- *Transformada 2-D*: Cada sub-bloco de tamanho 8×8 é transformado do domínio espacial ao domínio espectral usando uma transformada 2-D, como discutido no Apêndice B. Isto proporciona 64 coeficientes no domínio da transformada;
- *Quantização*: Seguidamente, cada um dos 64 coeficientes é quantizado com uma matriz de quantização, a qual deve ser especificada. Nesta etapa ocorre a compressão, uma vez que uma quantidade significativa de coeficientes são anulados. Esta etapa é detalhada no Apêndice B;

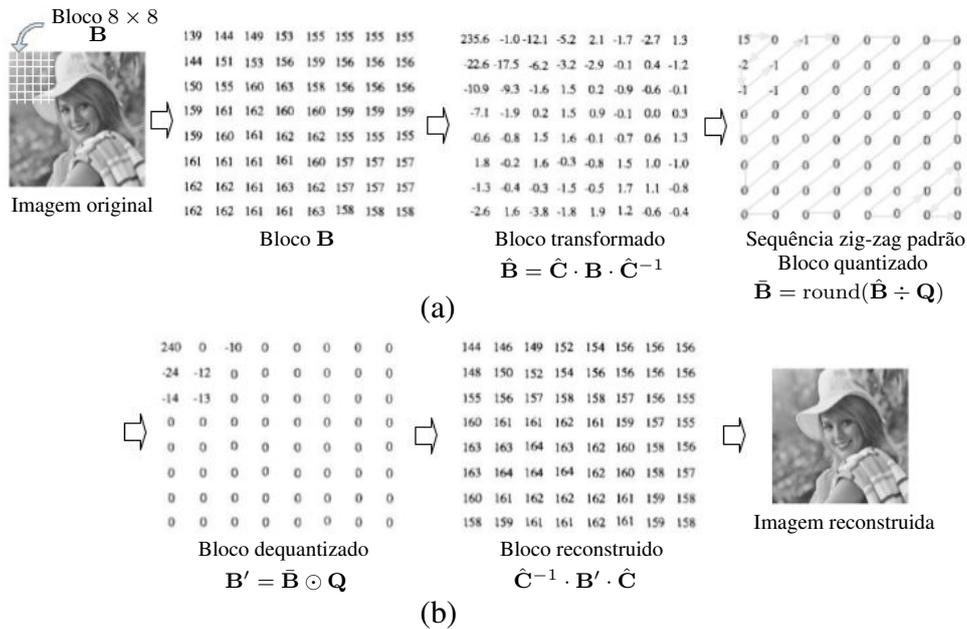


Figura A.1: Etapas da compressão tipo JPEG: (a) Etapa de *codificação*, (b) Etapa de *decodificação*.

- *Sequência zig-zag padrão*: Depois, os 64 coeficientes quantizados são ordenados na sequência zig-zag padrão. Assim, somente uma quantidade de r coeficientes significativos é retido no processo.

Na última etapa de codificação é feita uma compressão adicional ao codificar os coeficientes DCT quantizados mais compactamente com base nas suas características estatísticas. O padrão JPEG proporciona dois métodos para esta compressão adicional: o codificador Huffman [41] e o codificador aritmético.

A etapa de *decodificação* simplesmente inverte o processo de codificação, reconstruindo a imagem compactada.

Apêndice **B**

Transformação 2-D e Etapa de Quantização

Seja \mathbf{A} um bloco de imagem de tamanho 8×8 . Em geral, a transformação 2-D de \mathbf{A} respeito de uma aproximação $\hat{\mathbf{C}}$ é dada por:

$$\begin{aligned}
 \mathbf{B} &= \begin{cases} \hat{\mathbf{C}} \cdot \mathbf{A} \cdot \hat{\mathbf{C}}^T, & \text{se } \mathbf{T} \text{ é ortogonal,} \\ \hat{\mathbf{C}} \cdot \mathbf{A} \cdot \hat{\mathbf{C}}^{-1}, & \text{c.c.} \end{cases} \\
 &= \begin{cases} (\mathbf{v} \cdot \mathbf{v}^T) \odot (\mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^T), & \text{se } \mathbf{T} \text{ é ortogonal,} \\ (\mathbf{v} \cdot \mathbf{w}^T) \odot (\mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^{-1}), & \text{c.c.} \end{cases} \\
 &= \mathbf{R} \odot \hat{\mathbf{B}}, \tag{B.1}
 \end{aligned}$$

em que o operador \odot denota o produto matricial de Hadamard [69],

$$\mathbf{v} = \begin{cases} \text{diag}(\mathbf{S}), & \text{se } \mathbf{T} \text{ é ortogonal,} \\ \text{diag}(\hat{\mathbf{S}}), & \text{c.c. ,} \end{cases}$$

o vetor \mathbf{w} é dado pelos elementos inversos de \mathbf{v} ,

$$\mathbf{R} = \begin{cases} \mathbf{v} \cdot \mathbf{v}^T, & \text{se } \mathbf{T} \text{ é ortogonal,} \\ \mathbf{v} \cdot \mathbf{w}^T, & \text{c.c. ,} \end{cases}$$

e

$$\widehat{\mathbf{B}} = \begin{cases} \mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^\top, & \text{se } \mathbf{T} \text{ é ortogonal,} \\ \mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^{-1}, & \text{c.c.} \end{cases}$$

Observar que se a transformada \mathbf{T} é ortogonal, então a aproximação $\widehat{\mathbf{C}}$ resulta ortonormal. Portanto, nesse caso particular obtemos de (B.1) que

$$(\mathbf{v} \cdot \mathbf{v}^\top) \odot (\mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^\top) = (\mathbf{v} \cdot \mathbf{w}^\top) \odot (\mathbf{T} \cdot \mathbf{A} \cdot \mathbf{T}^{-1}).$$

A compressão de imagens segundo o esquema JPEG (cf. Apêndice A) fornece uma matriz de quantização \mathbf{Q} de tamanho 8×8 [81]. A matriz \mathbf{Q} depende da aproximação $\widehat{\mathbf{C}}$ considerada. No contexto de compressão tipo JPEG, uma matriz de coeficientes quantizados $\bar{\mathbf{B}}$ é calculada acorde a [81]:

$$\bar{\mathbf{B}} = \text{round}(\mathbf{B} \div \mathbf{Q}), \quad (\text{B.2})$$

em que o operador \div denota divisão matricial elemento a elemento.

Aplicando (B.1) e (B.2), obtemos:

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

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

Como conclusão, devido à absorção de \mathbf{R} na etapa de quantização, a complexidade da matriz \mathbf{S} ou $\hat{\mathbf{S}}$ pode ser desconsiderada nas aplicações de compressão de imagens.

Dado que, neste trabalho, os experimentos de compressão tipo JPEG foram aplicados a imagens em escala de cinzas, consideramos a matriz de quantização para luminância \mathbf{Q} dada no anexo da parte 1 do padrão JPEG [1]. Esta matriz é dada por:

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

A seguir, apresentam-se as matrizes $\tilde{\mathbf{Q}}$ envolvidas na etapa de quantização das

transformadas obtidas neste trabalho.

Transformadas do Capítulo 4:

$$\tilde{Q}_{T_1} = \begin{bmatrix} 128.0000 & 76.2102 & 63.2456 & 110.8513 & 192.0000 & 277.1281 & 322.5523 & 422.6204 \\ 83.1384 & 72.0000 & 76.6812 & 114.0000 & 180.1333 & 348.0000 & 328.6335 & 330.0000 \\ 88.5438 & 71.2039 & 80.0000 & 131.4534 & 252.9822 & 312.2019 & 345.0000 & 306.7246 \\ 96.9948 & 102.0000 & 120.4990 & 174.0000 & 353.3384 & 522.0000 & 438.1780 & 372.0000 \\ 144.0000 & 152.4205 & 234.0085 & 387.9794 & 544.0000 & 755.1742 & 651.4292 & 533.4716 \\ 166.2769 & 210.0000 & 301.2474 & 384.0000 & 561.1845 & 624.0000 & 618.9265 & 552.0000 \\ 309.9032 & 350.5424 & 390.0000 & 476.5186 & 651.4292 & 662.7443 & 600.0000 & 553.1998 \\ 498.8306 & 552.0000 & 520.3364 & 588.0000 & 775.9588 & 600.0000 & 564.1542 & 594.0000 \end{bmatrix},$$

$$\tilde{Q}_{T_2} = \begin{bmatrix} 128.0000 & 76.2102 & 56.5685 & 110.8513 & 192.0000 & 277.1281 & 288.4996 & 422.6204 \\ 83.1384 & 72.0000 & 68.5857 & 114.0000 & 180.1333 & 348.0000 & 293.9388 & 330.0000 \\ 79.1960 & 63.6867 & 64.0000 & 117.5755 & 226.2742 & 279.2418 & 276.0000 & 274.3429 \\ 96.9948 & 102.0000 & 107.7775 & 174.0000 & 353.3384 & 522.0000 & 391.9184 & 372.0000 \\ 144.0000 & 152.4205 & 209.3036 & 387.9794 & 544.0000 & 755.1742 & 582.6560 & 533.4716 \\ 166.2769 & 210.0000 & 269.4439 & 384.0000 & 561.1845 & 624.0000 & 553.5847 & 552.0000 \\ 277.1859 & 313.5347 & 312.0000 & 426.2112 & 582.6560 & 592.7765 & 480.0000 & 494.7969 \\ 498.8306 & 552.0000 & 465.4031 & 588.0000 & 775.9588 & 600.0000 & 504.5949 & 594.0000 \end{bmatrix},$$

$$\tilde{Q}_{T_3} = \begin{bmatrix} 128.0000 & 44.0000 & 56.5685 & 64.0000 & 192.0000 & 160.0000 & 288.4996 & 244.0000 \\ 48.0000 & 24.0000 & 39.5980 & 38.0000 & 104.0000 & 116.0000 & 169.7056 & 110.0000 \\ 79.1960 & 36.7696 & 64.0000 & 67.8823 & 226.2742 & 161.2203 & 276.0000 & 158.3919 \\ 56.0000 & 34.0000 & 62.2254 & 58.0000 & 204.0000 & 174.0000 & 226.2742 & 124.0000 \\ 144.0000 & 88.0000 & 209.3036 & 224.0000 & 544.0000 & 436.0000 & 582.6560 & 308.0000 \\ 96.0000 & 70.0000 & 155.5635 & 128.0000 & 324.0000 & 208.0000 & 319.6123 & 184.0000 \\ 277.1859 & 181.0193 & 312.0000 & 246.0732 & 582.6560 & 342.2397 & 480.0000 & 285.6711 \\ 288.0000 & 184.0000 & 268.7006 & 196.0000 & 448.0000 & 200.0000 & 291.3280 & 198.0000 \end{bmatrix},$$

$$\tilde{Q}_{T_4} = \begin{bmatrix} 128.0000 & 44.0000 & 126.4911 & 64.0000 & 192.0000 & 160.0000 & 645.1046 & 244.0000 \\ 48.0000 & 24.0000 & 88.5438 & 38.0000 & 104.0000 & 116.0000 & 379.4733 & 110.0000 \\ 177.0875 & 82.2192 & 320.0000 & 151.7893 & 505.9644 & 360.4997 & 1380.0000 & 354.1751 \\ 56.0000 & 34.0000 & 139.1402 & 58.0000 & 204.0000 & 174.0000 & 505.9644 & 124.0000 \\ 144.0000 & 88.0000 & 468.0171 & 224.0000 & 544.0000 & 436.0000 & 1302.8584 & 308.0000 \\ 96.0000 & 70.0000 & 347.8505 & 128.0000 & 324.0000 & 208.0000 & 714.6748 & 184.0000 \\ 619.8064 & 404.7715 & 1560.0000 & 550.2363 & 1302.8584 & 765.2712 & 2400.0000 & 638.7801 \\ 288.0000 & 184.0000 & 600.8328 & 196.0000 & 448.0000 & 200.0000 & 651.4292 & 198.0000 \end{bmatrix},$$

$$\tilde{Q}_{T_5} = \begin{bmatrix} 128.0000 & 62.2254 & 56.5685 & 90.5097 & 192.0000 & 226.2742 & 288.4996 & 345.0681 \\ 67.8823 & 48.0000 & 56.0000 & 76.0000 & 147.0782 & 232.0000 & 240.0000 & 220.0000 \\ 79.1960 & 52.0000 & 64.0000 & 96.0000 & 226.2742 & 228.0000 & 276.0000 & 224.0000 \\ 79.1960 & 68.0000 & 88.0000 & 116.0000 & 288.4996 & 348.0000 & 320.0000 & 248.0000 \\ 144.0000 & 124.4508 & 209.3036 & 316.7838 & 544.0000 & 616.5971 & 582.6560 & 435.5778 \\ 135.7645 & 140.0000 & 220.0000 & 256.0000 & 458.2052 & 416.0000 & 452.0000 & 368.0000 \\ 277.1859 & 256.0000 & 312.0000 & 348.0000 & 582.6560 & 484.0000 & 480.0000 & 404.0000 \\ 407.2935 & 368.0000 & 380.0000 & 392.0000 & 633.5677 & 400.0000 & 412.0000 & 396.0000 \end{bmatrix},$$

$$\tilde{Q}_{T_6} = \begin{bmatrix} 128.0000 & 62.2254 & 126.4911 & 90.5097 & 192.0000 & 226.2742 & 645.1046 & 345.0681 \\ 67.8823 & 48.0000 & 125.2198 & 76.0000 & 147.0782 & 232.0000 & 536.6563 & 220.0000 \\ 177.0875 & 116.2755 & 320.0000 & 214.6625 & 505.9644 & 509.8235 & 1380.0000 & 500.8792 \\ 79.1960 & 68.0000 & 196.7740 & 116.0000 & 288.4996 & 348.0000 & 715.5418 & 248.0000 \\ 144.0000 & 124.4508 & 468.0171 & 316.7838 & 544.0000 & 616.5971 & 1302.8584 & 435.5778 \\ 135.7645 & 140.0000 & 491.9350 & 256.0000 & 458.2052 & 416.0000 & 1010.7027 & 368.0000 \\ 619.8064 & 572.4334 & 1560.0000 & 778.1517 & 1302.8584 & 1082.2569 & 2400.0000 & 903.3715 \\ 407.2935 & 368.0000 & 849.7058 & 392.0000 & 633.5677 & 400.0000 & 921.2600 & 396.0000 \end{bmatrix},$$

$$\tilde{Q}_{T_7} = \begin{bmatrix} 128.0000 & 98.3870 & 126.4911 & 143.1084 & 192.0000 & 357.7709 & 645.1046 & 545.6006 \\ 107.3313 & 120.0000 & 197.9899 & 190.0000 & 232.5511 & 580.0000 & 848.5281 & 550.0000 \\ 177.0875 & 183.8478 & 320.0000 & 339.4113 & 505.9644 & 806.1017 & 1380.0000 & 791.9596 \\ 125.2198 & 170.0000 & 311.1270 & 290.0000 & 456.1579 & 870.0000 & 1131.3708 & 620.0000 \\ 144.0000 & 196.7740 & 468.0171 & 500.8792 & 544.0000 & 974.9256 & 1302.8584 & 688.7089 \\ 214.6625 & 350.0000 & 777.8175 & 640.0000 & 724.4860 & 1040.0000 & 1598.0613 & 920.0000 \\ 619.8064 & 905.0967 & 1560.0000 & 1230.3658 & 1302.8584 & 1711.1984 & 2400.0000 & 1428.3557 \\ 643.9876 & 920.0000 & 1343.5029 & 980.0000 & 1001.7585 & 1000.0000 & 1456.6400 & 990.0000 \end{bmatrix},$$

$$\tilde{Q}_{T_8} = \begin{bmatrix} 128.0000 & 90.7083 & 126.4911 & 131.9394 & 192.0000 & 329.8485 & 645.1046 & 503.0189 \\ 98.9545 & 102.0000 & 182.5377 & 161.5000 & 214.4015 & 493.0000 & 782.3043 & 467.5000 \\ 177.0875 & 169.4993 & 320.0000 & 312.9217 & 505.9644 & 743.1891 & 1380.0000 & 730.1507 \\ 115.4470 & 144.5000 & 286.8449 & 246.5000 & 420.5568 & 739.5000 & 1043.0724 & 527.0000 \\ 144.0000 & 181.4166 & 468.0171 & 461.7878 & 544.0000 & 898.8370 & 1302.8584 & 634.958 \\ 197.9091 & 297.5000 & 717.1123 & 544.0000 & 667.9431 & 884.0000 & 1473.3397 & 782.0000 \\ 619.8064 & 834.4579 & 1560.0000 & 1134.3412 & 1302.8584 & 1577.6470 & 2400.0000 & 1316.8789 \\ 593.7272 & 782.0000 & 1238.6485 & 833.0000 & 923.5757 & 850.0000 & 1342.9557 & 841.5000 \end{bmatrix},$$

$$\tilde{Q}_{T_{16}} = \begin{bmatrix} 16.0000 & 15.5563 & 14.1421 & 22.6274 & 24.0000 & 56.5685 & 72.1249 & 86.2670 \\ 8.4853 & 12.0000 & 14.0000 & 19.0000 & 18.3848 & 58.0000 & 60.0000 & 55.0000 \\ 9.8995 & 13.0000 & 16.0000 & 24.0000 & 28.2843 & 57.0000 & 69.0000 & 56.0000 \\ 9.8995 & 17.0000 & 22.0000 & 29.0000 & 36.0624 & 87.0000 & 80.0000 & 62.0000 \\ 18.0000 & 31.1127 & 52.3259 & 79.1960 & 68.0000 & 154.1493 & 145.6640 & 108.8944 \\ 16.9706 & 35.0000 & 55.0000 & 64.0000 & 57.2756 & 104.0000 & 113.0000 & 92.0000 \\ 34.6482 & 64.0000 & 78.0000 & 87.0000 & 72.8320 & 121.0000 & 120.0000 & 101.0000 \\ 50.9117 & 92.0000 & 95.0000 & 98.0000 & 79.1960 & 100.0000 & 103.0000 & 99.0000 \end{bmatrix}.$$

Transformadas do Capítulo 5:

$$\tilde{Q}_{\tilde{T}_8} = \begin{bmatrix} 16.0000 & 8.9815 & 10.0000 & 13.0639 & 24.0000 & 32.6599 & 51.0000 & 49.8063 \\ 14.6969 & 12.0000 & 17.1464 & 19.0000 & 31.8434 & 58.0000 & 73.4847 & 55.0000 \\ 14.0000 & 10.6145 & 16.0000 & 19.5959 & 40.0000 & 46.5403 & 69.0000 & 45.7238 \\ 17.1464 & 17.0000 & 26.9444 & 29.0000 & 62.4620 & 87.0000 & 97.9796 & 62.0000 \\ 18.0000 & 17.9629 & 37.0000 & 45.7238 & 68.0000 & 88.9981 & 103.0000 & 62.8702 \\ 29.3939 & 35.0000 & 67.3610 & 64.0000 & 99.2043 & 104.0000 & 138.3962 & 92.0000 \\ 49.0000 & 52.2558 & 78.0000 & 71.0352 & 103.0000 & 98.7961 & 120.0000 & 82.4662 \\ 88.1816 & 92.0000 & 116.3508 & 98.0000 & 137.1714 & 100.0000 & 126.1487 & 99.0000 \end{bmatrix},$$

$$\tilde{Q}_{\hat{T}_8} = \begin{bmatrix} 16.0000 & 12.7017 & 14.1421 & 13.0639 & 24.0000 & 32.6599 & 72.1249 & 70.4367 \\ 10.3923 & 12.0000 & 17.1464 & 13.4350 & 22.5167 & 41.0122 & 73.4847 & 55.0000 \\ 9.8995 & 10.6145 & 16.0000 & 13.8564 & 28.2843 & 32.9090 & 69.0000 & 45.7238 \\ 17.1464 & 24.0416 & 38.1051 & 29.0000 & 62.4620 & 87.0000 & 138.5641 & 87.6812 \\ 18.0000 & 25.4034 & 52.3259 & 45.7238 & 68.0000 & 88.9981 & 145.6640 & 88.9119 \\ 29.3939 & 49.4975 & 95.2628 & 64.0000 & 99.2043 & 104.0000 & 195.7217 & 130.1076 \\ 34.6482 & 52.2558 & 78.0000 & 50.2295 & 72.8320 & 69.8594 & 120.0000 & 82.4662 \\ 62.3538 & 92.0000 & 116.3508 & 69.2965 & 96.9948 & 70.7107 & 126.1487 & 99.0000 \end{bmatrix}.$$

Acrônimos usados

APE	Erro absoluto percentual
APE (PSNR)	Erro absoluto percentual relativo ao PSNR
APE (SSIM)	Erro absoluto percentual relativo ao SSIM
BAS	Transformada de Bouguezel-Ahmad-Swamy
CSD	<i>Canonical signed digital</i>
DCT	Transformada discreta do cosseno
DCT-I	Transformada discreta do cosseno tipo I
DCT-II	Transformada discreta do cosseno tipo II
DCT-III	Transformada discreta do cosseno tipo III
DCT-IV	Transformada discreta do cosseno tipo IV
FIR	Resposta ao impulso finito
HEVC	<i>High efficiency video coding</i>
HT	Transformada de Hadamard
JPEG	<i>Join photographic experts group</i>
KLT	Transformada de Karhunen-Loève
MPEG	<i>Moving picture experts group</i>
MRDCT	DCT arredondada modificada
MSE	Erro quadrático médio

PSNR	Relação sinal-ruído de pico
RDCT	DCT arredondada
RF	Rádio-frequência
SDCT	<i>Signed</i> DCT
SSIM	Índice de similaridade estrutural
WHT	Transformada de Walsh-Hadamard