



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CAMPUS AGRESTE
NÚCLEO DE TECNOLOGIA
CURSO DE ENGENHARIA DE PRODUÇÃO

LANIELSON ANDRADE SILVA

**APROXIMAÇÕES DE BAIXA COMPLEXIDADE COMPUTACIONAL PARA
PROCESSAMENTO DE IMAGENS EM SISTEMAS DE RASTREAMENTO VISUAL**

Caruaru
2026

LANIELSON ANDRADE SILVA

**APROXIMAÇÕES DE BAIXA COMPLEXIDADE COMPUTACIONAL PARA
PROCESSAMENTO DE IMAGENS EM SISTEMAS DE RASTREAMENTO VISUAL**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Engenharia de Produção do Centro Acadêmico do Agreste do Campus Agreste da Universidade Federal de Pernambuco – UFPE, na modalidade de monografia, como requisito parcial para a obtenção do grau de bacharel em Engenharia de Produção.

Área de concentração: Pesquisa Operacional.

Orientador: Prof. Dr. R. J. Cintra

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Andrade Silva, Lanielson.

Aproximações de Baixa Complexidade Computacional para Processamento de Imagens em Sistemas de Rastreamento Visual / Lanielson Andrade Silva. - Caruaru, 2026.

100 p. : il., tab.

Orientador(a): R. J. Cintra

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro Acadêmico do Agreste, Engenharia de Produção, 2026.

Inclui referências, apêndices.

1. aproximações matriciais. 2. transformada discreta do cosseno. 3. transformada discreta de Fourier. 4. baixa complexidade computacional. 5. rastreamento visual. 6. indústria 4.0. I. Cintra, R. J.. (Orientação). II. Título.

620 CDD (22.ed.)

LANIELSON ANDRADE SILVA

**APROXIMAÇÕES DE BAIXA COMPLEXIDADE COMPUTACIONAL PARA
PROCESSAMENTO DE IMAGENS EM SISTEMAS DE RASTREAMENTO VISUAL**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Engenharia de Produção do Centro Acadêmico do Agreste do Campus Agreste da Universidade Federal de Pernambuco – UFPE, na modalidade de monografia, como requisito parcial para a obtenção do grau de bacharel em Engenharia de Produção.

Aprovado em: 06/05/2026

BANCA EXAMINADORA

Prof. Dr. R. J. Cintra (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Augusto José da Silva Rodrigues (Examinador Interno)
Universidade Federal de Pernambuco

Prof. Dr. Lucimário Gois de Oliveira Silva (Examinador Interno)
Universidade Federal de Pernambuco

AGRADECIMENTOS

A Deus, princípio e fim de todas as coisas, expresso a minha mais profunda gratidão. Reconheço que a capacidade humana de compreender relações abstratas e fazer descobertas não surge de si mesma, mas tem origem em uma Inteligência maior. A busca incessante pela verdade reflete esse dom que nos foi concedido. Como escreveu Santo Agostinho, “*fecisti nos ad te, Domine, et inquietum est cor nostrum, donec requiescat in te*”¹: é desta inquietude que nos impele à busca da verdade em todas as suas formas — incluindo aquelas que se manifestam na linguagem matemática —, que nasce também o ato de pesquisar.

Aos meus pais, Sr. L. João da Silva e Sra. G. Andrade Torres Silva, expresso gratidão inexprimível pelo amor incondicional, pelo apoio constante, pela confiança em cada etapa da minha formação e pelos esforços que fizeram para que eu pudesse chegar até aqui. Nada do que aqui se apresenta teria sido possível sem o sacrifício silencioso e o exemplo de honestidade e integridade que me acompanham desde os primeiros anos. Se esta pesquisa tem algum mérito, ele lhes pertence tanto quanto a mim; se nela há virtudes é porque foram eles os primeiros a ensiná-las.

Ao meu orientador, Prof. Dr. R. J. Cintra, manifesto minha gratidão pelo acompanhamento, pelas orientações criteriosas e pelas discussões técnicas que, ao longo destes anos de Iniciação Científica, aprimoraram o meu modo de pensar e me transmitiram máximas da engenharia que carregarei por toda a trajetória profissional. Os conselhos recebidos ultrapassam em muito o escopo deste trabalho e permanecerão como bússola nas etapas futuras de minha trajetória.

À Gideão Milanez, agradeço pelo apoio constante durante o período de estágio. Sua disposição em compartilhar conhecimento foi fundamental nessa etapa da minha formação profissional.

Aos colegas do *Industrial Signal Processing Lab* (ISPL), agradeço pelo ambiente colaborativo. De forma particular, registro minha gratidão aos Srs. Luis Miguel e Wendy Aleixo, cujo apoio foi fundamental em diversos momentos — seja nas discussões técnicas, seja no auxílio desde a formatação deste trabalho até sua defesa.

A infraestrutura e os recursos técnicos necessários ao desenvolvimento deste trabalho foram disponibilizados pelo ISPL, do Centro Acadêmico do Agreste.

A todos que, nomeados ou não, contribuíram para que este trabalho se tornasse realidade, minha sincera gratidão.

¹ “Fizeste-nos para Ti, Senhor, e inquieto está o nosso coração enquanto não repousa em Ti.” AGOSTINHO, *Confissões*, Livro I, I.1.

“LABOR OMNIA VINCIT IMPROBUS ET DURIS
URGENS IN REBUS EGESTAS.”

— *Virgílio (Geórgicas, I, 145–146)*

RESUMO

O avanço dos sistemas digitais e o aumento do volume de dados multimídia impulsionaram o desenvolvimento de técnicas eficientes de processamento de sinais. Nesse contexto, as transformadas discretas têm papel fundamental em aplicações como compressão de imagens, análise espectral e sistemas de visão computacional. Contudo, a implementação direta dessas transformadas frequentemente envolve custo computacional elevado, devido à presença de operações de multiplicação, o que limita sua aplicação em ambientes com restrições de hardware. Como alternativa, as transformadas aproximadas buscam reduzir a complexidade aritmética, preservando, com perda tolerável de desempenho, as propriedades da transformada exata. Este trabalho apresenta um método para a obtenção de aproximações de baixa complexidade computacional da transformada discreta do cosseno (DCT), com aplicações em rastreamento visual para monitoramento contínuo da qualidade em linhas de produção industrial operadas por dispositivos embarcados. A proposta advém das demandas da Indústria 4.0, nas quais sistemas de monitoramento devem operar em tempo real, sob restrições de latência, consumo energético e capacidade de processamento. A metodologia adotada é formulada como um problema de otimização não linear, no qual se busca minimizar a diferença angular entre os elementos exatos da transformada discreta de Fourier (DFT) e candidatos pertencentes a conjuntos de inteiros gaussianos, i.e., multiplicandos triviais. A partir das aproximações da DFT, são obtidas aproximações da DCT com base em teorema que estabelece uma relação algébrica entre as transformadas de diferentes comprimentos. As aproximações da DCT resultantes são avaliadas por meio de figuras de mérito clássicas da literatura, como (i) erro total de energia, (ii) erro médio quadrático, (iii) ganho de codificação unificado e (iv) eficiência de transformação, permitindo comparação com o estado da arte. Adicionalmente, um experimento de compressão no padrão JPEG é conduzido sobre um conjunto de imagens de teste. O desempenho é avaliado por métricas amplamente utilizadas, como Relação Sinal-Ruído de Pico (PSNR), Índice de Similaridade Estrutural (SSIM) e Índice de Qualidade Universal (UQI). Os resultados evidenciam desempenho competitivo das aproximações propostas, com adequada preservação da qualidade perceptual das imagens reconstruídas. Destaca-se que todas as matrizes obtidas são livres de multiplicações, sendo implementáveis por operações de adição e deslocamento de bits, favorecendo sua aplicação em sistemas embarcados de baixo custo. Dessa forma, o trabalho contribui com um método para geração de aproximações eficientes da DCT, aplicável a sistemas de inspeção visual automatizada e adequado a soluções de monitoramento contínuo no contexto da Indústria 4.0.

Palavras-chave: aproximações matriciais; transformada discreta do cosseno; transformada discreta de Fourier; baixa complexidade computacional; rastreamento visual; indústria 4.0.

ABSTRACT

The advance of digital systems and the increasing volume of multimedia data have driven the development of efficient signal processing techniques. In this context, discrete transforms play a fundamental role in applications such as image compression, spectral analysis, and computer vision systems. However, the direct implementation of these transforms often entails high computational cost due to the presence of multiplication operations, which limits their application in hardware-constrained environments. As an alternative, approximate transforms aim to reduce the arithmetic complexity while preserving, with tolerable performance loss, the properties of the exact transform. This work presents a method for obtaining low-complexity approximations of the discrete cosine transform (DCT), with applications in visual tracking for continuous quality monitoring in industrial production lines operated by embedded devices. The proposal arises from the demands of Industry 4.0, in which monitoring systems must operate in real time, under constraints of latency, energy consumption, and processing capacity. The adopted methodology is formulated as a non-linear optimization problem, in which one seeks to minimize the angular difference between the exact elements of the discrete Fourier transform (DFT) and candidates belonging to sets of Gaussian integers, i.e., trivial multiplicands. From the DFT approximations, DCT approximations are obtained based on a theorem that establishes an algebraic relation between the transforms of different lengths. The resulting DCT approximations are evaluated by means of classical figures of merit from the literature, such as (i) total energy error, (ii) mean squared error, (iii) unified coding gain, and (iv) transform efficiency, enabling comparison with the state of the art. Furthermore, a compression experiment in the JPEG standard is carried out over a set of test images. Performance is assessed by widely used metrics, such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), and Universal Quality Index (UQI). The results indicate competitive performance of the proposed approximations, with adequate preservation of the perceptual quality of the reconstructed images. It is noteworthy that all obtained matrices are multiplication-free and can be implemented through addition and bit-shift operations, favoring their application in low-cost embedded systems. Thus, this work contributes with a method for generating efficient DCT approximations, applicable to automated visual inspection systems and suitable for continuous monitoring solutions in the context of Industry 4.0.

Keywords: matrix approximations; discrete cosine transform; discrete Fourier transform; low computational complexity; visual tracking; industry 4.0.

LISTA DE FIGURAS

Figura 1 – Arquitetura de um sistema de rastreamento.	24
Figura 2 – Representação de uma <i>Butterfly</i>	43
Figura 3 – Algoritmo rápido para a transformada de Walsh-Hadamard de ordem 8. Setas sólidas indicam soma; setas tracejadas, subtração.	45
Figura 4 – Fluxo metodológico adotado no trabalho	48
Figura 5 – Diagrama de fluxo da matriz $\mathbf{T}_4^{(1)}$	56
Figura 6 – Diagrama de fluxo da matriz $\mathbf{T}_4^{(2)}$	56
Figura 7 – Diagrama de fluxo da matriz $\mathbf{T}_8^{(1)}$	59
Figura 8 – Diagrama de fluxo das matrizes $\mathbf{T}_8^{(2)}$ e $\mathbf{T}_8^{(4)}$	59
Figura 9 – Diagrama do processo de compressão tipo-JPEG.	62
Figura 10 – Padrão de varredura zig-zag para bloco 8×8 com $r = 16$ coeficientes retidos.	63
Figura 11 – Comparação da compressão de imagens ($N = 4$) para a imagem 1.2.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	67
Figura 12 – Comparação da compressão de imagens ($N = 4$) para a imagem 1.3.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	68
Figura 13 – Comparação da compressão de imagens ($N = 4$) para a imagem 1.4.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	69
Figura 14 – Comparação da compressão de imagens ($N = 4$) para a imagem 3.2.25: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	70
Figura 15 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.1.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	71
Figura 16 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.1.11: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	72
Figura 17 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.1.12: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	73
Figura 18 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.3.02: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$	74

Figura 19 – Comparação da compressão de imagens ($N = 4$) para a imagem 7.1.01: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_4^{(1)}$ e (d) aproximação $\hat{C}_4^{(2)}$	75
Figura 20 – Comparação da compressão de imagens ($N = 8$) para a imagem 1.2.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	77
Figura 21 – Comparação da compressão de imagens ($N = 8$) para a imagem 1.3.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	78
Figura 22 – Comparação da compressão de imagens ($N = 8$) para a imagem 1.4.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	79
Figura 23 – Comparação da compressão de imagens ($N = 8$) para a imagem 3.2.25: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	80
Figura 24 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.1.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	81
Figura 25 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.1.11: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	82
Figura 26 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.1.12: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	83
Figura 27 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.3.02: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	84
Figura 28 – Comparação da compressão de imagens ($N = 8$) para a imagem 7.1.01: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$	85

LISTA DE TABELAS

Tabela 1	–	Comparação de desempenho das aproximações da DCT para $N = 4$.	57
Tabela 2	–	Complexidade aritmética das aproximações para $N = 4$.	57
Tabela 3	–	Comparação de desempenho de aproximações da DCT para $N = 8$.	60
Tabela 4	–	Comparação de complexidade computacional para $N = 8$.	60
Tabela 5	–	Resumo das métricas de qualidade de imagem.	65
Tabela 6	–	Imagens de teste selecionadas do banco USC-SIPI.	66
Tabela 7	–	Métricas de qualidade da compressão de imagens para $(N = 4, r = 4)$.	76
Tabela 8	–	Métricas de qualidade da compressão de imagens para $(N = 8, r = 16)$.	86

LISTA DE ABREVIATURAS E SIGLAS

BAS	Aproximação de Bouguezel, Ahmad e Swamy (2008)
CEP	Controle Estatístico de Processo
CPS	<i>Cyber-Physical Systems</i> – Sistemas Ciber-Físicos
DCT	<i>Discrete Cosine Transform</i> – Transformada Discreta do Cosseno
DCT-II	<i>Discrete Cosine Transform Type II</i> – DCT do Tipo II
DFT	<i>Discrete Fourier Transform</i> – Transformada Discreta de Fourier
HEVC	<i>High Efficiency Video Coding</i> – Padrão de codificação de vídeo de alta eficiência
IDCT	<i>Inverse Discrete Cosine Transform</i> – Transformada Discreta do Cosseno Inversa
IDFT	<i>Inverse Discrete Fourier Transform</i> – Transformada Discreta de Fourier Inversa
JPEG	<i>Joint Photographic Experts Group</i> – Padrão de compressão de imagens JPEG
KLT	<i>Karhunen–Loève Transform</i> – Transformada de Karhunen–Loève
LO	Aproximação de Lengwehasatit e Ortega
MPEG	<i>Moving Picture Experts Group</i> – Padrão de compressão de vídeo MPEG
MRDCT	<i>Modified Rounded DCT</i> – DCT arredondada modificada
MSE	<i>Mean Squared Error</i> – Erro Quadrático Médio
OCBSML	Aproximação de Oliveira, Cintra, Bayer, da Silveira, Madanayake e Leite
PCA	<i>Principal Component Analysis</i> – Análise de Componentes Principais
PSNR	<i>Peak Signal-to-Noise Ratio</i> – Razão Sinal-Ruído de Pico
RDCT	<i>Rounded DCT</i> – DCT arredondada
SDCT	<i>Signed DCT</i> – DCT sinalizada
SSIM	<i>Structural Similarity Index Measure</i> – Índice de Similaridade Estrutural
TCB	Aproximação de Tablada, Cintra e Bayer
UQI	<i>Universal Quality Index</i> – Índice Universal de Qualidade
USC-SIPI	<i>University of Southern California – Signal and Image Processing Institute</i>
WHT	<i>Walsh–Hadamard Transform</i> – Transformada de Walsh–Hadamard

LISTA DE SÍMBOLOS

$\mathbf{C}_N$	Matriz DCT exata de ordem N
$\widehat{\mathbf{C}}_N$	Aproximação da DCT de ordem N
$\widehat{\mathbf{C}}_N^{(p)}$	Aproximação da DCT obtida do espaço de busca $\mathcal{P}'_p$
$\mathbf{F}_N$	Matriz DFT exata de ordem N
$\widehat{\mathbf{F}}_N$	Aproximação da DFT de ordem N
$\widehat{\mathbf{F}}_N^{(p)}$	Aproximação da DFT obtida do espaço de busca $\mathcal{P}'_p$
$\mathbf{T}_N$	Matriz de baixa complexidade de ordem N
$\mathbf{T}_N^{(p)}$	Matriz de baixa complexidade obtida do espaço $\mathcal{P}'_p$
$\mathbf{S}$	Matriz diagonal de normalização (caso ortogonal)
$\widetilde{\mathbf{S}}$	Matriz diagonal de normalização (caso não-ortogonal)
$\mathbf{I}_N$	Matriz identidade de ordem N
$\mathbf{D}$	Matriz diagonal
$[\mathbf{F}_N]_{n,k}$	Elemento da n -ésima linha e k -ésima coluna de $\mathbf{F}_N$
$[\widehat{\mathbf{F}}_N]_{n,k}$	Elemento da n -ésima linha e k -ésima coluna de $\widehat{\mathbf{F}}_N$
$\Re\{\cdot\}$	Parte real
$\text{Im}\{\cdot\}$	Parte imaginária
$\text{Arg}(\cdot)$	Argumento de um número complexo
$\odot$	Produto de Hadamard (elemento a elemento)
$\delta[\cdot]$	Delta de Kronecker
ρ	Coefficiente de correlação de primeira ordem
$\mathbb{C}$	Conjunto dos números complexos
$\mathbb{Z}$	Conjunto dos números inteiros
$\mathbb{Z}[j]$	Conjunto dos inteiros gaussianos
S_p	Conjunto de multiplicandos triviais
$\mathcal{P}_p$	Espaço de busca de inteiros gaussianos derivado de S_p
$\mathcal{P}'_p$	Espaço de busca reduzido
$\mathcal{M}_N(\mathcal{P}'_p)$	Espaço das matrizes $N \times N$ com entradas em $\mathcal{P}'_p$
N	Ordem ou comprimento da transformada
j	Unidade imaginária, $j = \sqrt{-1}$
$\ \cdot\ _F$	Norma de Frobenius
ε	Erro total de energia
C_g^*	Ganho de codificação unificado
η	Eficiência de transformação
ϕ	Desvio de ortogonalidade
r	Número de coeficientes retidos na varredura zig-zag
$\mathbf{M}_r$	Máscara binária de retenção de r coeficientes

SUMÁRIO

	LISTA DE FIGURAS	8
	LISTA DE TABELAS	9
	LISTA DE ABREVIATURAS E SIGLAS	10
	LISTA DE SÍMBOLOS	11
1	INTRODUÇÃO	15
1.1	CONTEXTUALIZAÇÃO E MOTIVAÇÃO	15
1.2	TÓPICO DE INVESTIGAÇÃO	18
1.3	ESTADO DA ARTE	18
1.4	OBJETIVOS	19
1.4.1	Objetivo Geral	19
1.4.2	Objetivos Específicos	19
2	SISTEMA DE RASTREAMENTO VISUAL	21
2.1	INTRODUÇÃO	21
2.2	MODELO DE RASTREAMENTO VISUAL	21
2.2.1	Formulação por filtro de partículas	21
2.2.2	Tensor de aparência e transformada 3D-DCT	22
2.2.3	Filtragem espectral	22
2.2.4	Decisão e atualização incremental	23
2.2.5	Representação simplificada	23
2.3	COMPRESSÃO DE IMAGENS EM RASTREAMENTO VISUAL	23
2.4	RASTREAMENTO E INDÚSTRIA 4.0	25
3	TRANSFORMADA DISCRETA DE FOURIER	26
3.1	FUNDAMENTAÇÃO MATEMÁTICA	26
3.1.1	Inteiros Gaussianos	26
3.1.2	Módulo e argumento	26
3.1.3	Identidade de Euler	27
3.1.4	Raízes da Unidade	27
3.2	DEFINIÇÃO	28
3.2.1	Representação Matricial	29
3.2.2	Ortogonalidade	29
3.3	PROPRIEDADES	30
3.3.1	Linearidade	30
3.3.2	Deslocamento Circular no Tempo	31
3.3.3	Deslocamento Circular na Frequência	31
3.3.4	Dualidade	31
3.3.5	Convolução Circular	31

3.3.6	Teorema de Parseval	31
3.3.7	Propriedades de Simetria	32
3.3.8	Complexidade Computacional	32
3.3.9	Interpretação Geométrica	32
4	TRANSFORMADA DISCRETA DO COSSENO	34
4.1	INTRODUÇÃO	34
4.2	A TRANSFORMADA DE KARHUNEN-LOÈVE	34
4.2.1	Correlação	34
4.2.2	Formulação do Problema de Descorrelação	35
4.2.3	Propriedades Fundamentais	36
4.2.4	KLT para Sinais Markovianos de Primeira Ordem	36
4.3	DCT-II	37
4.3.1	Definição	37
4.3.2	DCT-II como Aproximação da KLT	38
4.3.3	Representação Matricial	39
4.4	DCT-II BIDIMENSIONAL	39
4.4.1	Caso Unidimensional	39
4.4.2	Caso Bidimensional	40
4.4.3	Aplicação em Compressão de Imagens	40
5	APROXIMAÇÕES DE BAIXA COMPLEXIDADE	41
5.1	INTRODUÇÃO	41
5.2	COMPLEXIDADE COMPUTACIONAL	41
5.2.1	Complexidade Aritmética	41
5.3	APROXIMAÇÃO DE BAIXA COMPLEXIDADE	42
5.4	ALGORITMOS RÁPIDOS	43
5.4.1	Fatoração de Matrizes	44
5.5	FIGURAS DE MÉRITO	45
5.5.1	Modelagem do Sinal de Entrada	45
5.5.2	Medidas de Similaridade	46
5.5.2.1	Erro de Energia Total	46
5.5.2.2	Erro Médio Quadrático	46
5.5.3	Medidas de Codificação	46
5.5.3.1	Ganho de Codificação Unificado	46
5.5.3.2	Eficiência da Transformada	47
6	METODOLOGIA	48
6.1	FLUXO METODOLÓGICO	48
6.2	PROBLEMA DE OTIMIZAÇÃO	48
6.2.1	Espaços de Busca	48

6.2.1.1	Conjuntos de multiplicandos triviais	49
6.2.1.2	Conjuntos de inteiros gaussianos	49
6.2.1.3	Espaço das matrizes aproximadas	50
6.2.2	Função Objetivo	50
6.3	CONVERSÃO DFT-DCT	51
6.3.1	Fundamentação Teórica	51
6.3.2	Demonstração do Teorema	52
6.3.3	Adaptação Matricial	53
6.3.4	Algoritmo	53
7	RESULTADOS	55
7.1	APROXIMAÇÕES PROPOSTAS	55
7.1.1	Comparação para $N = 4$	55
7.1.2	Comparação para $N = 8$	57
8	ENSAIO COMPUTACIONAL	62
8.1	COMPRESSÃO DE IMAGENS TIPO-JPEG	62
8.1.1	Etapas do Processo de Compressão	62
8.2	MÉTRICAS DE QUALIDADE DA IMAGEM	64
8.2.1	Métricas de Erro	64
8.2.1.1	MSE	64
8.2.1.2	PSNR	64
8.2.2	Métricas Perceptuais	64
8.2.2.1	UQI	64
8.2.2.2	SSIM	65
8.3	RESULTADOS NUMÉRICOS	65
8.3.1	Imagens de teste	65
8.3.2	Procedimento experimental	66
8.3.3	Análise dos resultados	87
9	CONCLUSÃO	88
9.1	PERSPECTIVAS DE TRABALHOS FUTUROS	88
	REFERÊNCIAS	90
	APÊNDICE A – APROXIMAÇÕES DA DFT	95

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E MOTIVAÇÃO

A manufatura atual passa por um processo de reconfiguração, impulsionado pela integração entre automação avançada, sistemas ciber-físicos (*Cyber-Physical Systems — CPS*), conectividade em larga escala e análise de dados, caracterizando o paradigma conhecido como Quarta Revolução Industrial ou Indústria 4.0 (Kagermann, Wahlster e Helbig, 2013).

Nesse paradigma, processos produtivos integram máquinas, sensores e sistemas computacionais em redes interconectadas capazes de coletar, processar e compartilhar dados em tempo real. A capacidade de supervisão e de resposta do sistema produtivo alinha-se aos cinco objetivos clássicos de desempenho, propostos por (Slack, Chambers e Johnston, 2010): qualidade, velocidade, credibilidade, flexibilidade e custo. Em particular, o monitoramento da produção fortalece a capacidade de detectar desvios em menor tempo, aumenta a previsibilidade do processo e reduz perdas associadas a falhas e retrabalho (Zhong *et al.*, 2017). Desse modo, a transformação digital dos processos proporciona aumento da eficiência operacional ao longo de toda a cadeia de valor, representando uma mudança fundamental na forma como produtos são concebidos, fabricados e entregues ao mercado (Thoben, Wiesner e Wuest, 2017).

Nesse contexto, a qualidade torna-se um elemento fundamental da competitividade industrial e um dos principais objetivos de desempenho da produção. A qualidade não se limita à conformidade do produto final, mas envolve a capacidade do sistema produtivo de manter consistência, reduzir variabilidade e evitar a ocorrência de não conformidades ao longo de toda a cadeia produtiva, desde o recebimento de matérias-primas até a expedição do produto final (Oakland, 2014). Em outras palavras, quanto maior a capacidade do processo de produzir corretamente desde a primeira tentativa, maior tende a ser seu desempenho em qualidade, credibilidade e custo.

Os métodos tradicionais de controle de qualidade, fundamentados no Controle Estatístico de Processo (CEP), baseiam-se na coleta periódica de amostras e na análise de parâmetros críticos (Montgomery, 2012). Embora eficazes para detectar tendências e desvios sistemáticos ao longo do tempo, tais métodos dependem de amostragem e de medições pontuais (Woodall, 2000). Em processos de alta velocidade ou nos quais defeitos surgem de forma esporádica — como, por exemplo, linhas de envase e rotulagem em indústrias de bebidas, processos de inspeção de placas em manufatura eletrônica ou linhas de estampagem e usinagem em larga escala — essa limitação pode comprometer simultaneamente a qualidade e a credibilidade do processo, além de aumentar custos associados a retrabalho e interrupções operacionais.

A superação de tais restrições requer a adoção de estratégias capazes de avaliar a totalidade dos itens fabricados, substituindo a lógica amostral pelo monitoramento contínuo da produção (Weese *et al.*, 2016). Tal abordagem, viabilizada por tecnologias de automação, possibilita a detecção imediata de não conformidades, reduzindo o intervalo entre a manifestação do pro-

blema e a intervenção corretiva (Kusiak, 2018). Esse tipo de monitoramento oferece benefícios como: (i) detecção imediata de desvios no processo, possibilitando intervenções corretivas antes que lotes sejam comprometidos; (ii) rastreabilidade completa de cada item produzido; (iii) redução de custos com retrabalho; e (iv) geração de dados para análise de tendências e melhoria contínua (He *et al.*, 2017).

Do ponto de vista econômico, embora a implantação de sistemas de rastreamento visual e de inspeção contínua demande investimento inicial em hardware, software e integração, trata-se de uma solução com potencial para reduzir custos operacionais ao longo do tempo. Isso ocorre porque a identificação precoce de defeitos diminui perdas por refugos, retrabalho, devoluções e paradas não planejadas, além de reduzir o custo da não qualidade. A longo prazo, a melhoria da estabilidade do processo e a diminuição de desperdícios tendem a elevar a produtividade e a margem operacional, favorecendo a amortização do investimento inicial.

O controle de qualidade contínuo é operacionalizado por sistemas ciber-físicos (Monostori *et al.*, 2016), que integram dispositivos computacionais embarcados — tais como sensores e atuadores — a redes de comunicação e *software*, a fim de monitorar e controlar operações do mundo real, como temperatura de fornos industriais, movimento de peças em uma esteira rolante, entre outros. Nesse contexto, os CPS são a infraestrutura subjacente em aplicações como monitoramento industrial, Internet das Coisas e redes de sensores sem fio.

Microcontroladores, sistemas embarcados e plataformas de hardware dedicado têm viabilizado a execução de algoritmos de processamento de imagens para monitorar a produção. Entretanto, tais dispositivos apresentam limitações de capacidade de processamento, memória e consumo energético, o que impõe restrições à complexidade dos algoritmos que podem ser executados localmente. Nesse cenário, há um *trade-off* entre desempenho computacional e precisão. Desse modo, a adoção de técnicas de processamento de imagens com baixa complexidade é requerida, especialmente em aplicações de rastreamento visual em tempo real (Madanayake *et al.*, 2015).

A inspeção visual automatizada ou rastreamento visual é uma das aplicações mais disseminadas da visão computacional no contexto industrial, substituindo progressivamente os métodos tradicionais baseados em amostragem (Malamas *et al.*, 2003). O rastreamento visual consiste na predição contínua da localização de alvos móveis em sequências de imagens capturadas por câmeras posicionadas ao longo do sistema produtivo (Yilmaz, Javed e Shah, 2006). A capacidade de monitorar o deslocamento de peças e produtos viabiliza a análise de indicadores para a gestão da produção. Essas informações são fundamentais para atividades de crono-análise, balanceamento de linhas, identificação de gargalos e melhoria contínua (Barnes, 1980), favorecendo, em algum grau, os cinco objetivos clássicos de desempenho.

Nesse contexto, a adoção de sistemas de rastreamento visual impõe desafios relacionados ao custo computacional das operações algorítmicas (Madanayake *et al.*, 2015). O processamento de dados visuais requer recursos que excedem as capacidades de dispositivos embarcados. Uma abordagem para contornar tal dificuldade consiste na redução da complexidade dos algoritmos e

de técnicas de descorrelação de dados (Blahut, 2010; Britanak, Yip e Rao, 2007; Sayood, 2017). Tais técnicas exploram a redundância intrínseca dos sinais, nesse caso, imagens, permitindo representar a informação de forma mais compacta, com menor custo de armazenamento e processamento, sem perda significativa de qualidade (Richardson, 2003). Esse processo é realizado por transformações discretas.

No processamento digital de sinais, as transformadas discretas têm papel de destaque como ferramentas para compressão e análise de dados (Ahmed e Rao, 1975). Dentre estas, as transformadas com núcleo trigonométrico possuem vasta aplicação, incluindo a transformada discreta de Fourier (DFT) e a transformada discreta do cosseno (DCT) (Briggs e Henson, 1995; Bracewell, 1986; Britanak, Yip e Rao, 2007). Especialmente a DCT tem larga aplicação em compressão de imagens, sendo utilizada em padrões como JPEG para imagens naturais (Wallace, 1992), em padrões de vídeo como MPEG (Gall, 1991) e os codificadores de nova geração H.264/AVC (Wiegand *et al.*, 2003) e HEVC (Sullivan *et al.*, 2012). O amplo interesse pela DCT deve-se à sua capacidade de aproximar o desempenho da Transformada de Karhunen-Loève (KLT) (Ahmed, Natarajan e Rao, 1974). A KLT possui propriedades de descorrelação e compactação de energia quando o sinal é modelado por meio de um processo markoviano de primeira ordem altamente correlacionado (Rao e Yip, 1990; Clarke, 1981).

Em imagens naturais, a redundância está presente na forma de correlação espacial entre pixels vizinhos (Gonzalez e Woods, 2018). Esse tipo de imagem é modelado como um processo markoviano de primeira ordem com alta correlação entre pixels vizinhos, o que torna a DCT particularmente eficiente para este domínio de aplicação (Britanak, Yip e Rao, 2007). A DCT concentra a maior parte da informação em poucos coeficientes de baixa frequência. Essa propriedade possibilita o descarte de componentes menos significativos, com perda na qualidade perceptual da imagem (Jain, 1989) indistinguível ao olho humano.

A DCT tem diversas aplicações no contexto industrial. A literatura é vasta em trabalhos que exploram as propriedades da DCT para detecção de defeitos em superfícies texturizadas, onde a transformada permite separar padrões regulares de texturas de anomalias localizadas (Perng e Chen, 2011). A aplicação da DCT na identificação de defeitos de forma em superfícies usadas também foi investigada na literatura (Lecompte, Legoff e Hascoet, 2010). Essas aplicações evidenciam o potencial da DCT como ferramenta para sistemas de rastreamento visual.

Entretanto, a implementação da DCT em sistemas embarcados e dispositivos de baixo consumo energético está sujeita a restrições de complexidade computacional que tornam sua aplicação pela forma exata proibitiva (Britanak, Yip e Rao, 2007). A DCT unidimensional de comprimento N , pela formulação direta, requer N^2 operações de multiplicação e $N(N - 1)$ adições, resultando em complexidade $O(N^2)$. No caso bidimensional, considerando a separabilidade da transformada — em que a DCT unidimensional é aplicada separadamente às linhas e colunas do bloco — o cômputo de um bloco $N \times N$ requer $2N^3$ multiplicações e $2N^2(N - 1)$ adições, resultando em complexidade $O(N^3)$ (Blahut, 2010).

Algoritmos rápidos baseados em fatorações matriciais reduzem a complexidade

para $O(N \log N)$ e $O(N^2 \log N)$ para DCT unidimensional e bidimensional, respectivamente (Blahut, 2010). Entretanto, ainda assim, as operações de multiplicações em ponto flutuante demandam elevado custo energético em implementações embarcadas, tornando a implementação direta proibitiva nesses dispositivos (Loeffler, Ligtenberg e Moschytz, 1989). Nesse contexto, o desenvolvimento de transformadas aproximadas surge como uma alternativa para reduzir a complexidade computacional dos algoritmos.

Aproximações matriciais devem preservar as propriedades desejáveis da transformada exata enquanto reduzem sua complexidade aritmética (Cintra e Bayer, 2011a). Aproximações da DCT substituem a matriz de transformação original por matrizes compostas de números inteiros, chamados de multiplicandos triviais. A busca por aproximações da DCT é um problema de otimização em que se busca identificar matrizes de baixa complexidade que maximizem a similaridade com a transformada exata segundo métricas apropriadas (Cintra e Bayer, 2011b).

Neste trabalho, é proposto um método de aproximação da DCT baseado numa relação algébrica entre esta e a DFT (Blahut, 2010). O método consiste em (i) desenvolver aproximações da DFT e (ii) convertê-las em aproximações da DCT, mantendo a baixa complexidade.

A motivação para este trabalho origina-se da demanda industrial por sistemas de rastreamento visual em dispositivos de baixo consumo energético. Sob a perspectiva da Engenharia de Produção, este trabalho contribui para a redução de restrições tecnológicas que limitam a adoção de sistemas de controle de qualidade, alinhando-se à filosofia de melhoria contínua, redução de desperdícios e aumento da produtividade.

1.2 TÓPICO DE INVESTIGAÇÃO

A DCT possui complexidade computacional que torna a execução em dispositivos embarcado e microcontroladores proibitiva. Nesse contexto, o presente trabalho investiga a viabilidade de desenvolver aproximações da DCT para múltiplos comprimentos de bloco, derivadas de aproximações da DFT por minimização de diferença angular, com complexidade aritmética suficientemente reduzida para implementação em dispositivos embarcados, preservando qualidade adequada para aplicações de inspeção visual e rastreamento em linhas de produção industrial.

1.3 ESTADO DA ARTE

Nas últimas décadas, aproximações de baixa complexidade computacional da DCT têm sido propostas para aplicações com restrições de hardware, consumo de energia ou processamento em tempo real, como sistemas embarcados.

A primeira aproximação da DCT nessa linha de pesquisa foi a SDCT (*signed DCT*), proposta por Haweel (2001). Nesse método, aplica-se a função sinal a cada elemento da matriz exata da DCT, produzindo uma aproximação composta apenas por elementos em $\{0, \pm 1\}$. Por

sua simplicidade e baixo custo computacional, a SDCT é considerada um trabalho pioneiro na área.

Seguindo uma ideia semelhante, em (Bayer e Cintra, 2010) foi proposta a MRDCT (*modified rounded DCT*), obtida pelo arredondamento dos elementos da DCT para o inteiro mais próximo. Posteriormente, Cintra e Bayer (2011a) propõem a RDCT (*rounded DCT*), abordagem que multiplica a matriz da DCT por um fator de escala antes do arredondamento.

Em (Tablada, Cintra e Bayer, 2014) é introduzida uma família de aproximações baseadas em funções inteiras — piso, teto, truncamento — aplicadas à matriz exata escalada por um fator α , isto é, $\hat{\mathbf{C}} = \text{int}(\alpha \cdot \mathbf{C})$. O espaço de busca foi estendido para $\{0, \pm 1, \pm 2, \pm 3\}$. Diversas aproximações foram identificadas, entre elas a $\hat{\mathbf{C}}_{\text{TCB}_6}$, considerada neste trabalho.

Aproximações multiparamétricas, que permitem o ajuste de parâmetros para balancear complexidade e desempenho, foram propostas em (Bouguezel, Ahmad e Swamy, 2008). No método, entradas da matriz são tratadas como parâmetros livres cuja escolha define diferentes níveis de complexidade e desempenho; o caso adotado neste trabalho é denotado $\hat{\mathbf{C}}_{\text{BAS-2008}}$.

Em (Lengwehasatit e Ortega, 2004) foi proposta a aproximação $\hat{\mathbf{C}}_{\text{LO}}$ com complexidade escalável, em que a transformada pode ser computada em diferentes níveis de precisão conforme a demanda da aplicação.

Mais recentemente, em (Oliveira *et al.*, 2018) foi proposto um método guloso, em que cada linha da DCT é otimizada independentemente. O critério de aproximação adotado é a minimização do ângulo entre o vetor candidato e a linha correspondente da matriz exata. As variáveis de decisão são vetores de oito componentes nos conjuntos $S_1 = \{0, \pm 1\}$ e $S_2 = \{0, \pm 1, \pm 2\}$, gerando, respectivamente, as aproximações $\hat{\mathbf{C}}_{\text{OCBSML}_1}$ e $\hat{\mathbf{C}}_{\text{OCBSML}_2}$.

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Desenvolver aproximações da DCT para comprimentos de bloco $N = 4$ e $N = 8$, a partir de aproximações da DFT de comprimentos $N = 16$ e $N = 32$, visando aplicação em sistemas de rastreamento para linhas de produção industrial com restrições de recursos computacionais.

1.4.2 Objetivos Específicos

Os objetivos específicos que norteiam o desenvolvimento deste trabalho são:

- Implementar o método de minimização da diferença angular para obtenção de aproximações de baixa complexidade da DFT, utilizando inteiros gaussianos $a + bj$ com $a, b \in \{0, \pm 1, \pm 2, \pm 4\}$ como candidatos;
- Obter aproximações da DCT de comprimentos $N = 4$ e $N = 8$ a partir das aproximações da DFT, empregando uma relação de conversão entre transformadas;

-
- Avaliar as aproximações propostas da DCT segundo figuras de mérito clássicas na literatura, como (i) erro total de energia, (ii) erro médio quadrático, (iii) ganho de codificação unificado, (iv) eficiência da transformada e (v) desvio de ortogonalidade;
 - Comparar as aproximações desenvolvidas com métodos do estado da arte, avaliando a complexidade aritmética;
 - Avaliar o desempenho das aproximações em compressão de imagens através de experimentos baseados no padrão JPEG, em ambiente computacional simulado, utilizando métricas de qualidade como MSE, UQI, PSNR e SSIM.

2 SISTEMA DE RASTREAMENTO VISUAL

2.1 INTRODUÇÃO

O rastreamento visual (*visual tracking*) é definido como o processo de estimar a trajetória de um ou mais objetos no plano da imagem ao longo de uma sequência de vídeo, atribuindo-lhes identificadores consistentes entre quadros consecutivos (Yilmaz, Javed e Shah, 2006). Em ambientes industriais, essa tarefa viabiliza a análise de eventos discretos (Brailsford, Churilov e Dangerfield, 2014). Entre os eventos discretos de maior interesse, pode-se citar o tempo gasto pelo trânsito de uma peça na linha produtiva, fundamental para avaliação estatística do desempenho da linha de montagem.

A execução do rastreamento enfrenta desafios inerentes ao cenário industrial: (i) perda de informação decorrente da projeção de cenas tridimensionais em imagens bidimensionais; (ii) ruído de aquisição e variações de iluminação no ambiente fabril; (iii) oclusões parciais e movimentos complexos dos objetos; e (iv) a exigência de processamento em tempo real, compatível com a cadência da linha (Yilmaz, Javed e Shah, 2006; Beyerer, León e Frese, 2016; Marvasti-Zadeh *et al.*, 2022). Somam-se a esses desafios os requisitos de baixa latência, baixo consumo energético e execução em *hardware* embarcado.

2.2 MODELO DE RASTREAMENTO VISUAL

O modelo de rastreamento adotado como referência neste trabalho é o esquema incremental baseado na DCT 3D (3D-DCT), proposto por Li *et al.* (2013), que combina a eficiência espectral da transformada com um mecanismo bayesiano de inferência do estado do objeto. A representação da aparência é construída não apenas sobre as dimensões espaciais (x, y) da região rastreada, mas também sobre a dimensão temporal t , empilhando amostras de aparências recentes em um tensor tridimensional.

2.2.1 Formulação por filtro de partículas

Seja $\mathbf{z}_t$ o estado oculto do objeto no instante t , parametrizando posição, escala e rotação da janela de rastreamento, e seja I_t o quadro corrente. O problema de rastreamento consiste em estimar a distribuição posterior $p(\mathbf{z}_t | I_{1:t})$. Na ausência de forma fechada para essa distribuição, emprega-se um filtro de partículas (algoritmo de condensação) (Isard e Blake, 1998), que a aproxima por um conjunto ponderado de N_p amostras (partículas) $\{\mathbf{z}_t^{(i)}\}_{i=1}^{N_p}$. Cada partícula corresponde a uma hipótese de localização do objeto e é avaliada por uma função de verossimilhança $p(I_t | \mathbf{z}_t^{(i)})$, cuja construção é o núcleo do método.

2.2.2 Tensor de aparência e transformada 3D-DCT

Para cada partícula, a região de imagem correspondente é recortada e redimensionada para um bloco de dimensão fixa $N \times N$, doravante denominado imagem-teste. A esse bloco associa-se um conjunto de k vizinhos mais próximos, selecionados em um conjunto de aparências confirmadas em instantes anteriores — o histórico do rastreador. A imagem-teste e seus k vizinhos são concatenados ao longo da dimensão temporal, formando um tensor tridimensional $\mathcal{X}$ dado por:

$$\mathcal{X} \in \mathbb{R}^{N \times N \times (k+1)}. \quad (2.1)$$

Aplica-se, então, a 3D-DCT a $\mathcal{X}$, produzindo o tensor de coeficientes espectrais $\mathcal{Y}$. Através da propriedade de separabilidade da transformada, essa operação decompõe-se em uma DCT bidimensional aplicada a cada fatia $N \times N$ (domínio espacial), seguida de uma DCT unidimensional ao longo da dimensão temporal (Li *et al.*, 2013). Pela propriedade de compactação de energia, quando a imagem-teste é efetivamente próxima do modelo de aparência, a maior parte da energia de $\mathcal{Y}$ concentra-se em seus coeficientes de baixa frequência.

2.2.3 Filtragem espectral

A compactação é quantificada pelo descarte dos coeficientes de alta frequência de $\mathcal{Y}$, realizado por meio da aplicação de uma máscara binária $\mathbf{M}$ que retém apenas os coeficientes correspondentes às baixas frequências. O tensor resultante

$$\mathcal{Y}_c = \mathbf{M} \odot \mathcal{Y}$$

é a representação compacta da aparência e o operador $\odot$ denota multiplicação elemento a elemento entre tensores de mesma dimensão (Oppenheim e Schaffer, 2010). Em seguida, aplica-se a DCT 3D inversa, denotada por 3D-IDCT, sobre $\mathcal{Y}_c$, recuperando um tensor reconstruído $\hat{\mathcal{X}}$. A fatia de $\hat{\mathcal{X}}$ associada à imagem-teste é comparada à imagem-teste original, resultando no erro de reconstrução dado por:

$$e^{(i)} = \left\| \mathcal{X}_{\text{teste}} - \hat{\mathcal{X}}_{\text{teste}} \right\|_F^2, \quad (2.2)$$

em que $\|\cdot\|_F$ denota a norma de Frobenius. A verossimilhança (Li *et al.*, 2013) associada à partícula i é então definida como:

$$p(I_t | \mathbf{z}_t^{(i)}) \propto \exp\left(-e^{(i)} / \sigma^2\right), \quad (2.3)$$

sendo σ^2 um parâmetro de dispersão. Partículas cuja imagem-teste é bem reconstruída a partir apenas das baixas frequências do subespaço recebem alta verossimilhança; as demais, recebem baixa verossimilhança.

2.2.4 Decisão e atualização incremental

A partícula com maior verossimilhança — equivalentemente, com menor erro de reconstrução — fornece a estimativa da posição do objeto no instante t . Sua aparência é incorporada ao histórico, substituindo a amostra mais antiga, o que permite a adaptação do modelo a variações graduais de iluminação, rotação e deformação. A principal contribuição de Li *et al.* (2013) reside exatamente neste ponto: a separabilidade da 3D-DCT permite atualizar os coeficientes da DCT unidimensional temporal de forma incremental a cada novo quadro, evitando o reprocessamento completo do histórico e viabilizando o rastreamento em tempo real.

2.2.5 Representação simplificada

A Figura 1 sintetiza a arquitetura do sistema de rastreamento visual adotada como referência neste trabalho, proposta por Li *et al.* (2013). O fluxo contém quatro subsistemas: aquisição e pré-processamento; geração e preparação de candidatos; núcleo espectral (transformada, descarte e reconstrução) e decisão.

As aproximações da DCT desenvolvidas neste trabalho são candidatas à substituição das matrizes exatas da DCT e IDCT por suas versões aproximadas de baixa complexidade (blocos em destaque na figura). Essa substituição preserva a estrutura do algoritmo original, mas elimina multiplicações em ponto flutuante, reduzindo o custo aritmético por quadro.

2.3 COMPRESSÃO DE IMAGENS EM RASTREAMENTO VISUAL

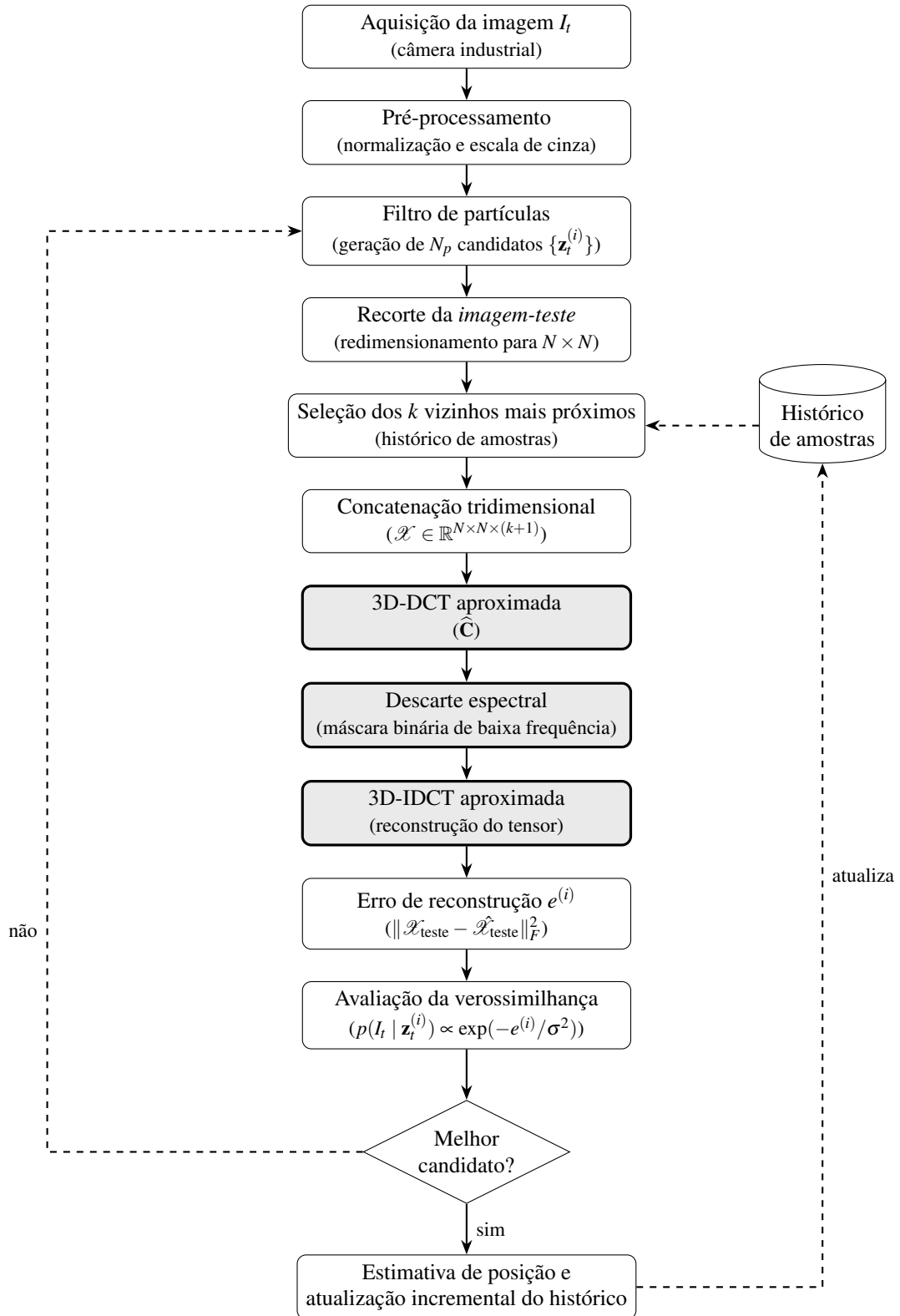
Uma sequência de vídeo é uma série de imagens digitais bidimensionais captadas em intervalos de tempo regulares. Cada unidade dessa sequência — *frame* ou quadro — é uma imagem como qualquer outra: uma matriz de intensidades de pixels definida em um plano bidimensional (Gonzalez e Woods, 2018).

No contexto do sistema de rastreamento descrito na Seção 2.2, a cada instante t , cada partícula $\mathbf{z}_t^{(i)}$ gera uma região de imagem recortada do quadro corrente I_t e redimensionada para um bloco $N \times N$ — a imagem-teste. Esse bloco é, para todos os fins, uma imagem. De modo análogo, as amostras do histórico do rastreador, selecionadas entre quadros anteriores, são igualmente blocos $N \times N$ extraídos de *frames* passados.

A 3D-DCT aplicada ao tensor $\mathcal{X} \in \mathbb{R}^{N \times N \times (k+1)}$ é separável: pela propriedade de separabilidade das transformadas multidimensionais, ela se decompõe em uma DCT bidimensional aplicada na dimensão espacial $N \times N$, seguida de uma DCT unidimensional ao longo da dimensão temporal (Li *et al.*, 2013; Oppenheim e Schaffer, 2010). A componente bidimensional dessa operação é, portanto, a DCT 2D aplicada a imagens — o mesmo contexto avaliado nos experimentos de compressão de imagens deste trabalho.

Desse modo, qualquer aproximação de baixa complexidade para a DCT que demonstre bom desempenho em compressão de imagens preserva essa qualidade quando aplicada aos *frames* de um sistema de rastreamento visual, pois os *frames* são imagens 2D submetidas ao mesmo

Figura 1 – Arquitetura de um sistema de rastreamento.

Fonte: Adaptado de Li *et al.* (2013).

processamento. Essa relação justifica a adoção dos experimentos de compressão de imagens como critério de avaliação e comparação das aproximações propostas antes de sua integração no sistema de rastreamento.

2.4 RASTREAMENTO E INDÚSTRIA 4.0

A Indústria 4.0 caracteriza-se pela integração entre sistemas físicos e cibernéticos e pela execução de análises preditivas no chão de fábrica (Kalsoom *et al.*, 2020). Nesse contexto, o controle de qualidade deixou de ser predominantemente reativo e passou a incorporar abordagens preditivas, viabilizadas pela adoção de sistemas de rastreamento visual (Johanesa, Equeter e Mahmoudi, 2024).

Os principais requisitos das soluções industriais de inspeção e rastreamento convergem para três pontos: *(i)* baixa latência, compatível com a cadência da linha; *(ii)* alta credibilidade, com taxas controladas de falso positivo e falso negativo; e *(iii)* execução em *hardware* de baixo custo e baixo consumo (Beyerer, León e Frese, 2016; Kalsoom *et al.*, 2020). Esse terceiro requisito, em particular, impõe restrições sobre a complexidade aritmética dos algoritmos empregados e motiva diretamente o uso de aproximações matriciais da DCT, que compõem o núcleo de processamento.

3 TRANSFORMADA DISCRETA DE FOURIER

3.1 FUNDAMENTAÇÃO MATEMÁTICA

A análise de sinais e sistemas no domínio da frequência é uma ferramenta fundamental na engenharia. Enquanto a transformada de Fourier contínua é empregada para sinais analógicos definidos em um intervalo de tempo infinito, o processamento digital de sinais exige uma representação que seja discreta tanto no tempo quanto na frequência, e de duração finita, empregando a DFT. A DFT mapeia uma sequência finita de N amostras no domínio do tempo para uma sequência de N coeficientes no domínio da frequência, permitindo a análise espectral e a implementação eficiente de filtros digitais.

Esta seção apresenta a fundamentação matemática requerida para o entendimento da DFT.

3.1.1 Inteiros Gaussianos

Inteiros gaussianos são números complexos cujas partes real e imaginária são inteiros usuais.

Definição 1. Um inteiro gaussiano z é um número complexo da forma

$$z = a + bj, \quad (3.1)$$

em que $j \triangleq \sqrt{-1}$ e $a, b \in \mathbb{Z}$.

O conjunto de todos os inteiros gaussianos é denotado por $\mathbb{Z}[j]$.

3.1.2 Módulo e argumento

Além da representação cartesiana, os números complexos podem ser expressos em coordenadas polares (r, θ) , em que r denota a distância do ponto à origem e θ representa o ângulo formado com o semieixo real positivo. A conversão entre as representações cartesiana e polar é estabelecida pelas seguintes relações (Oppenheim e Schaffer, 2014):

$$r = |z| = \sqrt{x^2 + y^2}, \quad (3.2)$$

$$\theta = \angle z = \arctan\left(\frac{y}{x}\right), \quad (3.3)$$

em que $|z|$ é o módulo (ou magnitude) de z , e θ é o argumento (fase ou ângulo) de z . Reciprocamente, a conversão de coordenadas polares para cartesianas é dada por:

$$x = r \cos(\theta), \quad (3.4)$$

$$y = r \sin(\theta). \quad (3.5)$$

Dessas relações, obtém-se a forma trigonométrica do número complexo:

$$z = r [\cos(\theta) + j \sin(\theta)]. \quad (3.6)$$

A partir das definições apresentadas, podem ser verificadas as seguintes identidades envolvendo o conjugado complexo (Smith, 2002):

$$z + \bar{z} = 2\Re\{z\}, \quad (3.7)$$

$$z - \bar{z} = 2j\Im\{z\}, \quad (3.8)$$

$$z \cdot \bar{z} = |z|^2. \quad (3.9)$$

3.1.3 Identidade de Euler

A identidade de Euler estabelece uma relação fundamental entre as funções exponencial e trigonométrica, sendo a base para a representação polar de números complexos (Bracewell, 2000):

$$e^{j\theta} = \cos(\theta) + j\sin(\theta). \quad (3.10)$$

Esta identidade pode ser demonstrada por meio da expansão em série de Taylor das funções exponencial, seno e cosseno. Considerando que:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, \quad (3.11)$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots, \quad (3.12)$$

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots, \quad (3.13)$$

e substituindo $x = j\theta$ em (3.11), obtém-se, após separar os termos reais e imaginários, a identidade de Euler apresentada em (3.10). A partir da identidade de Euler, podem ser derivadas expressões para as funções trigonométricas em termos de exponenciais complexas (Oppenheim e Schafer, 2014):

$$\cos(\theta) = \frac{e^{j\theta} + e^{-j\theta}}{2}, \quad (3.14)$$

$$\sin(\theta) = \frac{e^{j\theta} - e^{-j\theta}}{2j}. \quad (3.15)$$

Essas relações são particularmente úteis na análise de sinais, pois permitem expressar senoides reais como combinações de exponenciais complexas. A equação (3.14) será útil na demonstração do teorema utilizado no Capítulo 6.

3.1.4 Raízes da Unidade

As raízes N -ésimas da unidade são os N números complexos que satisfazem a equação $z^N = 1$. Essas raízes podem ser expressas na forma exponencial como (Oppenheim e Schafer, 2014):

$$\omega_N^k = e^{-j\frac{2\pi k}{N}}, \quad k = 0, 1, \dots, N-1, \quad (3.16)$$

em que $\omega_N \triangleq e^{-j\frac{2\pi}{N}}$ é chamada de raiz N -ésima primitiva da unidade ou fator de rotação (em inglês, *twiddle factor*).

As raízes da unidade distribuem-se uniformemente sobre o círculo unitário no plano complexo, espaçadas por ângulos de $\frac{2\pi}{N}$ radianos. Cada potência ω_N^k divide o círculo unitário no plano complexo em segmentos de arco com aberturas angulares iguais.

Uma propriedade fundamental das raízes da unidade é sua periodicidade (Brigham, 1988):

$$\omega_N^{k+N} = \omega_N^k, \quad (3.17)$$

o que decorre diretamente da relação $(\omega_N)^N = e^{-j2\pi} = 1$. Outra propriedade importante é a simetria:

$$\omega_N^{k+N/2} = -\omega_N^k, \quad (3.18)$$

válida para N par, que é extensivamente utilizada em algoritmos rápidos para o cálculo da DFT.

3.2 DEFINIÇÃO

A DFT é uma transformação linear que mapeia uma sequência discreta de N amostras no domínio do tempo em uma sequência de N coeficientes no domínio da frequência (Oppenheim e Schaffer, 2014). Seja $x[n]$ uma sequência de N amostras complexas, com $n = 0, 1, \dots, N-1$. A DFT de $x[n]$, denotada por $X[k]$, é definida por:

$$X[k] \triangleq \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi nk}{N}}, \quad k = 0, 1, \dots, N-1. \quad (3.19)$$

Utilizando a notação $\omega_N = e^{-j\frac{2\pi}{N}}$, a DFT pode ser reescrita de forma mais compacta:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot \omega_N^{nk}, \quad k = 0, 1, \dots, N-1. \quad (3.20)$$

A transformada inversa (IDFT), que permite recuperar a sequência temporal a partir de sua representação espectral, é dada por (Oppenheim e Schaffer, 2014):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{j\frac{2\pi nk}{N}}, \quad n = 0, 1, \dots, N-1. \quad (3.21)$$

A exponencial complexa $e^{-j\frac{2\pi nk}{N}}$ é núcleo (ou *kernel*) da transformada. A identidade de Euler (3.10) permite decompor este núcleo em suas componentes cosseno e seno:

$$e^{-j\frac{2\pi nk}{N}} = \cos\left(\frac{2\pi nk}{N}\right) - j \sin\left(\frac{2\pi nk}{N}\right). \quad (3.22)$$

Do ponto de vista geométrico, o somatório na definição da DFT pode ser interpretado como o produto interno entre o sinal de entrada $x[n]$ e as senoides complexas amostradas (Smith, 2002). Cada coeficiente $X[k]$ quantifica a amplitude e a fase da componente senoidal na frequência angular $\omega_k = \frac{2\pi k}{N}$ presente no sinal $x[n]$.

3.2.1 Representação Matricial

A DFT mapeia um sinal de entrada de N pontos,

$$\mathbf{x} \triangleq \begin{bmatrix} x[0] & x[1] & \cdots & x[N-1] \end{bmatrix}^\top, \quad (3.23)$$

em um sinal de saída,

$$\mathbf{X} \triangleq \begin{bmatrix} X[0] & X[1] & \cdots & X[N-1] \end{bmatrix}^\top, \quad (3.24)$$

através do seguinte produto matricial (Rao e Yip, 2018):

$$\mathbf{X} = \mathbf{F}_N \mathbf{x}, \quad (3.25)$$

em que $\mathbf{F}_N$ é a matriz DFT de dimensão $N \times N$, definida por:

$$\mathbf{F}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_N & \omega_N^2 & \cdots & \omega_N^{N-1} \\ 1 & \omega_N^2 & \omega_N^4 & \cdots & \omega_N^{2(N-1)} \\ 1 & \omega_N^3 & \omega_N^6 & \cdots & \omega_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \cdots & \omega_N^{(N-1)(N-1)} \end{bmatrix}. \quad (3.26)$$

O elemento na k -ésima linha n -ésima coluna da matriz $\mathbf{F}_N$ é dado por:

$$[\mathbf{F}_N]_{k,n} = \omega_N^{kn} = e^{-j\frac{2\pi kn}{N}}. \quad (3.27)$$

A matriz $\mathbf{F}_N$ é uma matriz de Vandermonde (Briggs e Henson, 1995). A IDFT, por sua vez, pode ser expressa matricialmente como:

$$\mathbf{x} = \frac{1}{N} \bar{\mathbf{F}}_N \mathbf{X} = \mathbf{F}_N^{-1} \mathbf{X}, \quad (3.28)$$

em que $\bar{\mathbf{F}}_N$ denota o conjugado complexo elemento a elemento de $\mathbf{F}_N$. A título de exemplo, a matriz DFT para $N = 4$ é dada por:

$$\mathbf{F}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}, \quad (3.29)$$

uma vez que $\omega_4 = e^{-j\frac{\pi}{2}} = -j$.

3.2.2 Ortogonalidade

Uma propriedade fundamental da matriz DFT é sua ortogonalidade. As linhas (ou colunas) da matriz $\mathbf{F}_N$ formam um conjunto de vetores ortogonais (Briggs e Henson, 1995).

Teorema 1 (Ortogonalidade da DFT). *Sejam k e m inteiros e N um inteiro positivo. Então:*

$$\sum_{n=0}^{N-1} e^{j\frac{2\pi nk}{N}} e^{-j\frac{2\pi nm}{N}} = N \cdot \delta[k-m], \quad (3.30)$$

em que $\delta[\cdot]$ é o delta de Kronecker, definido por:

$$\delta[k-m] = \begin{cases} 1, & \text{se } k = m, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.31)$$

Demonstração. O lado esquerdo de (3.30) pode ser reescrito como:

$$\sum_{n=0}^{N-1} e^{j\frac{2\pi n(k-m)}{N}} = \sum_{n=0}^{N-1} \left(\omega_N^{-(k-m)} \right)^n. \quad (3.32)$$

Consideram-se dois casos. Se $k \equiv m \pmod{N}$, então $\omega_N^{-(k-m)} = 1$, e o somatório torna-se:

$$\sum_{n=0}^{N-1} 1^n = N. \quad (3.33)$$

Se $k \not\equiv m \pmod{N}$, então $\omega_N^{-(k-m)} \neq 1$, e o somatório constitui uma série geométrica finita:

$$\sum_{n=0}^{N-1} \left(\omega_N^{-(k-m)} \right)^n = \frac{1 - \left(\omega_N^{-(k-m)} \right)^N}{1 - \omega_N^{-(k-m)}} = \frac{1 - e^{-j2\pi(k-m)}}{1 - \omega_N^{-(k-m)}} = \frac{1 - 1}{1 - \omega_N^{-(k-m)}} = 0. \quad (3.34)$$

□

Essa propriedade de ortogonalidade garante que a matriz $\mathbf{F}_N$ seja invertível. De fato, pode-se demonstrar que (Oppenheim e Schaffer, 2014):

$$\mathbf{F}_N^{-1} = \frac{1}{N} \bar{\mathbf{F}}_N = \frac{1}{N} \mathbf{F}_N^*. \quad (3.35)$$

Se a matriz $\mathbf{F}_N$ for normalizada pelo fator $\frac{1}{\sqrt{N}}$, a matriz resultante $\tilde{\mathbf{F}}_N = \frac{1}{\sqrt{N}} \mathbf{F}_N$ é unitária, ou seja:

$$\tilde{\mathbf{F}}_N^* \tilde{\mathbf{F}}_N = \tilde{\mathbf{F}}_N \tilde{\mathbf{F}}_N^* = \mathbf{I}_N, \quad (3.36)$$

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

3.3 PROPRIEDADES

A DFT possui diversas propriedades que a tornam uma ferramenta poderosa para a análise e o processamento de sinais (Oppenheim e Schaffer, 2014; Brigham, 1988).

3.3.1 Linearidade

A DFT é uma transformação linear. Sejam $x_1[n]$ e $x_2[n]$ duas sequências com DFTs $X_1[k]$ e $X_2[k]$, respectivamente. Então, para quaisquer constantes $a, b \in \mathbb{C}$:

$$\text{DFT}\{a \cdot x_1[n] + b \cdot x_2[n]\} = a \cdot X_1[k] + b \cdot X_2[k]. \quad (3.37)$$

3.3.2 Deslocamento Circular no Tempo

Um deslocamento circular de n_0 posições na sequência temporal corresponde a uma multiplicação por um fator de fase linear no domínio da frequência:

$$\text{DFT}\{x[(n - n_0) \bmod N]\} = e^{-j\frac{2\pi kn_0}{N}} X[k] = \omega_N^{kn_0} X[k]. \quad (3.38)$$

3.3.3 Deslocamento Circular na Frequência

Analogamente, um deslocamento circular no domínio da frequência corresponde a uma modulação no domínio do tempo:

$$\text{DFT}\{x[n] \cdot e^{j\frac{2\pi k_0 n}{N}}\} = X[(k - k_0) \bmod N]. \quad (3.39)$$

3.3.4 Dualidade

Se $x[n] \xleftrightarrow{\text{DFT}} X[k]$, então:

$$X[n] \xleftrightarrow{\text{DFT}} N \cdot x[-k \bmod N]. \quad (3.40)$$

3.3.5 Convolução Circular

A convolução circular no domínio do tempo corresponde à multiplicação ponto a ponto no domínio da frequência:

$$\text{DFT}\{x_1[n] \circledast x_2[n]\} = X_1[k] \cdot X_2[k], \quad (3.41)$$

em que $\circledast$ denota a convolução circular de período N , definida por:

$$x_1[n] \circledast x_2[n] = \sum_{m=0}^{N-1} x_1[m] \cdot x_2[(n - m) \bmod N]. \quad (3.42)$$

Reciprocamente, a multiplicação no domínio do tempo corresponde à convolução circular no domínio da frequência:

$$\text{DFT}\{x_1[n] \cdot x_2[n]\} = \frac{1}{N} X_1[k] \circledast X_2[k]. \quad (3.43)$$

3.3.6 Teorema de Parseval

O teorema de Parseval estabelece a conservação de energia entre os domínios do tempo e da frequência (Bracewell, 2000):

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2. \quad (3.44)$$

Este resultado mostra que a energia total do sinal, calculada no domínio do tempo como a soma dos quadrados das magnitudes das amostras, é proporcional à energia calculada no domínio da frequência.

3.3.7 Propriedades de Simetria

Para uma sequência $x[n]$ de valores reais, a DFT $X[k]$ exibe simetria conjugada (ou simetria hermitiana) (Oppenheim e Schafer, 2014):

$$X[N - k] = \overline{X[k]}, \quad k = 0, 1, \dots, N - 1. \quad (3.45)$$

Esta propriedade implica que $|X[N - k]| = |X[k]|$ (simetria de magnitude) e $\angle X[N - k] = -\angle X[k]$ (antissimetria de fase).

3.3.8 Complexidade Computacional

A computação direta da DFT, conforme definida em (3.19), requer, para cada coeficiente $X[k]$, a realização de N multiplicações complexas e $N - 1$ adições complexas. Como são calculados N coeficientes, a complexidade computacional total da DFT direta é da ordem de $\mathcal{O}(N^2)$ (Oppenheim e Schafer, 2014).

Considerando que cada multiplicação complexa $(a + jb)(c + jd) = (ac - bd) + j(ad + bc)$ requer quatro multiplicações reais e duas adições reais, e cada adição complexa requer duas adições reais, o cálculo direto da DFT de uma sequência de N pontos demanda, no máximo, $4N^2$ multiplicações reais e $N(4N - 2)$ adições reais (Brigham, 1988).

Esta complexidade torna-se proibitiva para valores elevados de N . Por exemplo, para $N = 1024$, são necessárias aproximadamente 4×10^6 multiplicações reais. Esta limitação motivou o desenvolvimento de algoritmos rápidos, como o algoritmo de Cooley-Tukey, que reduzem a complexidade para $\mathcal{O}(N \log N)$, viabilizando a aplicação da DFT em diversos sistemas de tempo real (Cooley e Tukey, 1965).

3.3.9 Interpretação Geométrica

Uma perspectiva alternativa e geometricamente intuitiva para compreender a DFT consiste em interpretar os sinais discretos como vetores em um espaço vetorial de dimensão N (Smith, 2002). Nessa interpretação, uma sequência $x[n]$ de comprimento N é representada como um vetor $\mathbf{x} \in \mathbb{C}^N$, em que cada amostra $x[n]$ corresponde a uma coordenada. O produto interno entre dois vetores complexos $\mathbf{x}$ e $\mathbf{y}$ em $\mathbb{C}^N$ é definido por:

$$\langle \mathbf{x}, \mathbf{y} \rangle \triangleq \sum_{n=0}^{N-1} x[n] \cdot \overline{y[n]}. \quad (3.46)$$

Definindo os vetores de base senoidal complexa como:

$$\mathbf{s}_k[n] \triangleq e^{j\frac{2\pi kn}{N}}, \quad n = 0, 1, \dots, N - 1, \quad k = 0, 1, \dots, N - 1, \quad (3.47)$$

o coeficiente da DFT pode ser expresso como o produto interno entre o sinal $\mathbf{x}$ e o vetor de base $\mathbf{s}_k$ (Smith, 2002):

$$X[k] = \langle \mathbf{x}, \mathbf{s}_k \rangle = \sum_{n=0}^{N-1} x[n] \cdot \overline{s_k[n]} = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi kn}{N}}. \quad (3.48)$$

Sob essa perspectiva, a DFT calcula a projeção do sinal de entrada sobre cada um dos vetores de base senoidal, determinando “quanto” de cada componente de frequência está presente no sinal e em que fase. A propriedade de ortogonalidade demonstrada na Seção 3.2.2 garante que os vetores $\{\mathbf{s}_k\}_{k=0}^{N-1}$ formam uma base ortogonal para $\mathbb{C}^N$. Consequentemente, qualquer sinal pode ser unicamente representado como uma combinação linear dessas senoides complexas.

4 TRANSFORMADA DISCRETA DO COSSENO

4.1 INTRODUÇÃO

A DCT foi introduzida por Ahmed, Natarajan e Rao (1974) como uma alternativa computacionalmente viável à Transformada de Karhunen-Loève (KLT). A KLT é reconhecida como a transformada ótima para descorrelação e compactação de energia de sinais. Entretanto, a dependência dos dados de entrada inviabiliza o desenvolvimento de algoritmos rápidos para sua implementação (Britanak, Yip e Rao, 2007). A DCT contorna tal limitação ao apresentar uma forma fechada que independe do sinal de entrada, mantendo propriedades próximas às da KLT para sinais altamente correlacionados, como imagens naturais (Rao e Yip, 1990). A DCT tornou-se uma das transformadas mais importantes em processamento digital de sinais, sendo amplamente utilizada em padrões de compressão de imagens e vídeos, como JPEG (Wallace, 1992), MPEG (Gall, 1991) e HEVC (Sullivan *et al.*, 2012).

A Seção 4.2 introduz a Transformada de Karhunen-Loève e sua caracterização para sinais markovianos de primeira ordem. Na Seção 4.3.2, demonstra-se que a DCT tipo II (DCT-II) emerge como caso assintótico da KLT quando o coeficiente de correlação do sinal tende à unidade, justificando suas propriedades de descorrelação e compactação de energia.

4.2 A TRANSFORMADA DE KARHUNEN-LOÈVE

A Transformada de Karhunen-Loève (KLT), também conhecida como Análise de Componentes Principais (PCA) ou Transformada de Hotelling (Hotelling, 1933; Pearson, 1901), é uma transformação linear que possui a propriedade de descorrelacionar completamente um sinal de entrada, concentrando a variância (energia) do sinal em poucos coeficientes (Britanak, Yip e Rao, 2007).

4.2.1 Correlação

Considere um sinal de entrada representado pelo vetor aleatório $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^T \in \mathbb{R}^N$. Em muitos sinais de interesse prático, como imagens naturais, as componentes x_i e x_j não são estatisticamente independentes, i.e., conhecer o valor de uma componente fornece informação sobre as demais. Essa dependência é denominada correlação e representa uma forma de redundância no sinal (Ray e Driver, 1970; Jain, 1989).

A correlação entre duas componentes x_i e x_j pode ser quantificada pelo valor esperado do produto das variáveis aleatórias centralizadas:

$$\sigma_{i,j} = \mathbb{E}\{(x_i - \mu_i)(x_j - \mu_j)\}, \quad (4.1)$$

em que $\mathbb{E}\{\cdot\}$ denota o operador valor esperado e $\mu_i = \mathbb{E}\{x_i\}$ é a média da i -ésima componente. Quando $\sigma_{i,j} = 0$ para $i \neq j$, as componentes são ditas descorrelacionadas, indicando ausência de dependência linear entre elas.

Para caracterizar completamente a estrutura de correlação de um sinal com N componentes, é necessário calcular $\sigma_{i,j}$ para todos os pares (i, j) . Esses valores podem ser organizados em uma estrutura matricial denominada matriz de covariância, definida por (Kay, 2012):

$$\mathbf{R}_x = \mathbb{E} \left\{ (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top \right\}, \quad (4.2)$$

em que $\boldsymbol{\mu} = \mathbb{E}\{\mathbf{x}\} = [\mu_0, \mu_1, \dots, \mu_{N-1}]^\top$ é o vetor de médias. Definindo a covariância entre os elementos i e j como $\sigma_{i,j} = \mathbb{E}[(x_i - \mu_i)(x_j - \mu_j)]$, a matriz $\mathbf{R}_x$ pode ser expressa por:

$$\mathbf{R}_x = \begin{bmatrix} \sigma_{0,0} & \sigma_{0,1} & \cdots & \sigma_{0,N-1} \\ \sigma_{1,0} & \sigma_{1,1} & \cdots & \sigma_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N-1,0} & \sigma_{N-1,1} & \cdots & \sigma_{N-1,N-1} \end{bmatrix}. \quad (4.3)$$

Os elementos da diagonal principal, $[\mathbf{R}_x]_{ii} = \mathbb{E}\{(x_i - \mu_i)^2\}$, representam a variância de cada componente. Os elementos fora da diagonal, $[\mathbf{R}_x]_{ij}$ com $i \neq j$, quantificam a covariância entre componentes distintas. Um sinal é completamente descorrelacionado se, e somente se, sua matriz de covariância é diagonal. Caso o sinal $\mathbf{x}$ não possua média nula, deve-se realizar a centralização dos dados antes de aplicar a transformação, subtraindo o vetor de médias:

$$\tilde{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}. \quad (4.4)$$

Dessa forma, o sinal centralizado $\tilde{\mathbf{x}}$ satisfaz $\mathbb{E}\{\tilde{\mathbf{x}}\} = \mathbf{0}$. Por conveniência de notação, assume-se doravante que essa centralização já foi realizada, de modo que $\boldsymbol{\mu} = \mathbf{0}$ e a matriz de covariância simplifica-se para:

$$\mathbf{R}_x = \mathbb{E} \left\{ \mathbf{x} \mathbf{x}^\top \right\}. \quad (4.5)$$

O objetivo fundamental da KLT é encontrar uma transformação linear que produza um sinal cuja matriz de covariância seja diagonal, eliminando assim toda a redundância associada à correlação entre componentes.

4.2.2 Formulação do Problema de Descorrelação

Busca-se uma transformação linear ortogonal $\mathbf{W}$ tal que o sinal transformado $\mathbf{y} = \mathbf{W}\mathbf{x}$ possua matriz de covariância diagonal:

$$\mathbf{R}_y = \mathbb{E} \left\{ \mathbf{y} \mathbf{y}^\top \right\} = \text{diag}(\lambda_0, \lambda_1, \dots, \lambda_{N-1}), \quad (4.6)$$

em que $\lambda_k = \mathbb{E}\{y_k^2\}$ representa a variância da k -ésima componente do sinal transformado.

A relação entre as matrizes de covariância dos sinais original e transformado é estabelecida por:

$$\mathbf{R}_y = \mathbb{E} \left\{ (\mathbf{W}\mathbf{x})(\mathbf{W}\mathbf{x})^\top \right\} = \mathbf{W} \mathbb{E} \left\{ \mathbf{x}\mathbf{x}^\top \right\} \mathbf{W}^\top = \mathbf{W}\mathbf{R}_x\mathbf{W}^\top. \quad (4.7)$$

Para que $\mathbf{R}_y$ seja diagonal, a matriz $\mathbf{W}$ deve ser construída de modo que suas linhas sejam os autovetores normalizados da matriz de covariância $\mathbf{R}_x$ (Strang, 1980). Especificamente, se $\mathbf{w}_k$ é a k -ésima linha de $\mathbf{W}$, então:

$$\mathbf{R}_x \mathbf{w}_k^\top = \lambda_k \mathbf{w}_k^\top, \quad k = 0, 1, \dots, N-1, \quad (4.8)$$

em que λ_k é o autovalor associado ao autovetor $\mathbf{w}_k$. Quando os autovalores são ordenados em ordem decrescente, a matriz $\mathbf{W}$ resultante é denominada matriz da KLT (Britanak, Yip e Rao, 2007).

4.2.3 Propriedades Fundamentais

A KLT apresenta as seguintes propriedades fundamentais (Rao e Yip, 1990):

1. **Descorrelação completa:** A matriz de covariância do sinal transformado é diagonal, eliminando a correlação entre componentes distintas.
2. **Compactação de energia:** Dentre todas as transformadas ortogonais, a KLT é aquela que concentra a maior quantidade de energia nos primeiros coeficientes, minimizando o erro quadrático médio (MSE) na reconstrução com número reduzido de coeficientes.
3. **Preservação de energia:** Por ser ortogonal, a KLT preserva a norma do sinal, isto é, $\|\mathbf{y}\|^2 = \|\mathbf{x}\|^2$.

4.2.4 KLT para Sinais Markovianos de Primeira Ordem

Embora a KLT não possua uma expressão fechada para sinais arbitrários, é possível derivar expressões analíticas quando se considera uma classe específica de sinais. Imagens naturais, em particular, podem ser modeladas com boa aproximação como processos markovianos estacionários de primeira ordem (Jain, 1989). Para um sinal modelado como processo de Markov de primeira ordem, a matriz de covariância assume a forma de uma matriz de Toeplitz simétrica (Britanak, Yip e Rao, 2007):

$$[\mathbf{R}_x]_{m,n} = \rho^{|m-n|}, \quad m, n = 0, 1, \dots, N-1, \quad (4.9)$$

em que ρ é o coeficiente de correlação entre amostras adjacentes, com $0 < \rho < 1$. Para imagens naturais, valores típicos situam-se na faixa $0.95 \leq \rho \leq 0.99$ (Britanak, Yip e Rao, 2007). A

matriz de covariância expressa em (4.9) possui a seguinte estrutura explícita:

$$\mathbf{R}_x = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \rho^2 & \rho & 1 & \dots & \rho^{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \dots & 1 \end{bmatrix}. \quad (4.10)$$

Para esta classe de sinais, a m -ésima componente do k -ésimo autovetor que satisfaz o problema de autovalores em (4.8) é dada por (Ray e Driver, 1970):

$$w_{k,m} = \sqrt{\frac{2}{N + \lambda_k}} \sin \left[\mu_k \left(m + 1 - \frac{N+1}{2} \right) + \frac{(k+1)\pi}{2} \right], \quad (4.11)$$

em que os autovalores correspondentes são:

$$\lambda_k = \frac{1 - \rho^2}{1 - 2\rho \cos(\mu_k) + \rho^2}, \quad (4.12)$$

e os parâmetros μ_k , para $k = 0, 1, \dots, N-1$, são as raízes reais positivas da equação transcendental:

$$\tan(N\mu) = -\frac{(1 - \rho^2) \sin(\mu)}{(1 + \rho^2) \cos(\mu) - 2\rho}. \quad (4.13)$$

4.3 DCT-II

4.3.1 Definição

A Transformada Discreta do Cosseno tipo II (DCT-II) de um sinal discreto $\mathbf{x} = [x_0, x_1, \dots, x_{N-1}]^\top$ é definida como a transformação linear que produz o vetor de coeficientes $\mathbf{X} = [X_0, X_1, \dots, X_{N-1}]^\top$, em que cada coeficiente é dado por (Rao e Yip, 1990):

$$X_k = \alpha_k \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi k(2n+1)}{2N} \right], \quad k = 0, 1, \dots, N-1, \quad (4.14)$$

com α_k definido em (4.24). A transformada inversa (IDCT-II), que recupera o sinal original a partir de seus coeficientes, é expressa por:

$$x_n = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha_k X_k \cos \left[\frac{\pi k(2n+1)}{2N} \right], \quad n = 0, 1, \dots, N-1. \quad (4.15)$$

O coeficiente X_0 , correspondente a $k = 0$, possui interpretação física especial. Substituindo $k = 0$ em (4.14):

$$X_0 = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n, \quad (4.16)$$

que é proporcional à média aritmética das amostras do sinal. Por essa razão, X_0 é denominado coeficiente DC, enquanto os demais coeficientes X_k para $k \neq 0$ são denominados coeficientes AC¹ (Gonzalez e Woods, 2018).

4.3.2 DCT-II como Aproximação da KLT

A relação entre a DCT e a KLT pode ser estabelecida analisando o comportamento da KLT para sinais markovianos de primeira ordem quando o coeficiente de correlação tende à unidade, isto é, quando $\rho \rightarrow 1$ (Clarke, 1981). Aplicando o limite $\rho \rightarrow 1$ à equação transcendental (4.13), obtém-se:

$$\lim_{\rho \rightarrow 1} \tan(N\mu) = \lim_{\rho \rightarrow 1} \left[-\frac{(1 - \rho^2) \sin(\mu)}{(1 + \rho^2) \cos(\mu) - 2\rho} \right] = 0. \quad (4.17)$$

A condição $\tan(N\mu) = 0$ implica que $N\mu$ deve ser um múltiplo inteiro de π . Consequentemente, as N raízes reais não-negativas são dadas por:

$$\mu_k = \frac{k\pi}{N}, \quad k = 0, 1, \dots, N-1. \quad (4.18)$$

Substituindo (4.18) em (4.12) e tomando o limite $\rho \rightarrow 1$, verifica-se que $\lambda_k = 0$ para $k \neq 0$. Para determinar λ_0 , utiliza-se a propriedade de que o traço de uma matriz é igual à soma de seus autovalores (Strang, 1980). O traço da matriz de covariância é:

$$\text{tr}(\mathbf{R}_x) = \sum_{n=0}^{N-1} [\mathbf{R}_x]_{n,n} = \sum_{n=0}^{N-1} 1 = N. \quad (4.19)$$

Portanto:

$$\sum_{k=0}^{N-1} \lambda_k = \lambda_0 + \sum_{k=1}^{N-1} 0 = \lambda_0 = N. \quad (4.20)$$

Substituindo os valores de μ_k e λ_k na expressão dos autovetores (4.11), obtém-se para $k = 0$:

$$w_{0,m} = \sqrt{\frac{2}{N+N}} \sin \left[\frac{\pi}{2} \right] = \sqrt{\frac{1}{N}}, \quad (4.21)$$

e para $k \neq 0$:

$$w_{k,m} = \sqrt{\frac{2}{N}} \sin \left[\frac{k\pi}{N} \left(m + \frac{1-N}{2} \right) + \frac{(k+1)\pi}{2} \right] = \sqrt{\frac{2}{N}} \cos \left[\frac{k\pi(2m+1)}{2N} \right]. \quad (4.22)$$

Unificando as equações (4.21) e (4.22), os elementos da matriz de transformação podem ser expressos como:

$$w_{k,m} = \alpha_k \sqrt{\frac{2}{N}} \cos \left[\frac{k\pi(2m+1)}{2N} \right], \quad k, m = 0, 1, \dots, N-1, \quad (4.23)$$

¹ A nomenclatura DC/AC é uma convenção da engenharia elétrica, em que *direct current* (corrente contínua) designa o componente constante de um sinal e *alternating current* (corrente alternada) designa seus componentes oscilatórios.

em que:

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{se } k = 0, \\ 1, & \text{se } k \neq 0. \end{cases} \quad (4.24)$$

A expressão (4.23) é a definição da DCT tipo II (DCT-II) (Ahmed, Natarajan e Rao, 1974). Este resultado fundamental demonstra que a DCT-II é o caso assintótico da KLT para sinais markovianos de primeira ordem com coeficiente de correlação unitário. Consequentemente, a DCT-II herda as propriedades de descorrelação e compactação de energia da KLT para sinais altamente correlacionados, como imagens naturais (Britanak, Yip e Rao, 2007).

4.3.3 Representação Matricial

A DCT-II pode ser expressa através de uma multiplicação matricial. Definindo a matriz de transformação $\mathbf{C}_N$ de dimensão $N \times N$, cujos elementos são dados por:

$$[\mathbf{C}_N]_{k,n} = \alpha_k \sqrt{\frac{2}{N}} \cos \left[\frac{\pi k(2n+1)}{2N} \right], \quad k, n = 0, 1, \dots, N-1, \quad (4.25)$$

a DCT-II e sua inversa podem ser escritas como:

$$\mathbf{X} = \mathbf{C}_N \cdot \mathbf{x}, \quad (4.26)$$

$$\mathbf{x} = \mathbf{C}_N^\top \cdot \mathbf{X}. \quad (4.27)$$

A matriz $\mathbf{C}_N$ é ortonormal, satisfazendo:

$$\mathbf{C}_N \cdot \mathbf{C}_N^\top = \mathbf{C}_N^\top \cdot \mathbf{C}_N = \mathbf{I}_N, \quad (4.28)$$

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

4.4 DCT-II BIDIMENSIONAL

A DCT, conforme definida anteriormente, opera sobre vetores unidimensionais, sendo adequada para o processamento de sinais como séries temporais e sinais de áudio. Contudo, aplicações bidimensionais — em particular a compressão de imagens — requerem, necessariamente, a extensão da DCT ao domínio bidimensional.

4.4.1 Caso Unidimensional

No caso unidimensional, seja $\mathbf{x} \in \mathbb{R}^N$ um vetor de entrada e $\mathbf{C}_N$ a matriz da DCT de tamanho $N \times N$. O vetor transformado $\mathbf{X} \in \mathbb{R}^N$ é obtido pela Equação (4.26).

4.4.2 Caso Bidimensional

Quando os dados são bidimensionais — como em imagens naturais, representadas por matrizes de intensidade de pixels — a decorrelação deve ser aplicada em ambas as dimensões. Seja $\mathbf{A} \in \mathbb{R}^{N \times N}$ uma matriz de entrada, como um bloco de pixels de uma imagem. A DCT bidimensional (DCT-2D) é obtida pela aplicação sequencial da transformada unidimensional nas linhas e nas colunas de $\mathbf{A}$ (Rao e Yip, 1990; Britanak, Yip e Rao, 2007):

$$\mathbf{B} = \mathbf{C}_N \cdot \mathbf{A} \cdot \mathbf{C}_N^\top, \quad (4.29)$$

em que $\mathbf{A}$ é a matriz de entrada (domínio espacial), $\mathbf{B}$ é a matriz transformada (domínio da frequência) e $\mathbf{C}_N$ é a DCT de tamanho N . A pré-multiplicação $\mathbf{C}_N \cdot \mathbf{A}$ aplica a transformada às colunas de $\mathbf{A}$, enquanto a pós-multiplicação por $\mathbf{C}_N^\top$ aplica a transformada às linhas do resultado intermediário. Essa propriedade, denominada separabilidade (Blahut, 2010), é fundamental para a viabilidade computacional da DCT-2D: em vez de operar diretamente sobre a matriz $N \times N$, o que implicaria um custo de ordem $O(N^4)$, a decomposição em duas etapas unidimensionais sucessivas reduz a complexidade para $O(N^3)$ (Gonzalez e Woods, 2018). Cabe destacar, contudo, que algoritmos rápidos reduzem a complexidade para $O(N^2 \log N)$. A operação inversa, que recupera a matriz original a partir dos coeficientes transformados, é dada por:

$$\mathbf{A} = \mathbf{C}_N^\top \cdot \mathbf{B} \cdot \mathbf{C}_N. \quad (4.30)$$

4.4.3 Aplicação em Compressão de Imagens

No contexto de compressão de imagens, como no padrão JPEG (Wallace, 1992; Gonzalez e Woods, 2018), a DCT-2D é aplicada a blocos de $N \times N$ pixels extraídos da imagem original. A transformada concentra a maior parte da energia do sinal em um número reduzido de coeficientes, localizados na região de baixas frequências da matriz $\mathbf{B}$. Os coeficientes de alta frequência, que tipicamente possuem magnitude pequena e contribuem pouco para a percepção visual, podem ser descartados ou quantizados com menor precisão, possibilitando a compressão dos dados.

Quando se utiliza uma aproximação $\hat{\mathbf{C}}_N$ da DCT, a operação bidimensional torna-se:

$$\hat{\mathbf{B}} = \hat{\mathbf{C}}_N \cdot \mathbf{A} \cdot \hat{\mathbf{C}}_N^\top. \quad (4.31)$$

Nesse caso, a qualidade da compressão depende diretamente do desempenho da aproximação $\hat{\mathbf{C}}_N$ em relação à DCT exata $\mathbf{C}_N$.

5 APROXIMAÇÕES DE BAIXA COMPLEXIDADE

5.1 INTRODUÇÃO

A DCT é uma ferramenta amplamente empregada em tarefas de compressão de imagens, decorrelação de dados na área de processamento de sinais (Ahmed, Natarajan e Rao, 1974; Britanak, Yip e Rao, 2007). Entretanto, sua implementação direta, i.e., pela definição, envolve operações aritméticas computacionalmente custosas, em especial multiplicações por coeficientes não triviais (Rao e Yip, 1990), dificultando sua utilização em sistemas com restrições de processamento e consumo energético.

Nesse contexto, as transformadas aproximadas de baixa complexidade devem preservar as propriedades estatísticas da DCT exata e, simultaneamente, reduzir o custo computacional da implementação (Bouguezel, Ahmad e Swamy, 2008; Cintra e Bayer, 2011a). Em geral, aproximações substituem multiplicações por operações mais simples — triviais —, como somas, deslocamentos de bits e mudanças de sinal (Blahut, 2010).

Ao longo deste trabalho, $\mathbf{T}_N$ representa a matriz de baixa complexidade, enquanto $\hat{\mathbf{C}}_N$ é a aproximação de $\mathbf{T}_N$ efetivamente utilizada para comparação com a DCT exata e com a literatura.

5.2 COMPLEXIDADE COMPUTACIONAL

A complexidade computacional é um critério para avaliar a viabilidade de implementação de transformadas discretas em sistemas com recursos limitados (Blahut, 2010). Em aplicações embarcadas, o desempenho de uma transformada não depende apenas de figuras de mérito de erro, introduzidas na Seção 5.5, mas também da quantidade de operações como multiplicação, adição e deslocamento de bits (Parhi, 1999).

5.2.1 Complexidade Aritmética

A complexidade aritmética é uma medida que quantifica o número de operações elementares necessárias para a execução de um algoritmo em função do tamanho da entrada (Levitin, 2012). No contexto de transformadas discretas, a complexidade aritmética leva em conta a complexidade multiplicativa, aditiva e demais operações como deslocamento de bit. Desse modo, tal complexidade pode ser expressa pela seguinte expressão:

$$\mathcal{C} = f(\mathcal{M}, \mathcal{A}, \mathcal{S}), \quad (5.1)$$

em que $\mathcal{M}$, $\mathcal{A}$ e $\mathcal{S}$ denotam, respectivamente, a complexidade multiplicativa, complexidade aditiva e deslocamento de bit (Levitin, 2012; Blahut, 2010).

Em processadores digitais, multiplicações em ponto flutuante — não triviais — exigem mais recursos do que somas ou deslocamento de bit. A complexidade multiplicativa é determinada pelo número de operações de multiplicação não-triviais.

Multiplicações triviais dispensam aritmética de ponto flutuante, e.g., multiplicações por elementos do conjunto $\{0, \pm 1, \pm 2, \pm 4\}$, podem ser implementadas com adições, inversão de sinal e deslocamentos de bits. A viabilidade de implementação é determinada pelo conjunto ao qual pertencem os coeficientes da transformada. Tipicamente, são empregados conjuntos de potências de dois (Portella, Bayer e Cintra, 2025), definidos como:

$$\mathcal{P}_1 = \{0, \pm 1\}, \quad (5.2)$$

$$\mathcal{P}_2 = \{0, \pm 1, \pm 2\}, \quad (5.3)$$

$$\mathcal{P}_4 = \{0, \pm 1, \pm 2, \pm 4\}, \quad (5.4)$$

$$\mathcal{P}_{2^n} = \{0, \pm 1, \pm 2, \dots, \pm 2^n\} \quad \forall n \in \mathbb{N}. \quad (5.5)$$

Coeficientes pertencentes a conjuntos como $\mathcal{P}_1$, $\mathcal{P}_2$ e $\mathcal{P}_4$ são atrativos em projetos de transformadas aproximadas. O cômputo direto da DCT-II de comprimento N , por multiplicação matricial, requer N^2 multiplicações e $N(N-1)$ adições, sendo a cota superior de complexidade aritmética para essa transformada (Heideman, 1988). Para o caso $N = 8$, isso resulta em 64 multiplicações e 56 adições.

5.3 APROXIMAÇÃO DE BAIXA COMPLEXIDADE

As aproximações da DCT visam reduzir o custo computacional e preservar propriedades desejáveis da transformada exata. A matriz de baixa complexidade $\mathbf{T}_N$ dá origem à matriz aproximada $\hat{\mathbf{C}}_N$, que é a forma efetivamente utilizada na comparação com a DCT exata.

Particularmente, uma propriedade desejável em aproximações da DCT é a ortogonalidade (Britanak, Yip e Rao, 2007; Rao e Yip, 1990). Considere-se as seguintes definições para ortogonalidade e ortonormalidade:

Teorema 2 (Ortogonalidade). *Uma matriz quadrada $\mathbf{T}$ é dita ortogonal se $\mathbf{T} \cdot \mathbf{T}^\top = \mathbf{D}$ em que $\mathbf{D}$ é uma matriz diagonal.*

Teorema 3 (Ortonormalidade). *Uma matriz quadrada $\mathbf{T}$ é dita ortonormal se $\mathbf{T} \cdot \mathbf{T}^\top = \mathbf{T}^\top \cdot \mathbf{T} = \mathbf{I}$ em que $\mathbf{I}$ é a matriz identidade.*

Quando $\mathbf{T}$ é ortogonal, uma aproximação $\mathbf{C}$ é obtida normalizando $\mathbf{T}$ com uma matriz diagonal de ajuste, denotada por $\mathbf{S}$. O método de ortonormalização é baseado na decomposição polar (Higham, 1986). A partir da decomposição polar, a matriz diagonal de ajuste pode ser obtida por:

$$\mathbf{S} = \sqrt{[\mathbf{T} \cdot \mathbf{T}^\top]^{-1}}, \quad (5.6)$$

em que $\sqrt{\cdot}$ denota a raiz quadrada matricial (Higham, 1987). Então, a aproximação para a DCT é dada por:

$$\hat{\mathbf{C}} = \mathbf{S} \cdot \mathbf{T}. \quad (5.7)$$

Para o caso de $\mathbf{T}$ não-ortogonal, a matriz $\mathbf{S}$ não é diagonal. Desta forma, uma matriz aproximada $\hat{\mathbf{C}}$ é obtida por um método de quase-ortogonalização, adaptando a Equação (5.6) para abranger matrizes não-ortogonais. Então, o método é dado pela seguinte expressão:

$$\tilde{\mathbf{S}} = \sqrt{[\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)]^{-1}}, \quad (5.8)$$

em que $\text{diag}(\cdot)$ constrói uma matriz diagonal a partir dos elementos da diagonal principal do argumento. Se $\mathbf{T}$ é não-ortogonal, uma aproximação não-ortonormal $\tilde{\mathbf{C}}$ é obtida por:

$$\tilde{\mathbf{C}} = \tilde{\mathbf{S}} \cdot \mathbf{T}. \quad (5.9)$$

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

$$\hat{\mathbf{X}} = \tilde{\mathbf{C}} \cdot \mathbf{x} = \tilde{\mathbf{S}} \cdot \mathbf{T} \cdot \mathbf{x}, \quad (5.10)$$

e sua transformada inversa é dada por:

$$\mathbf{x} = \tilde{\mathbf{C}}^{-1} \cdot \hat{\mathbf{X}} = \mathbf{T}^{-1} \cdot \tilde{\mathbf{S}}^{-1} \cdot \hat{\mathbf{X}}. \quad (5.11)$$

Se há ortogonalidade, as transformadas direta e inversa são dadas, respectivamente, por

$$\hat{\mathbf{X}} = \hat{\mathbf{C}} \cdot \mathbf{x} = \mathbf{S} \cdot \mathbf{T} \cdot \mathbf{x}, \quad (5.12)$$

$$\mathbf{x} = \hat{\mathbf{C}}^\top \cdot \hat{\mathbf{X}} = \mathbf{T}^\top \cdot \mathbf{S} \cdot \hat{\mathbf{X}}. \quad (5.13)$$

5.4 ALGORITMOS RÁPIDOS

Algoritmos rápidos reduzem significativamente o número de operações aritméticas necessárias decompondo uma transformada de comprimento N em duas transformadas de comprimento $N/2$ (Blahut, 2010). Tal decomposição é aplicada recursivamente até que se atinja transformadas de comprimento unitário. Uma estrutura comumente empregada na literatura é a *butterfly*, que combina dois sinais de entrada para produzir dois sinais de saída por meio de operações de soma, diferença e escalonamento ((Blahut, 2010), p. 74).

A *butterfly* pode ser representada como um grafo de fluxo de sinal com duas entradas x_0 e x_1 e duas saídas X_0 e X_1 , conectadas por quatro ramos com ganhos a , b , c e d , conforme ilustrado na Figura 2.

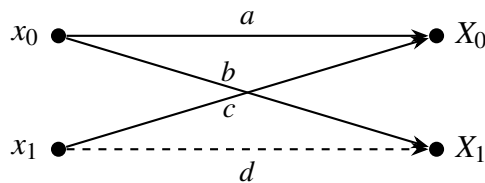


Figura 2 – Representação de uma *Butterfly*

As saídas são dadas pelas combinações lineares:

$$X_0 = ax_0 + cx_1, \quad (5.14)$$

$$X_1 = bx_0 + dx_1. \quad (5.15)$$

A *butterfly* de dois ramos funciona segundo a forma simétrica

$$X_0 = \frac{x_0 + x_1}{2}, \quad (5.16)$$

$$X_1 = \frac{x_0 - x_1}{2}, \quad (5.17)$$

correspondendo ao caso particular $a = c = \frac{1}{2}$, $b = \frac{1}{2}$ e $d = -\frac{1}{2}$. Essa estrutura admite implementação de baixo custo aritmético. A linha tracejada na Figura 2 denota a negação de sinal na estrutura (Blahut, 2010).

5.4.1 Fatoração de Matrizes

De modo simplificado, a representação de uma transformada como produto de matrizes esparsas — cada uma sendo um estágio de *butterfly* — é o princípio dos algoritmos rápidos. Para fins de ilustração, considere-se a transformada de Walsh-Hadamard (WHT) de ordem $N = 8$, cuja expressão matricial é:

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}. \quad (5.18)$$

O cômputo direto por Equação (5.18) requer 64 adições; um algoritmo rápido baseado em *butterfly* reduz esse número para 24 adições, como mostra a Figura 3.

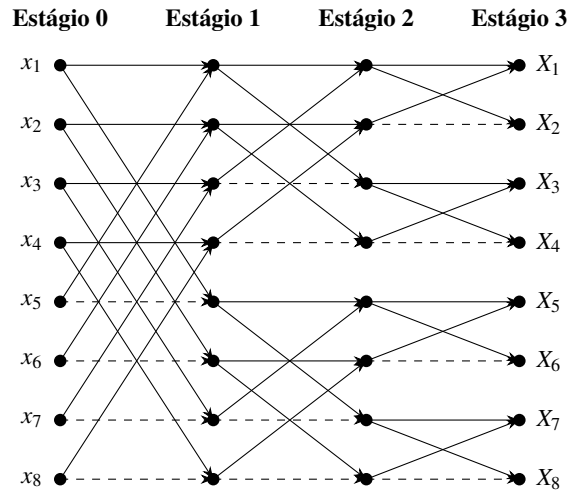


Figura 3 – Algoritmo rápido para a transformada de Walsh-Hadamard de ordem 8. Setas sólidas indicam soma; setas tracejadas, subtração.

O algoritmo tem 3 estágios de *butterfly*. Essa estrutura de fatoração é empregada nas matrizes aproximadas propostas neste trabalho, no Capítulo 7.

5.5 FIGURAS DE MÉRITO

A avaliação quantitativa das aproximações da DCT é realizada por meio de figuras de mérito que medem diferentes aspectos do desempenho das matrizes aproximadas em relação à transformada exata. De forma geral, essas métricas podem ser agrupadas em (i) medidas de erro, que avaliam o quanto a matriz aproximada se distancia da DCT exata e (ii) medidas de codificação, que avaliam a capacidade da transformada em descorrelacionar sinais e concentrar energia em poucos coeficientes.

5.5.1 Modelagem do Sinal de Entrada

Para as métricas ligadas à codificação, o sinal de entrada é modelado como um processo de Markov estacionário de primeira ordem, com média nula, variância unitária e coeficiente de correlação ρ (Britanak, Yip e Rao, 2007). Esse modelo é amplamente empregado na literatura por representar adequadamente a correlação entre pixels adjacentes, característica de imagens naturais. A matriz de covariância $\mathbf{R}_x$ desse processo possui elementos dados por

$$[\mathbf{R}_x]_{i,j} = \rho^{|i-j|}, \quad i, j = 0, 1, \dots, N-1. \quad (5.19)$$

Neste trabalho, adota-se $\rho = 0.95$, valor comumente utilizado em estudos comparativos de transformadas aproximadas (Cintra e Bayer, 2011b). Essa escolha permite confrontar os resultados obtidos neste trabalho com aqueles reportados na literatura.

5.5.2 Medidas de Similaridade

5.5.2.1 Erro de Energia Total

O erro de energia total mensura a distância espectral entre a aproximação e a DCT exata (Cintra e Bayer, 2011a). Cada linha da matriz de transformação pode ser interpretada como a resposta ao impulso. Assim, o erro de energia é calculado a partir da diferença entre as respectivas funções de transferência.

Sejam $H_m(\omega)$ e $\hat{H}_m(\omega)$ as transformadas de Fourier de tempo discreto da m -ésima linha de $\mathbf{C}$ e $\hat{\mathbf{C}}$, respectivamente:

$$H_m(\omega) = \sum_{n=0}^{N-1} c_{m,n} e^{-jn\omega}, \quad \hat{H}_m(\omega) = \sum_{n=0}^{N-1} \hat{c}_{m,n} e^{-jn\omega}. \quad (5.20)$$

O erro de energia da m -ésima linha é definido por

$$\varepsilon_m = \int_0^\pi |H_m(\omega) - \hat{H}_m(\omega)|^2 d\omega. \quad (5.21)$$

Pelo teorema de Parseval, apresentado na Seção 3.3, essa expressão pode ser calculada no domínio do tempo como:

$$\varepsilon_m = \pi \sum_{n=0}^{N-1} |c_{m,n} - \hat{c}_{m,n}|^2. \quad (5.22)$$

O erro de energia total é então dado por

$$\varepsilon(\hat{\mathbf{C}}) = \sum_{m=0}^{N-1} \varepsilon_m = \pi \|\mathbf{C} - \hat{\mathbf{C}}\|_F^2, \quad (5.23)$$

em que $\|\cdot\|_F$ representa a norma de Frobenius (Seber, 2008). Quanto menor esse valor, maior a semelhança entre a transformada aproximada e a DCT exata.

5.5.2.2 Erro Médio Quadrático

O erro médio quadrático ou MSE (*Mean Squared Error*), mede a diferença entre a saída da transformada aproximada e a saída da DCT exata quando ambas são aplicadas ao mesmo sinal de entrada modelado por $\mathbf{R}_x$ (Britanak, Yip e Rao, 2007; Wang e Bovik, 2009):

$$\text{MSE}(\hat{\mathbf{C}}) = \frac{1}{N} \text{tr} \left[(\mathbf{C} - \hat{\mathbf{C}}) \mathbf{R}_x (\mathbf{C} - \hat{\mathbf{C}})^\top \right], \quad (5.24)$$

em que o operador $\text{tr}(\cdot)$ denota a soma dos elementos da diagonal principal. Valores menores de MSE indicam menor distorção na representação do sinal.

5.5.3 Medidas de Codificação

5.5.3.1 Ganho de Codificação Unificado

O ganho de codificação (*Coding Gain*) quantifica a eficiência de uma transformada em concentrar energia em poucos coeficientes, em comparação com a codificação direta no domínio do

tempo (Goyal, 2001). Para transformadas não-ortogonais, utiliza-se a versão unificada proposta por Katto e Yasuda (1991).

Seja $\mathbf{h}_m^\top$ a m -ésima linha de $\hat{\mathbf{C}}$ e $\mathbf{g}_m^\top$ a m -ésima linha de $\hat{\mathbf{C}}^{-1}$. O ganho de codificação unificado é definido por:

$$C_g^*(\hat{\mathbf{C}}) = 10 \log_{10} \left(\prod_{m=0}^{N-1} \frac{1}{\sqrt[N]{A_m \cdot B_m}} \right), \quad (5.25)$$

em que

$$A_m = \text{su} \left[\left(\mathbf{h}_m \mathbf{h}_m^\top \right) \odot \mathbf{R}_x \right], \quad (5.26)$$

$$B_m = \|\mathbf{g}_m\|_2^2. \quad (5.27)$$

Na Equação (5.26), o operador $\text{su}(\cdot)$ calcula a soma de todos os elementos da matriz argumento, e $\odot$ denota o produto de Hadamard, i.e., produto elemento a elemento.

Valores maiores de C_g^* indicam melhor capacidade de compactação de energia. Para referência, a KLT e a DCT exata apresentam ganhos de 8.8462 dB e 8.8259 dB, respectivamente, quando $\rho = 0.95$ (Britanak, Yip e Rao, 2007).

5.5.3.2 Eficiência da Transformada

A eficiência da transformada (*Transform Efficiency*) mede a capacidade de descorrelação da aproximação, quantificando a parcela da energia total do sinal transformado que se concentra na diagonal principal da matriz de covariância de saída (Britanak, Yip e Rao, 2007).

Seja $\mathbf{R}_y = \hat{\mathbf{C}} \mathbf{R}_x \hat{\mathbf{C}}^\top$ a matriz de covariância do sinal transformado. A eficiência é definida por:

$$\eta(\hat{\mathbf{C}}) = \frac{\sum_{m=0}^{N-1} |r_{m,m}|}{\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} |r_{m,n}|} \times 100\%, \quad (5.28)$$

em que $r_{m,n} = [\mathbf{R}_y]_{m,n}$.

Uma transformada que descorrelaciona completamente o sinal produz uma matriz $\mathbf{R}_y$ diagonal, de modo que $\eta = 100\%$. A KLT, por definição, atinge esse valor para qualquer ρ .

6 METODOLOGIA

Este capítulo descreve a metodologia adotada para obtenção de aproximações de baixa complexidade da DCT. O método divide-se em duas etapas: (i) aproximação de matrizes da DFT por minimização de diferença angular com conjuntos de multiplicandos triviais e (ii) conversão dos resultados em aproximações da DCT, através de uma relação algébrica entre as transformadas, elucidada na Seção 6.3.

6.1 FLUXO METODOLÓGICO

A Figura 4 ilustra o encadeamento das etapas do trabalho. Inicialmente, são obtidas aproximações matriciais de baixa complexidade da DFT. Em seguida, tais aproximações são convertidas em matrizes aproximadas da DCT, por meio de teorema que leva uma DFT de comprimento $4N$ a uma DCT de dimensão N (Blahut, 2010).

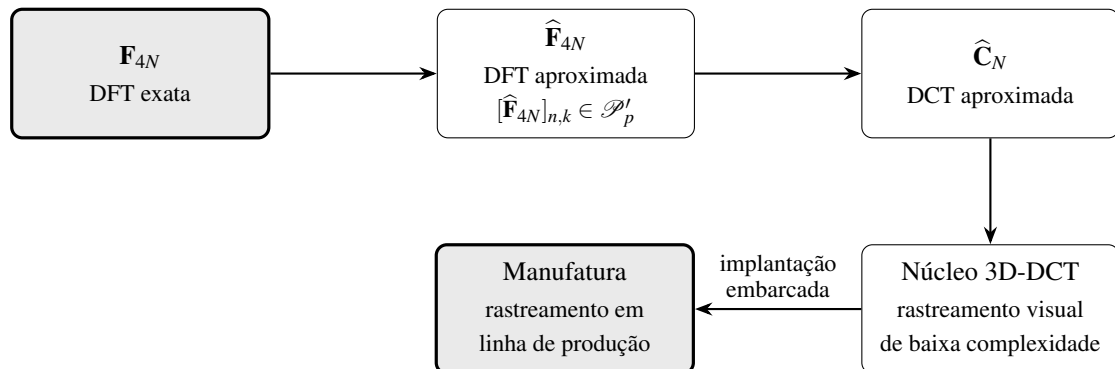
6.2 PROBLEMA DE OTIMIZAÇÃO

A primeira etapa consiste em obter $\hat{\mathbf{F}}_N$, substituindo cada elemento exato por um inteiro gaussiano. Esta substituição é formulada como um problema de otimização cujo objetivo é minimizar o erro angular total em relação à DFT exata.

6.2.1 Espaços de Busca

Os elementos das matrizes aproximadas, $[\hat{\mathbf{F}}_N]_{n,k}$, são selecionados de conjuntos de inteiros gaussianos, denominados espaços de busca. Cada posição (n,k) da matriz aproximada é uma variável de decisão, que assume valor em um dos espaços de busca definidos nesta seção.

Figura 4 – Fluxo metodológico adotado no trabalho



Fonte: O autor (2026).

6.2.1.1 Conjuntos de multiplicandos triviais

Seja $E = \{2^m : m \in \mathbb{Z}_{\geq 0}\}$ o conjunto das potências de dois com expoente não-negativo. Para cada $p \in E$, define-se:

$$S_p = \{0\} \cup \{\pm 2^i : i \in \{0, 1, \dots, \log_2 p\}\}, \quad (6.1)$$

de cardinalidade $|S_p| = 2\log_2 p + 3$. Então, para $p = 1, 2, 4$, têm-se os seguintes conjuntos:

$$S_1 = \{0, \pm 1\}, \quad (6.2)$$

$$S_2 = \{0, \pm 1, \pm 2\}, \quad (6.3)$$

$$S_4 = \{0, \pm 1, \pm 2, \pm 4\}. \quad (6.4)$$

6.2.1.2 Conjuntos de inteiros gaussianos

A partir de S_p com $p \in \{1, 2, 4\}$, formam-se os conjuntos de inteiros gaussianos $\mathcal{P}_p$:

$$\mathcal{P}_p = \{a + jb : a, b \in S_p, a \neq 0 \vee b \neq 0\}. \quad (6.5)$$

Para $p \geq 2$, os conjuntos $\mathcal{P}_p$ contêm elementos com o mesmo argumento, diferindo apenas em magnitude. Dois inteiros gaussianos não-nulos z_1 e z_2 possuem o mesmo argumento se, e somente se, um é múltiplo positivo do outro, isto é, $z_2 = \alpha z_1$ para $\alpha \in \mathbb{N}$.

Por exemplo, em $\mathcal{P}_2$, os elementos $z_1 = 1 + j$ e $z_2 = 2 + 2j$ são tais que $\text{Arg}(z_2) = \text{Arg}(z_1) = \pi/4$. A presença de múltiplos no espaço de busca introduz multiplicidade de seleção: para um dado argumento alvo $\theta_{n,k}$, mais de um candidato produz a mesma diferença angular. Para contornar tal problema, adotou-se o critério de maximização de magnitude: dentre todos os elementos múltiplos, retém-se apenas o de maior módulo. O espaço de busca reduzido $\mathcal{P}'_p$ é, portanto:

$$\mathcal{P}'_p = \{z \in \mathcal{P}_p \mid \nexists w \in \mathcal{P}_p : w = \alpha z, \text{ com } \alpha \in \mathbb{N}, \alpha > 1\}. \quad (6.6)$$

Aplicando (6.6), obtêm-se os espaços reduzidos:

- $\mathcal{P}'_1 = \mathcal{P}_1$, com $|\mathcal{P}'_1| = 8$, pois $\mathcal{P}_1$ não contém elementos múltiplos.
- $\mathcal{P}'_2 = \mathcal{P}_2 \setminus \mathcal{P}_1$, pois os 8 elementos de $\mathcal{P}_1$ possuem correspondentes de maior magnitude em $\mathcal{P}_2$.
- $\mathcal{P}'_4 = \mathcal{P}_4 \setminus (\mathcal{P}'_1 \cup \mathcal{P}'_2)$.

Cabe destacar que os espaços reduzidos $\mathcal{P}'_p$ são utilizados apenas na etapa de aproximação da DFT. Após a conversão descrita na Seção 6.3, a aproximação resultante $\hat{\mathbf{C}}_N$ tem entradas reais, com valores no conjunto S_p . Desse modo, as aproximações $\hat{\mathbf{C}}_4^{(p)}$ e $\hat{\mathbf{C}}_8^{(p)}$ apresentadas no Capítulo 7 possuem elementos em S_1 , S_2 ou S_4 , conforme o espaço $\mathcal{P}'_p$ empregado na aproximação da DFT de origem.

6.2.1.3 Espaço das matrizes aproximadas

Definido o espaço de busca $\mathcal{P}'_p$, o espaço das matrizes aproximadas $N \times N$ com entradas em $\mathcal{P}'_p$ é dado por:

$$\mathcal{M}_N(\mathcal{P}'_p) = \{ \mathbf{M} \in \mathbb{C}^{N \times N} \mid [\mathbf{M}]_{n,k} \in \mathcal{P}'_p, n, k = 0, 1, \dots, N-1 \}. \quad (6.7)$$

Cada matriz em $\mathcal{M}_N(\mathcal{P}'_p)$ representa uma combinação das N^2 variáveis de decisão. É sobre este conjunto que o problema de otimização é definido.

6.2.2 Função Objetivo

O critério de aproximação adotado é a minimização do somatório da diferença angular entre as entradas de $\mathbf{F}_N$ e de uma matriz candidata $\mathbf{M} \in \mathcal{M}_N(\mathcal{P}'_p)$. O argumento de um número complexo $z = x + yj \in \mathbb{C}^*$ é calculado pela função de dois argumentos $\text{atan2}(y, x)$, que resolve as ambiguidades de quadrante da função arco-tangente convencional. Sua definição é:

$$\text{atan2}(y, x) = \begin{cases} \arctan(y/x), & \text{se } x > 0, \\ \arctan(y/x) + \pi, & \text{se } x < 0 \text{ e } y \geq 0, \\ \arctan(y/x) - \pi, & \text{se } x < 0 \text{ e } y < 0, \\ +\pi/2, & \text{se } x = 0 \text{ e } y > 0, \\ -\pi/2, & \text{se } x = 0 \text{ e } y < 0. \end{cases} \quad (6.8)$$

O caso $x = y = 0$, em que atan2 é indefinida, corresponde ao elemento nulo. A imagem de atan2 é o intervalo $(-\pi, \pi]$. Como o argumento é uma quantidade periódica de período 2π , a diferença angular é dada por:

$$\Delta(\theta_1, \theta_2) = \min(|\theta_1 - \theta_2|, 2\pi - |\theta_1 - \theta_2|), \quad (6.9)$$

que garante $\Delta(\theta_1, \theta_2) \in [0, \pi]$ para quaisquer $\theta_1, \theta_2 \in (-\pi, \pi]$.

A matriz aproximada $\hat{\mathbf{F}}_N$ é a solução do problema global de minimização do somatório das diferenças angulares elemento-a-elemento entre $\mathbf{F}_N$ e uma matriz candidata $\mathbf{M}$ sobre o espaço $\mathcal{M}_N(\mathcal{P}'_p)$:

$$\hat{\mathbf{F}}_N = \arg \min_{\mathbf{M} \in \mathcal{M}_N(\mathcal{P}'_p)} \sum_{n=0}^{N-1} \sum_{k=0}^{N-1} \Delta(\text{Arg}([\mathbf{F}_N]_{n,k}), \text{Arg}([\mathbf{M}]_{n,k})). \quad (6.10)$$

O problema (6.10) é combinatório sobre o espaço $\mathcal{M}_N(\mathcal{P}'_p)$, cuja cardinalidade cresce com N :

$$|\mathcal{M}_N(\mathcal{P}'_p)| = |\mathcal{P}'_p|^{N^2}. \quad (6.11)$$

Por exemplo, para $N = 8$ e $p = 1$, há $8^{64} \approx 6,3 \times 10^{57}$ candidatos. Uma busca exaustiva é, portanto, computacionalmente inviável. Entretanto, a função em (6.10) é separável nas variáveis de decisão: cada parcela do somatório depende somente do elemento $[\mathbf{M}]_{n,k}$ na posição (n, k) . Sob esta condição, o mínimo global da soma é alcançado pela minimização independente de cada

termo. Consequentemente, o problema global decompõe-se em N^2 subproblemas independentes:

$$[\hat{\mathbf{F}}_N]_{n,k} = \arg \min_{z \in \mathcal{P}'_p} \Delta(\text{Arg}([\mathbf{F}_N]_{n,k}), \text{Arg}(z)), \quad n, k = 0, 1, \dots, N-1. \quad (6.12)$$

Cada subproblema (6.12) é resolvido por busca exaustiva sobre os $|\mathcal{P}'_p|$ candidatos. O número total de avaliações da função objetivo é $|\mathcal{P}'_p| \cdot N^2$, em contraste com as $|\mathcal{P}'_p|^{N^2}$ avaliações exigidas pela busca exaustiva sobre $\mathcal{M}_N(\mathcal{P}'_p)$. Para $N = 8$ e $p = 1$, são apenas $8 \cdot 64 = 512$ avaliações.

O procedimento computacional que retorna $\hat{\mathbf{F}}$ é descrito no Algoritmo 1.

Algoritmo 1: Aproximação da DFT por minimização de diferença angular.

```

1 Função ApproxDFT( $N, \mathcal{P}'_p$ ):
   Entrada: Tamanho da transformada  $N$ ; conjunto  $\mathcal{P}'_p$ .
   Saída: Matriz aproximada  $\hat{\mathbf{F}}_N$  com elementos em  $\mathcal{P}'_p$ .

2   for  $k \leftarrow 0$  to  $N-1$  do
3     for  $n \leftarrow 0$  to  $N-1$  do
4        $\theta \leftarrow \text{Arg}([\mathbf{F}_N]_{n,k})$ 
5       for  $z_i \in \mathcal{P}'_p$  do
6          $\phi \leftarrow \text{Arg}(z_i)$ 
7          $d \leftarrow |\theta - \phi|$ 
8          $e_i \leftarrow \min(d, 2\pi - d)$ 
9        $i^* \leftarrow \arg \min_i e_i$ 
10       $[\hat{\mathbf{F}}_N]_{n,k} \leftarrow z_{i^*}$ 
11   return  $\hat{\mathbf{F}}_N$ 

```

Fonte: O autor (2026).

Este algoritmo foi aplicado para os tamanhos $N = 16$ e $N = 32$ utilizando os três espaços de busca, $\mathcal{P}'_1$, $\mathcal{P}'_2$ e $\mathcal{P}'_4$, resultando em 6 aproximações da DFT.

6.3 CONVERSÃO DFT-DCT

A segunda etapa da metodologia consiste na conversão dos resultados obtidos na etapa anterior em aproximações da DCT.

6.3.1 Fundamentação Teórica

A conexão entre a DFT e a DCT foi estabelecida por Blahut (Blahut, 2010), que demonstrou como os coeficientes da DCT podem ser obtidos a partir dos coeficientes da DFT de um vetor apropriadamente construído. O resultado central é enunciado a seguir.

Teorema 4 ((Blahut, 2010), Teorema 3.5.1). *A Transformada Discreta do Cosseno de comprimento N de um vetor $\mathbf{v} = [v_0, v_1, \dots, v_{N-1}]^\top$ é igual aos primeiros N componentes da Transformada Discreta de Fourier de comprimento $4N$ de um vetor $\mathbf{u}$, construído da seguinte forma:*

$$u_{2i+1} = \begin{cases} \frac{1}{2}v_i, & i = 0, \dots, N-1, \\ \frac{1}{2}v_{2N-i-1}, & i = N, \dots, 2N-1, \end{cases} \quad (6.13)$$

$$u_{2i} = 0, \quad i = 0, \dots, 2N-1.$$

em que os componentes de $\mathbf{u}$ são indexados por $i = 0, 1, \dots, 4N-1$.

Em termos conceituais, o vetor $\mathbf{u}$ é construído por operações. Primeiro, realiza-se a expansão simétrica: o vetor $\mathbf{v}$ é concatenado com sua versão temporalmente reversa, formando um vetor de comprimento $2N$. Em seguida, aplica-se a intercalação com zeros: zeros são inseridos entre cada par de elementos consecutivos, de modo que os valores não nulos ocupem apenas os índices ímpares. Por fim, efetua-se o escalonamento: o vetor resultante é dividido por 2 para normalização.

6.3.2 Demonstração do Teorema

Demonstração. Seja $\omega = e^{-j\frac{2\pi}{4N}} = e^{-j\frac{\pi}{2N}}$ a raiz primitiva de ordem $4N$ da unidade ($\omega^{4N} = e^{-j2\pi} = 1$). Os primeiros N componentes da DFT de comprimento $4N$ do vetor $\mathbf{u}$ são dados por:

$$V_k = \sum_{\ell=0}^{4N-1} \omega^{k\ell} u_\ell, \quad k = 0, 1, \dots, N-1. \quad (6.14)$$

Pela construção de $\mathbf{u}$ em (6.13), os únicos termos não nulos ocorrem nos índices $\ell = 2i+1$. Substituindo:

$$V_k = \frac{1}{2} \sum_{i=0}^{N-1} \omega^{(2i+1)k} v_i + \frac{1}{2} \sum_{i=N}^{2N-1} \omega^{(2i+1)k} v_{2N-i-1}. \quad (6.15)$$

No segundo somatório, define-se a variável i' como $i' = 2N - i - 1$, de modo que quando $i = N$ tem-se que $i' = N - 1$; e quando $i = 2N - 1$ tem-se que $i' = 0$. Além disso, $2i+1 = 2(2N - i' - 1) + 1 = 4N - 2i' - 1$. Portanto, quando i cresce de N a $2N - 1$, o índice i' decresce de $N - 1$ a 0. Como $\omega^{4N} = 1$, segue que:

$$\omega^{k(2i+1)} = \omega^{k(4N-2i'-1)} = \omega^{-(2i'+1)k}. \quad (6.16)$$

Portanto, a equação (6.15) torna-se:

$$V_k = \frac{1}{2} \sum_{i=0}^{N-1} \omega^{(2i+1)k} v_i + \frac{1}{2} \sum_{i'=0}^{N-1} \omega^{-(2i'+1)k} v_{i'}. \quad (6.17)$$

Renomeando i' de volta para i , combinando os dois somatórios e utilizando a identidade de Euler, apresentada em (3.14), com $\theta = \frac{\pi}{2N}k(2i+1)$, tem-se:

$$V_k = \frac{1}{2} \sum_{i=0}^{N-1} v_i \left(\omega^{(2i+1)k} + \omega^{-(2i+1)k} \right) = \sum_{i=0}^{N-1} v_i \cos \left(\frac{\pi(2i+1)k}{2N} \right). \quad (6.18)$$

A expressão (6.18) corresponde à DCT-II de comprimento N , sem os fatores de normalização α_k e $\sqrt{2/N}$, completando a demonstração. $\square$

6.3.3 Adaptação Matricial

O Teorema 4 estabelece uma relação entre vetores transformados. Para os propósitos deste trabalho, é necessário adaptar esse resultado para a conversão direta entre matrizes de transformação, independentemente de vetores de entrada.

Seja $\mathbf{F}_{4N}$ a matriz da DFT de dimensão $4N \times 4N$. A matriz da DCT de dimensão $N \times N$ pode ser obtida observando que cada coluna da DCT corresponde à transformada do vetor da base canônica associado à respectiva posição.

Para um vetor de entrada $\mathbf{v} = \boldsymbol{\delta}_j$ definido em termos do delta de Kronecker em (3.31), i.e., $(\delta_j)_n = \delta[n - j]$, a construção (6.13) produz um vetor $\mathbf{u}$ com valores não nulos nas posições:

$$\ell_1 = 2j + 1, \quad (6.19)$$

$$\ell_2 = 4N - (2j + 1) = 4N - 2j - 1. \quad (6.20)$$

Como $\mathbf{A}\boldsymbol{\delta}_j$ corresponde à j -ésima coluna de uma matriz $\mathbf{A}$, obtêm-se:

$$\mathbf{F}_{4N}\mathbf{u} = \frac{1}{2} \left[\mathbf{F}_{4N}\boldsymbol{\delta}_{\ell_1} + \mathbf{F}_{4N}\boldsymbol{\delta}_{\ell_2} \right] = \frac{1}{2} \left[\mathbf{F}_{4N}(:, \ell_1) + \mathbf{F}_{4N}(:, \ell_2) \right].$$

Tomando apenas as primeiras N componentes e a parte real, obtém-se a j -ésima coluna de $\mathbf{C}_N$:

$$\mathbf{C}_N(:, j) = \frac{1}{2} \Re \{ \mathbf{F}_{4N}(:, \ell_1) + \mathbf{F}_{4N}(:, \ell_2) \}, \quad (6.21)$$

em que $\mathbf{F}_{4N}(:, \ell)$ denota a ℓ -ésima coluna de $\mathbf{F}_{4N}$, $\Re(\cdot)$ extrai a parte real e apenas as primeiras N componentes são consideradas. Cada coluna da matriz DCT pode, portanto, ser obtida a partir da soma de duas colunas simétricas da DFT de dimensão $4N$, seguida da extração da parte real e da restrição às primeiras N linhas. Essa relação estabelece uma conversão matricial direta:

$$\hat{\mathbf{C}}_N \longleftarrow \hat{\mathbf{F}}_{4N}.$$

6.3.4 Algoritmo

O Algoritmo 2 apresenta o procedimento computacional para a conversão de uma DFT aproximada de dimensão $4N \times 4N$ ($\hat{\mathbf{F}}_{4N}$) em uma DCT aproximada de tamanho $N \times N$ ($\hat{\mathbf{C}}_N$).

Algoritmo 2: Conversão de $\widehat{\mathbf{F}}_{4N}$ em $\widehat{\mathbf{C}}_N$.

```

1 Função DFTparaDCT( $\widehat{\mathbf{F}}_{4N}, N$ ):
    Entrada: DFT aproximada  $\widehat{\mathbf{F}}_{4N}$  de dimensão  $4N \times 4N$ .
    Saída: DCT aproximada  $\widehat{\mathbf{C}}_N$  de dimensão  $N \times N$ .

2    $\widehat{\mathbf{C}}_N \leftarrow \mathbf{0}_{N \times N}$ 
3    $\widetilde{\mathbf{F}} \leftarrow \widehat{\mathbf{F}}_{4N}[0:N-1, :]$ 
4   for  $j \leftarrow 0$  to  $N-1$  do
5        $k_1 \leftarrow 2j+1$ 
6        $k_2 \leftarrow 4N-(2j+1)$ 
7        $\mathbf{s} \leftarrow \widetilde{\mathbf{F}}[:, k_1] + \widetilde{\mathbf{F}}[:, k_2]$ 
8        $\widehat{\mathbf{C}}_N[:, j] \leftarrow \lfloor \Re\{\mathbf{s}\}/2 \rfloor$ 
9   return  $\widehat{\mathbf{C}}_N$ 

```

Fonte: O autor (2026).

O algoritmo utiliza a simetria conjugada da DFT, de modo que a soma das colunas associadas aos índices ℓ_1 e ℓ_2 cancela as componentes imaginárias no caso exato — ou as reduz no caso aproximado — resultando em uma matriz real. A etapa de divisão por 2 seguida de arredondamento ao inteiro mais próximo preserva a baixa complexidade computacional da aproximação e evita o uso de fatores de normalização irracionais, como $\sqrt{2/N}$, presentes na DCT normalizada.

7 RESULTADOS

Neste capítulo são apresentados os resultados obtidos pelo método descrito no Capítulo 6. As aproximações propostas são avaliadas segundo as figuras de mérito definidas anteriormente — erro total de energia (ϵ), erro quadrático médio (MSE), ganho de codificação (C_g^*), eficiência de transformada (η) e desvio de ortogonalidade (ϕ) — e comparadas com aproximações amplamente reportadas na literatura.

As matrizes de baixa complexidade obtidas diretamente pelo método de conversão são denotadas por $\mathbf{T}_N^{(p)}$, em que o sobrescrito (p) indica o espaço de busca $\mathcal{P}'_p$ utilizado. Para obter a aproximação final $\widehat{\mathbf{C}}_N^{(p)}$, é necessário aplicar um processo de normalização sobre $\mathbf{T}_N^{(p)}$, cuja formulação depende da ortogonalidade da matriz. Quando $\mathbf{T}_N^{(p)}$ é ortogonal, isto é, $\mathbf{T}_N^{(p)} \cdot \mathbf{T}_N^{(p)\top} = \mathbf{D}_N$ para alguma matriz diagonal $\mathbf{D}_N$, a aproximação final é dada pelo processo de ortogonalização, discutido na Seção 5.3. Quando $\mathbf{T}_N^{(p)}$ não satisfaz a condição de ortogonalidade, aplica-se o processo de quase-ortogonalização, conforme descrito na Seção 5.3. Em ambos os casos, as fatorações e diagramas de fluxo apresentados neste capítulo referem-se à matriz $\mathbf{T}_N^{(p)}$, que é a estrutura efetivamente implementada em hardware.

Aplicando o procedimento de conversão às matrizes DFT aproximadas obtidas na etapa anterior, para dois tamanhos de transformada e três espaços de busca, obtêm-se os seguintes resultados:

- $\widehat{\mathbf{F}}_{16}^{(1)} \rightarrow \widehat{\mathbf{C}}_4^{(1)}$, com elementos em S_1 ;
- $\widehat{\mathbf{F}}_{16}^{(2)} \rightarrow \widehat{\mathbf{C}}_4^{(2)}$, com elementos em S_2 ;
- $\widehat{\mathbf{F}}_{32}^{(1)} \rightarrow \widehat{\mathbf{C}}_8^{(1)}$, com elementos em S_1 ;
- $\widehat{\mathbf{F}}_{32}^{(2)} \rightarrow \widehat{\mathbf{C}}_8^{(2)}$, com elementos em S_2 ;
- $\widehat{\mathbf{F}}_{32}^{(4)} \rightarrow \widehat{\mathbf{C}}_8^{(4)}$, com elementos em S_4 .

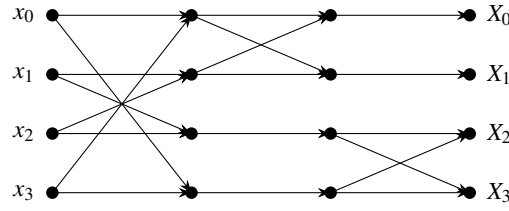
As aproximações $\widehat{\mathbf{F}}_{16}^{(1)}$, $\widehat{\mathbf{F}}_{16}^{(2)}$, $\widehat{\mathbf{F}}_{32}^{(1)}$, $\widehat{\mathbf{F}}_{32}^{(2)}$ e $\widehat{\mathbf{F}}_{32}^{(4)}$ estão explícitas no Apêndice A. As subseções a seguir apresentam, para cada comprimento N , as matrizes propostas, suas fatorações, diagramas de fluxo, complexidade aritmética e a comparação de desempenho com o estado da arte.

7.1 APROXIMAÇÕES PROPOSTAS

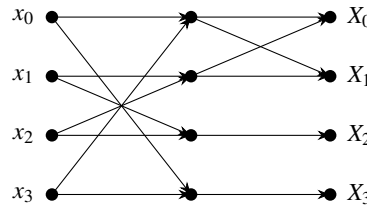
Nesta seção, são comparadas as aproximações obtidas com aquelas presentes no estado da arte.

7.1.1 Comparação para $N = 4$

Para o comprimento $N = 4$, foram obtidas duas matrizes de baixa complexidade, correspondentes aos espaços de busca S_1 e S_2 :

Figura 5 – Diagrama de fluxo da matriz $\mathbf{T}_4^{(1)}$.

Fonte: O autor (2026).

Figura 6 – Diagrama de fluxo da matriz $\mathbf{T}_4^{(2)}$.

Fonte: O autor (2026).

$$\mathbf{T}_4^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}, \quad \mathbf{T}_4^{(2)} = \begin{bmatrix} 2 & 2 & 2 & 2 \\ 2 & 1 & -1 & -2 \\ 2 & -2 & -2 & 2 \\ 1 & -2 & 2 & -1 \end{bmatrix}.$$

As matrizes $\mathbf{T}_4^{(1)}$ e $\mathbf{T}_4^{(2)}$ admitem fatoração dada por:

$$\mathbf{T}_4^{(1)} = \mathbf{B}_4 \cdot \mathbf{B}_3^{(1)} \cdot \mathbf{B}_2 \cdot \mathbf{B}_1$$

e

$$\mathbf{T}_4^{(2)} = \mathbf{B}_3^{(2)} \cdot \mathbf{B}_2 \cdot \mathbf{B}_1,$$

em que as matrizes esparsas dos fatores são dadas por:

$$\mathbf{B}_1 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{B}_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_3^{(2)} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

$$\mathbf{B}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{B}_3^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix},$$

As figuras a seguir apresentam o diagrama de fluxo dos algoritmos rápidos das aproximações propostas. As figuras 5 e 6 ilustram respectivamente o diagrama de fluxo para $\mathbf{T}_4^{(1)}$ e $\mathbf{T}_4^{(2)}$.

Tabela 1 – Comparação de desempenho das aproximações da DCT para $N = 4$.

Método	ε	MSE	C_g^*	η
$\mathbf{C}_{\text{Exata}}$	0.0000	0.0000	7.5701	97.6966
$\hat{\mathbf{C}}_4^{(1)}$	0.9565	0.0073	7.1744	95.5737
$\hat{\mathbf{C}}_4^{(2)}$	0.0316	0.0002	7.5541	97.3301

Fonte: O autor (2026).

Tabela 2 – Complexidade aritmética das aproximações para $N = 4$.

Matriz	Multiplicações	Somas	Deslocamento de Bit	Conjunto
$\mathbf{T}_4^{(1)}$	0	8	0	$\mathcal{P}_1$
$\mathbf{T}_4^{(2)}$	0	10	4	$\mathcal{P}_2$

Fonte: O autor (2026).

A Tabela 1 apresenta o comparativo de desempenho das aproximações propostas para $N = 4$ em relação à DCT exata. Observa-se que ambas as aproximações são ortogonais. A aproximação proposta $\hat{\mathbf{C}}_4^{(2)}$ apresenta desempenho superior à $\hat{\mathbf{C}}_4^{(1)}$ em todas as figuras de mérito.

A Tabela 2 apresenta a complexidade aritmética associada às fatorações das matrizes $\mathbf{T}_4^{(1)}$ e $\mathbf{T}_4^{(2)}$.

7.1.2 Comparação para $N = 8$

Para o comprimento $N = 8$, foram obtidas aproximações de baixa complexidade $\mathbf{T}_8^{(1)}$, $\mathbf{T}_8^{(2)}$ e $\mathbf{T}_8^{(4)}$, com elementos nos conjuntos S_1 , S_2 e S_4 , respectivamente:

$$\mathbf{T}_8^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}, \quad \mathbf{T}_8^{(2)} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 1 & 0 & 0 & -1 & -2 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 0 & -2 & -1 & 1 & 2 & 0 & -2 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 1 & -2 & 0 & 2 & -2 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix}$$

e

$$\mathbf{T}_8^{(4)} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4 & 2 & 1 & -1 & -2 & -4 & -4 \\ 4 & 2 & -2 & -4 & -4 & -2 & 2 & 4 \\ 4 & -1 & -4 & -2 & 2 & 4 & 1 & -4 \\ 4 & -4 & -4 & 4 & 4 & -4 & -4 & 4 \\ 2 & -4 & 1 & 4 & -4 & -1 & 4 & -2 \\ 2 & -4 & 4 & -2 & -2 & 4 & -4 & 2 \\ 1 & -2 & 4 & -4 & 4 & -4 & 2 & -1 \end{bmatrix}$$

As aproximações $\mathbf{T}_8^{(1)}$, $\mathbf{T}_8^{(2)}$ e $\mathbf{T}_8^{(4)}$ admitem as seguintes fatorações:

$$\mathbf{T}_8^{(1)} = \mathbf{E}_4 \cdot \mathbf{E}_3 \cdot \mathbf{E}_2 \cdot \mathbf{E}_1,$$

$$\mathbf{T}_8^{(2)} = \mathbf{E}_4^{(2)} \cdot \mathbf{E}_3^{(2)} \cdot \mathbf{E}_2 \cdot \mathbf{E}_1,$$

e

$$\mathbf{T}_8^{(4)} = \mathbf{E}_4^{(4)} \cdot \mathbf{E}_3^{(2)} \cdot \mathbf{E}_2 \cdot \mathbf{E}_1,$$

em que $\mathbf{E}_1$ e $\mathbf{E}_2$ são comuns às três fatorações e $\mathbf{E}_3^{(2)}$ é compartilhado pelas fatorações de $\mathbf{T}_8^{(2)}$ e $\mathbf{T}_8^{(4)}$. Cada termo $\mathbf{E}_i$, $i = 1, 2, 3, 4$, é definido explicitamente abaixo:

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

$$\mathbf{E}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{E}_3^{(2)} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

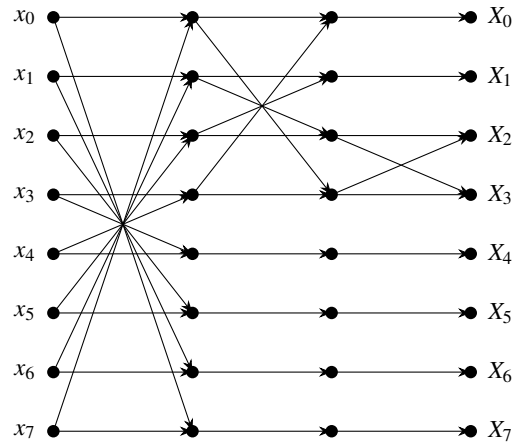
$$\mathbf{E}_4 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 0 \end{bmatrix}, \quad \mathbf{E}_4^{(2)} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -2 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & -2 & 1 \\ 0 & 0 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 2 & -1 & 0 \end{bmatrix},$$

e

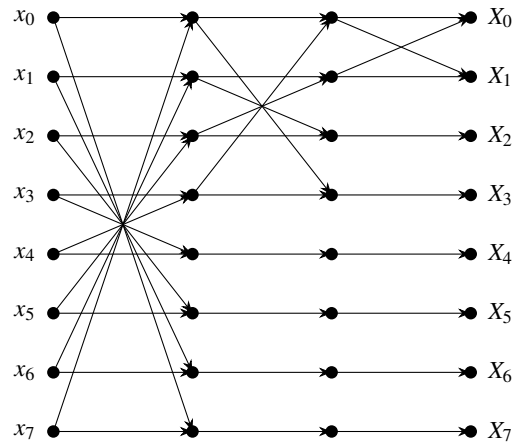
$$\mathbf{E}_4^{(4)} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 & 4 & 4 \\ 0 & 0 & 2 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & -4 & -1 & 4 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 1 & -4 & 2 \\ 0 & 0 & -4 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4 & 4 & -2 & 1 \end{bmatrix}$$

A Figura 7 ilustra o diagrama de fluxo da fatoração de $\mathbf{T}_8^{(1)}$ e a Figura 8, das fatorações de $\mathbf{T}_8^{(2)}$ e $\mathbf{T}_8^{(4)}$, que compartilham a mesma estrutura.

A Tabela 3 apresenta o comparativo de desempenho das aproximações propostas com diversas aproximações reportadas na literatura para $N = 8$. São apresentados os resultados para as aproximações propostas em (Oliveira *et al.*, 2018) ($\hat{\mathbf{C}}_{\text{OCBSML}_1}$ e $\hat{\mathbf{C}}_{\text{OCBSML}_2}$), (Lengwehasatit e Ortega, 2004) ($\hat{\mathbf{C}}_{\text{LO}}$), (Haweel, 2001) ($\hat{\mathbf{C}}_{\text{SDCT}}$), (Cintra e Bayer, 2011a) ($\hat{\mathbf{C}}_{\text{RDCT}}$), (Bayer e Cintra, 2010) ($\hat{\mathbf{C}}_{\text{MRDCT}}$), (Bouguezal, Ahmad e Swamy, 2008) ($\hat{\mathbf{C}}_{\text{BAS-2008}}$) e (Tablada, Cintra e Bayer, 2014) ($\hat{\mathbf{C}}_{\text{TCB}_6}$).

Figura 7 – Diagrama de fluxo da matriz $\mathbf{T}_8^{(1)}$.

Fonte: O autor (2026).

Figura 8 – Diagrama de fluxo das matrizes $\mathbf{T}_8^{(2)}$ e $\mathbf{T}_8^{(4)}$.

Fonte: O autor (2026).

Para o comprimento $N = 8$, conforme ilustrado na Tabela 3, os resultados obtidos pelas aproximações propostas mostram-se competitivos em relação aos resultados reportados na literatura. No espaço S_1 , a aproximação $\hat{\mathbf{C}}_8^{(1)}$ apresenta desempenho equivalente ao da $\hat{\mathbf{C}}_{\text{RDCT}}$ proposta por Cintra e Bayer (2011a), com métricas quase idênticas. Esse resultado sugere que o método de conversão, aplicado ao conjunto $\mathcal{P}'_1$, resulta em estruturas próximas daquelas já conhecidas na literatura.

Em S_2 , a aproximação $\hat{\mathbf{C}}_8^{(2)}$ supera todas as aproximações ortogonais da literatura em erro total de energia (ε) e MSE. Com relação às métricas de codificação, a aproximação $\hat{\mathbf{C}}_8^{(2)}$ só é superada por $\hat{\mathbf{C}}_{\text{OCBSML}_1}$, cujo desempenho em codificação é o maior reportado na literatura em S_2 .

A aproximação $\hat{\mathbf{C}}_8^{(4)}$, em S_4 , tem o melhor desempenho dentre todas as aproximações avaliadas, com valores mais próximos da DCT exata dentre os métodos comparados. A progressão das métricas ao longo dos conjuntos S_1, S_2 e S_4 evidencia um comportamento monotônico de

Tabela 3 – Comparação de desempenho de aproximações da DCT para $N = 8$.

Método	ε	MSE	C_g^*	η	ϕ	Referência
$\mathbf{C}_{\text{Exata}}$	0.0000	0.0000	8.8259	93.9912	0	-
$\hat{\mathbf{C}}_{\text{OCBSML}_1}$	1.2194	0.0046	8.6337	90.4615	0	Oliveira <i>et al.</i> (2018)
$\hat{\mathbf{C}}_{\text{OCBSML}_2}$	1.2194	0.0127	8.1024	87.2275	0	Oliveira <i>et al.</i> (2018)
$\hat{\mathbf{C}}_{\text{LO}}$	0.8695	0.0061	8.3902	88.7023	0	Lengwehasatit e Ortega (2004)
$\hat{\mathbf{C}}_{\text{SDCT}}$	3.3158	0.0207	6.0261	82.6190	0	Haweel (2001)
$\hat{\mathbf{C}}_{\text{RDCT}}$	1.7945	0.0098	8.1827	87.4297	0	Cintra e Bayer (2011a)
$\hat{\mathbf{C}}_{\text{MRDCT}}$	8.6592	0.0594	7.3326	80.8969	0	Bayer e Cintra (2010)
$\hat{\mathbf{C}}_{\text{BAS-2008}}$	4.1875	0.0191	6.2684	83.1734	0	Bouguezal, Ahmad e Swamy (2008)
$\hat{\mathbf{C}}_{\text{TCB}_6}$	0.8695	0.0062	8.3437	88.0594	0	Tablada, Cintra e Bayer (2014)
$\hat{\mathbf{C}}_8^{(1)}$	1.7944	0.0098	8.1834	87.1566	0	Proposta
$\hat{\mathbf{C}}_8^{(2)}$	0.4022	0.0027	8.5879	90.1603	0.0238	Proposta
$\hat{\mathbf{C}}_8^{(4)}$	0.1691	0.0011	8.7248	91.9695	0.0015	Proposta

Fonte: O autor (2026).

melhora: à medida que o espaço de busca é ampliado, a similaridade angular dos candidatos aumenta, resultando em melhores aproximações. Cabe destacar que todas as aproximações da literatura listadas na Tabela 3 são ortogonais. Dentre as aproximações propostas, apenas $\hat{\mathbf{C}}_8^{(1)}$ satisfaz essa condição. As aproximações $\hat{\mathbf{C}}_8^{(2)}$ e $\hat{\mathbf{C}}_8^{(4)}$, embora não-ortogonais, apresentam valores baixos de desvio de ortogonalidade.

A Tabela 4 apresenta a complexidade aritmética das aproximações propostas e daquelas reportadas na literatura para $N = 8$.

Tabela 4 – Comparação de complexidade computacional para $N = 8$.

Método	Multiplicações	Somas	Deslocamento de Bit	Conjunto
$\hat{\mathbf{C}}_{\text{OCBSML}_1}$	0	24	6	$\mathcal{P}_2$
$\hat{\mathbf{C}}_{\text{OCBSML}_2}$	0	24	6	$\mathcal{P}_2$
$\hat{\mathbf{C}}_{\text{LO}}$	0	24	2	$\mathcal{P}_2$
$\hat{\mathbf{C}}_{\text{SDCT}}$	0	24	0	$\mathcal{P}_1$
$\hat{\mathbf{C}}_{\text{RDCT}}$	0	22	0	$\mathcal{P}_1$
$\hat{\mathbf{C}}_{\text{MRDCT}}$	0	14	0	$\mathcal{P}_1$
$\hat{\mathbf{C}}_{\text{BAS-2008}}$	0	18	0	$\mathcal{P}_1$
$\hat{\mathbf{C}}_{\text{TCB}_6}$	0	24	6	$\mathcal{P}_1$
$\hat{\mathbf{C}}_8^{(1)}$	0	24	0	$\mathcal{P}_1$
$\hat{\mathbf{C}}_8^{(2)}$	0	24	12	$\mathcal{P}_2$
$\hat{\mathbf{C}}_8^{(4)}$	0	24	28	$\mathcal{P}_4$

Fonte: O autor (2026).

Todas as aproximações avaliadas são livres de multiplicações, requisito fundamental para implementação em hardware de baixo consumo energético. No conjunto S_1 , a aproximação $\hat{\mathbf{C}}_8^{(1)}$ requer 24 somas e nenhum deslocamento de bit, apresentando complexidade comparável à $\hat{\mathbf{C}}_{\text{SDCT}}$ e inferior à $\hat{\mathbf{C}}_{\text{RDCT}}$ (22 somas). Dentre as aproximações do conjunto S_1 , a $\hat{\mathbf{C}}_{\text{MRDCT}}$ apresenta a menor complexidade (14 somas), porém ao custo de um desempenho significativamente inferior em métricas de erro.

As aproximações $\hat{\mathbf{C}}_8^{(2)}$ e $\hat{\mathbf{C}}_8^{(4)}$ mantêm o mesmo número de somas (24), mas com 12 e 28 deslocamentos de bits, respectivamente. Embora a quantidade de deslocamentos de bits da $\hat{\mathbf{C}}_8^{(4)}$ seja maior, essas operações possuem custo desprezível em hardware quando comparadas a operações multiplicativas, de modo que o incremento de complexidade pode ser justificado pela superioridade de seu desempenho.

Comparando com as aproximações da literatura no mesmo conjunto S_2 , as aproximações $\hat{\mathbf{C}}_{\text{OCBSML}_1}$, $\hat{\mathbf{C}}_{\text{OCBSML}_2}$ e $\hat{\mathbf{C}}_{\text{LO}}$ requerem 24 somas e entre 2 e 6 deslocamentos de bits. A aproximação proposta $\hat{\mathbf{C}}_8^{(2)}$ utiliza mais deslocamentos (12), porém apresenta desempenho superior em todas as figuras de mérito, conforme discutido anteriormente. O *trade-off* entre complexidade aritmética e desempenho da aproximação deve ser avaliado conforme os requisitos da aplicação.

8 ENSAIO COMPUTACIONAL

Este capítulo apresenta os resultados obtidos na avaliação das aproximações da DCT propostas neste trabalho. A validação experimental foi conduzida por meio de um processo de compressão de imagens do tipo JPEG, metodologia amplamente utilizada na literatura para aferir o desempenho de transformadas em aplicações práticas (Wallace, 1992; Britanak, Yip e Rao, 2007).

A Seção 8.1 descreve o processo de compressão tipo-JPEG empregado. A Seção 8.2 apresenta as métricas de qualidade de imagem utilizadas na avaliação. Por fim, a Seção 8.3 expõe os resultados numéricos e a análise comparativa das aproximações.

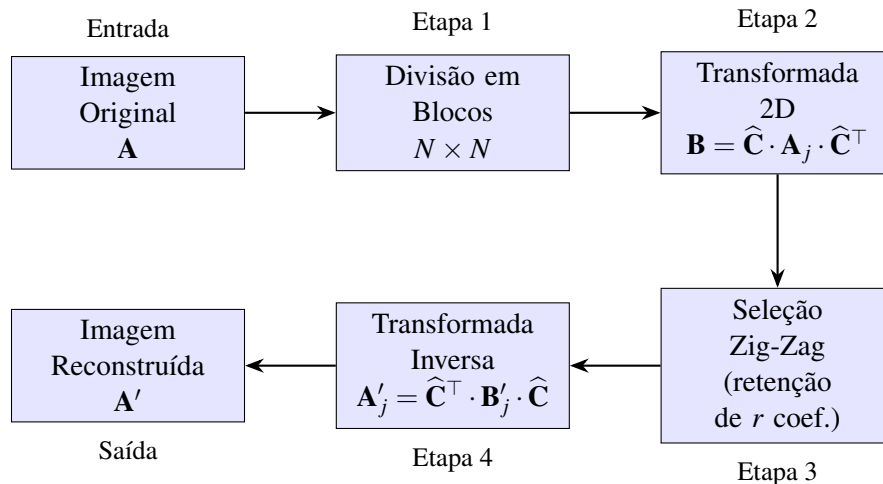
8.1 COMPRESSÃO DE IMAGENS TIPO-JPEG

O padrão JPEG (*Joint Photographic Experts Group*) constitui um dos formatos de compressão de imagens mais difundidos, empregando a DCT como núcleo do processo de transformação (Wallace, 1992). Neste trabalho, adota-se uma versão simplificada desse processo, denominada compressão tipo-JPEG, que preserva a etapa fundamental de transformação, permitindo uma avaliação direta do impacto das aproximações propostas na qualidade da imagem reconstruída.

8.1.1 Etapas do Processo de Compressão

O processo de compressão tipo-JPEG implementado neste trabalho é composto por cinco etapas sequenciais, ilustradas no diagrama da Figura 9.

Figura 9 – Diagrama do processo de compressão tipo-JPEG.



Fonte: Elaborado pelo autor (2026).

Etapa 1 – Segmentação em blocos: A imagem original $\mathbf{A}$, de dimensões $M \times M$ pixels, é subdividida em blocos disjuntos $\mathbf{A}_j$ de dimensão $N \times N$, em que N corresponde ao comprimento da transformada utilizada. Este particionamento resulta em $(M/N)^2$ blocos independentes que serão processados individualmente.

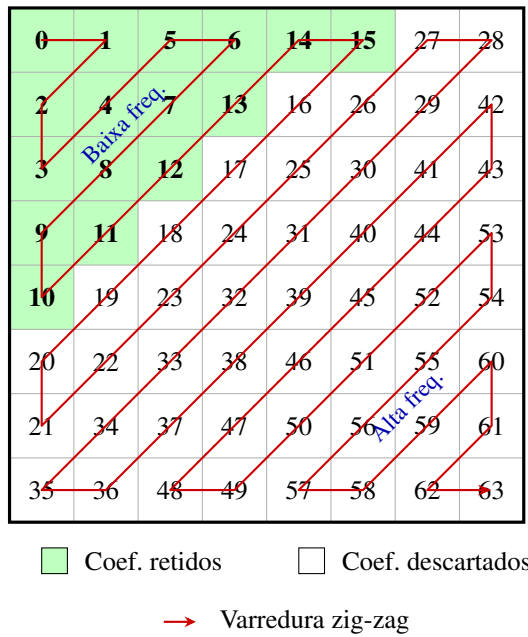
Etapa 2 – Transformação bidimensional: Cada bloco $\mathbf{A}_j$ é submetido à transformada bidimensional, produzindo o bloco de coeficientes espectrais $\mathbf{B}_j$:

$$\mathbf{B}_j = \hat{\mathbf{C}} \cdot \mathbf{A}_j \cdot \hat{\mathbf{C}}^\top, \quad (8.1)$$

em que $\hat{\mathbf{C}}$ representa a matriz de transformação (DCT exata ou aproximação).

Etapa 3 – Seleção de coeficientes via varredura zig-zag: Os coeficientes do bloco transformado são selecionados segundo o padrão de varredura *zig-zag*, ilustrado na Figura 10. Esta etapa agrupa os coeficientes de menor frequência (maior energia) nas primeiras posições. A seleção é implementada pela aplicação da máscara binária $\mathbf{M}_r$, que retém apenas os r primeiros coeficientes e anula os $N^2 - r$ restantes, produzindo o bloco truncado $\mathbf{B}'_j = \mathbf{M}_r \odot \mathbf{B}_j$. A taxa de retenção é definida como r/N^2 .

Figura 10 – Padrão de varredura zig-zag para bloco 8×8 com $r = 16$ coeficientes retidos.



Fonte: Elaborado pelo autor (2026).

Etapa 4 – Transformação inversa: O bloco truncado é convertido de volta ao domínio espacial por meio da transformada inversa:

$$\mathbf{A}'_j = \hat{\mathbf{C}}^\top \cdot \mathbf{B}'_j \cdot \hat{\mathbf{C}}, \quad (8.2)$$

assumindo que a matriz $\hat{\mathbf{C}}$ é ortogonal. Para matrizes não ortogonais, utiliza-se $\hat{\mathbf{C}}^{-1}$ no lugar de $\hat{\mathbf{C}}^\top$.

Etapa 5 – Reconstrução da imagem: Os blocos reconstruídos $\mathbf{A}'_j$ são reagrupados em suas posições originais, compondo a imagem comprimida $\mathbf{A}'$.

8.2 MÉTRICAS DE QUALIDADE DA IMAGEM

A avaliação da qualidade das imagens comprimidas foi realizada por meio de quatro métricas complementares, que capturam diferentes aspectos da fidelidade da reconstrução. Essas métricas são classificadas em dois grupos: métricas baseadas em erro e métricas perceptuais.

8.2.1 Métricas de Erro

8.2.1.1 MSE

O erro quadrático médio (MSE) quantifica a diferença ponto a ponto entre a imagem original $\mathbf{A}$ e a imagem reconstruída $\mathbf{A}'$ (Britanak, Yip e Rao, 2007):

$$\text{MSE} = \frac{1}{M^2} \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [a_{i,j} - a'_{i,j}]^2, \quad (8.3)$$

em que $a_{i,j}$ e $a'_{i,j}$ são os valores dos pixels nas posições (i, j) das imagens original e reconstruída, respectivamente, e M é a dimensão da imagem. Valores menores de MSE indicam maior fidelidade na reconstrução.

8.2.1.2 PSNR

A Relação Sinal-Ruído de Pico (PSNR) expressa a razão entre a máxima potência do sinal e a potência do ruído de distorção, em escala logarítmica (Huynh-Thu e Ghanbari, 2008):

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{L^2}{\text{MSE}} \right), \quad (8.4)$$

em que $L = 2^8 - 1$ para imagens de 8 bits. A PSNR é expressa em decibéis (dB), e valores maiores indicam melhor qualidade.

8.2.2 Métricas Perceptuais

As métricas perceptuais incorporam modelos que aproximam características do sistema visual humano, fornecendo avaliações mais alinhadas com a percepção subjetiva de qualidade.

8.2.2.1 UQI

O Índice Universal de Qualidade (UQI), proposto por Wang e Bovik (Wang e Bovik, 2002), modela a distorção como uma combinação de três fatores: perda de correlação, distorção de luminância e distorção de contraste:

$$\text{UQI} = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)(\bar{x}^2 + \bar{y}^2)}, \quad (8.5)$$

em que $\bar{x}$ e $\bar{y}$ são as médias das imagens original e reconstruída, σ_x^2 e σ_y^2 são as variâncias, e σ_{xy} é a covariância entre as imagens. O UQI assume valores no intervalo $[-1, 1]$, sendo que $\text{UQI} = 1$ indica imagens idênticas.

8.2.2.2 SSIM

O Índice de Similaridade Estrutural (SSIM), também proposto por Wang et al. (Wang *et al.*, 2004), estende o UQI ao incorporar constantes de estabilização que evitam instabilidades numéricas:

$$\text{SSIM} = \frac{(2\bar{x}\bar{y} + C_1)(2\sigma_{xy} + C_2)}{(\bar{x}^2 + \bar{y}^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (8.6)$$

em que $C_1 = (K_1L)^2$ e $C_2 = (K_2L)^2$ são constantes de estabilização, com $L = 255$ para imagens de 8 bits, e tipicamente $K_1 = 0.01$ e $K_2 = 0.03$. O SSIM também assume valores no intervalo $[-1, 1]$, com $\text{SSIM} = 1$ indicando similaridade perfeita. A Tabela 5 sintetiza as métricas utilizadas e seus critérios de interpretação.

Tabela 5 – Resumo das métricas de qualidade de imagem.

Métrica	Intervalo	Objetivo
MSE	$[0, \infty)$	Minimizar
PSNR	$(0, \infty)$ dB	Maximizar
UQI	$[-1, 1]$	Maximizar
SSIM	$[-1, 1]$	Maximizar

8.3 RESULTADOS NUMÉRICOS

A avaliação do desempenho das transformadas aproximadas propostas foi conduzida por meio de um experimento de compressão de imagens no estilo JPEG. O procedimento adotado consiste em submeter imagens ao algoritmo de compressão, empregando as aproximações propostas e avaliar a qualidade das imagens reconstruídas por meio de métricas objetivas. O desempenho da reconstrução é medido em relação a DCT exata. A Subseção 8.3.2 apresenta os resultados da avaliação experimental.

8.3.1 Imagens de teste

As imagens de teste utilizadas neste trabalho foram obtidas do *USC-SIPI Image Database* (USC-SIPI, 2024), repositório de acesso público mantido pelo *Signal and Image Processing Institute* da *University of Southern California*. Trata-se de uma base amplamente adotada na literatura de processamento de imagens e compressão, fornecendo um conjunto padronizado de sinais de referência que viabiliza a comparação dos resultados obtidos com aqueles reportados em outros trabalhos da área (Cintra e Bayer, 2011a; Bouguezal, Ahmad e Swamy, 2008).

Para o procedimento experimental, foram selecionadas nove imagens distribuídas em três volumes distintos da base USC-SIPI: Volume 1 (*Textures*), Volume 2 (*Aerials*) e Volume 3 (*Miscellaneous*). A seleção inclui imagens de naturezas distintas — texturas, fotografia aérea, objetos manufaturados — a fim de avaliar o desempenho das transformadas propostas em diferentes características estatísticas.

Conforme procedimento padrão em estudos de compressão por transformada (Cintra e Bayer, 2011a; Bouguezzel, Ahmad e Swamy, 2008), todas as imagens foram convertidas para escala de cinza mediante extração da componente de luminância. A Tabela 6 sumariza as características das imagens utilizadas.

Tabela 6 – Imagens de teste selecionadas do banco USC-SIPI.

Volume	ID	Descrição	Dimensão (px)
Vol. 1 — <i>Textures</i>	1.2.09	Brodatz - Wood grain (D68 H.E.)	512×512
	1.3.09	Brodatz - Wood grain (D68)	1024×1024
	1.4.09	Brick wall	1024×1024
Vol. 2 — <i>Aerials</i>	3.2.25	Pentagon	1024×1024
Vol. 3 — <i>Miscellaneous</i>	5.1.09	Moon surface	256×256
	5.1.11	Airplane	256×256
	5.1.12	Clock	256×256
	5.3.02	Airport	1024×1024
	7.1.01	Truck	512×512

Fonte: adaptado de USC-SIPI (USC-SIPI, 2024).

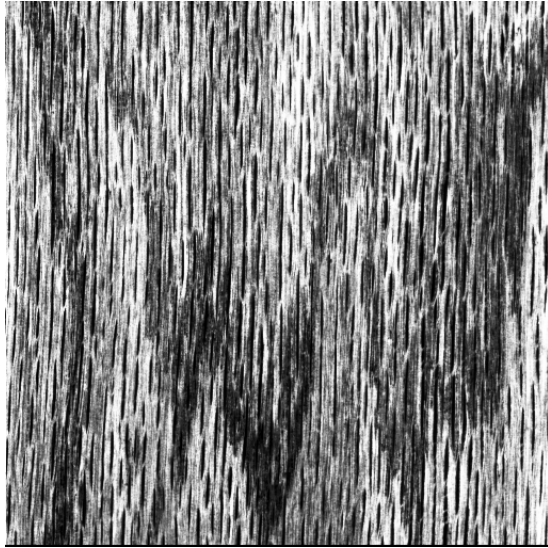
8.3.2 Procedimento experimental

As imagens da Tabela 6 foram submetidas ao processo de compressão descrito na Seção 8.1.1, empregando as aproximações $\hat{\mathbf{C}}_4^{(1)}$, $\hat{\mathbf{C}}_4^{(2)}$, $\hat{\mathbf{C}}_8^{(1)}$, $\hat{\mathbf{C}}_8^{(2)}$ e $\hat{\mathbf{C}}_8^{(4)}$.

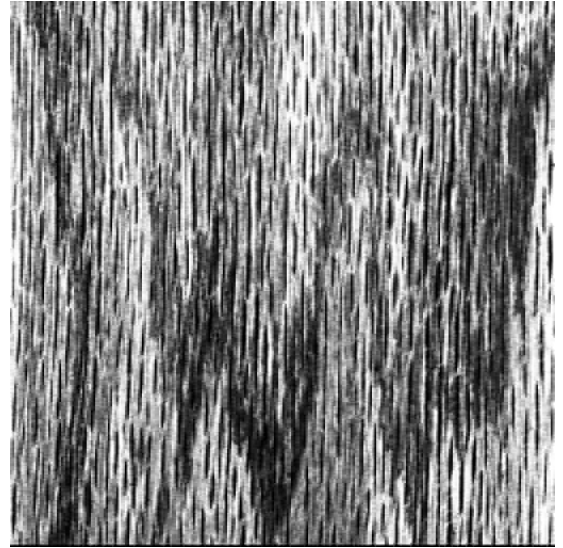
Aplicou-se a máscara binária *zigzag* $\mathbf{M}_r$, que retém exclusivamente os r coeficientes de menor frequência, conforme ilustrado na Figura 10. Adotou-se a taxa de retenção $r = \lfloor 0,25 \cdot N^2 \rfloor$, que corresponde à preservação de 25% dos coeficientes. Para os tamanhos considerados, $r = 4$ quando $N = 4$ e $r = 16$ para $N = 8$.

Finalmente, os valores reconstruídos foram arredondados e recortados ao intervalo $[0, 255]$, preservando a representação em 8 bits por pixel. A qualidade da imagem recuperada foi quantificada pelas métricas descritas na Seção 8.2, computadas entre a imagem original e a imagem reconstruída. Os resultados visuais da compressão encontram-se nas figuras 11–28. Os valores das métricas de qualidade de todas as imagens para $N = 4$ e $N = 8$ estão listados nas tabelas 7 e 8, respectivamente.

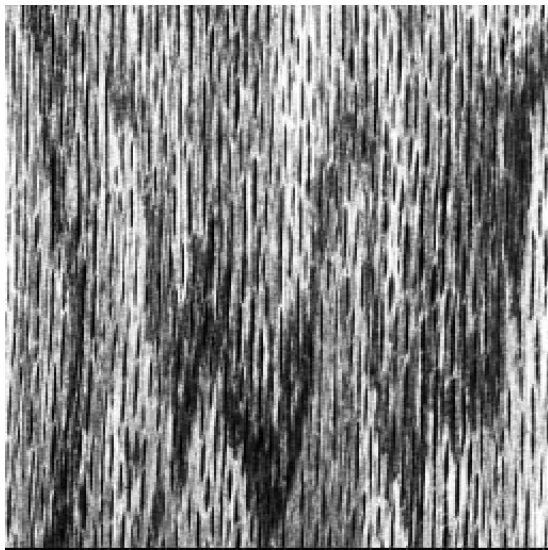
Figura 11 – Comparação da compressão de imagens ($N = 4$) para a imagem 1.2.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$.



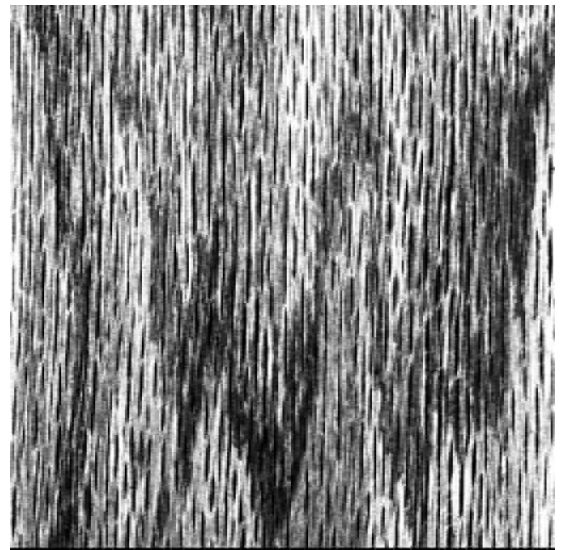
(a)



(b)



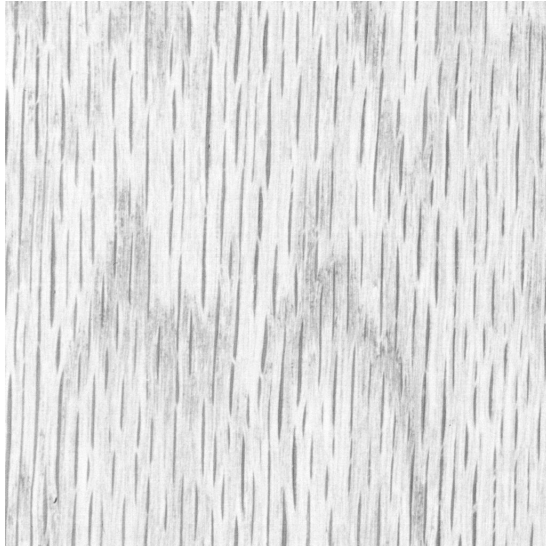
(c)



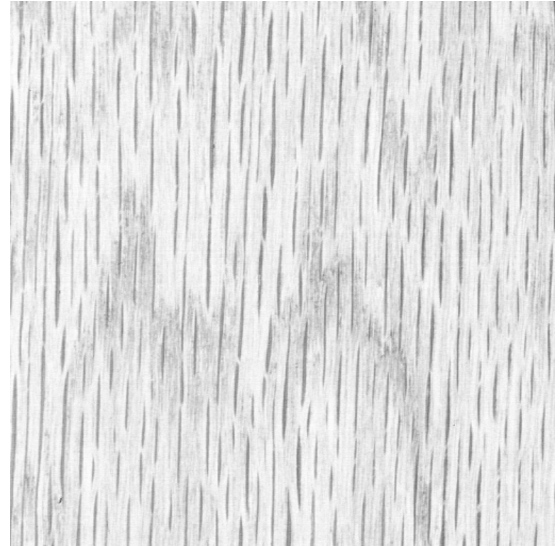
(d)

Fonte: O autor (2026).

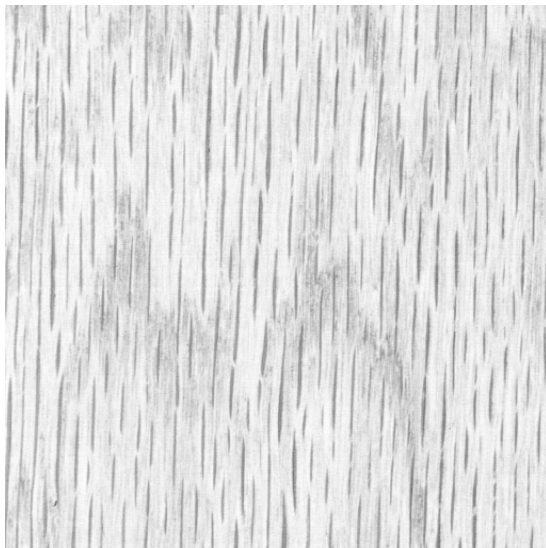
Figura 12 – Comparação da compressão de imagens ($N = 4$) para a imagem 1.3.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$.



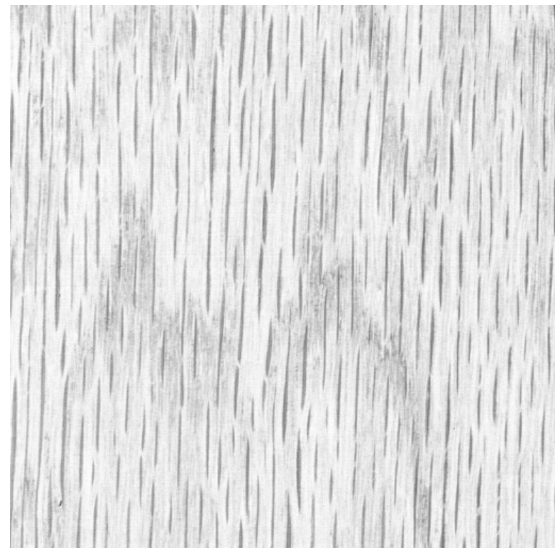
(a)



(b)



(c)



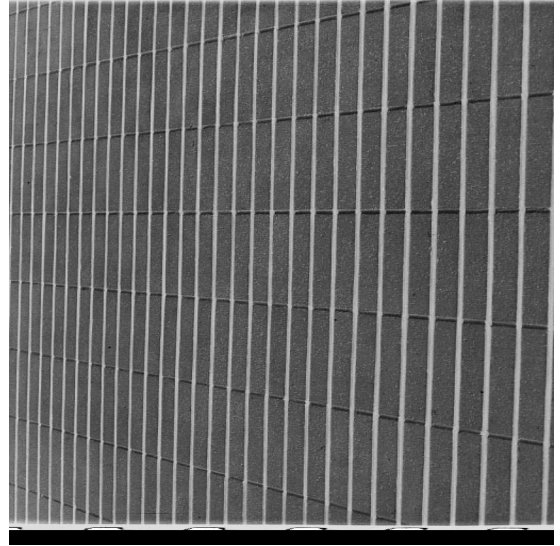
(d)

Fonte: O autor (2026).

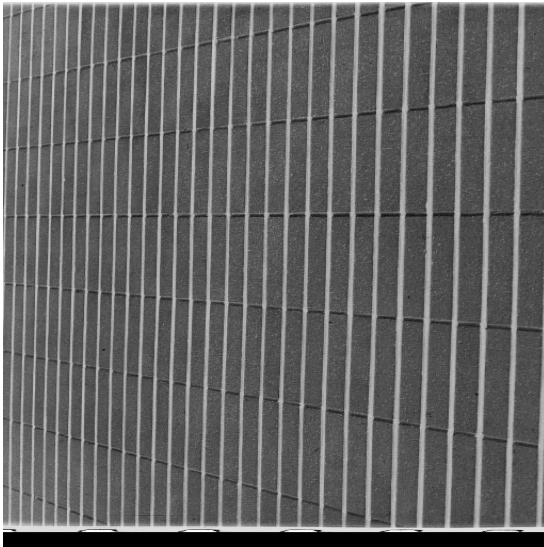
Figura 13 – Comparação da compressão de imagens ($N = 4$) para a imagem 1.4.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$.



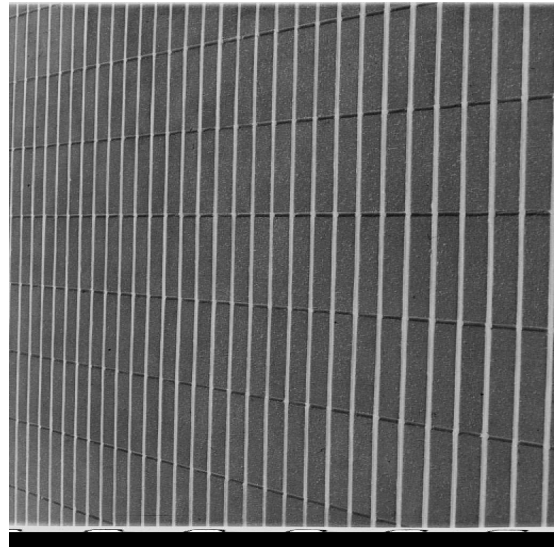
(a)



(b)



(c)



(d)

Fonte: O autor (2026).

Figura 14 – Comparação da compressão de imagens ($N = 4$) para a imagem 3.2.25: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$.



(a)



(b)



(c)



(d)

Fonte: O autor (2026).

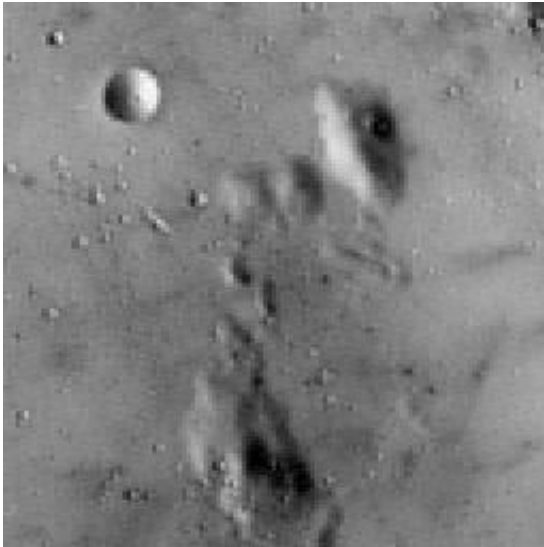
Figura 15 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.1.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$.



(a)



(b)



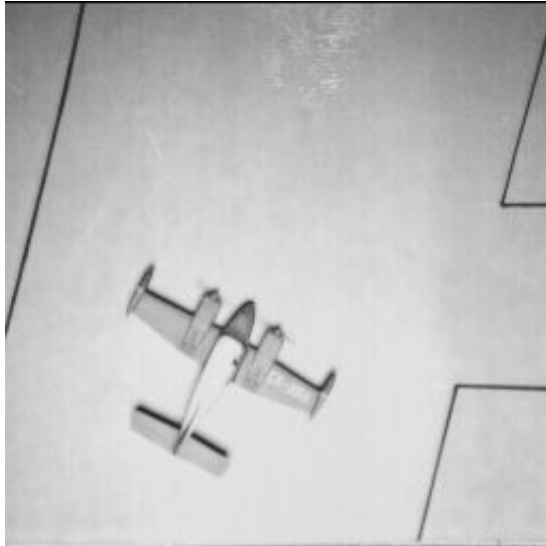
(c)



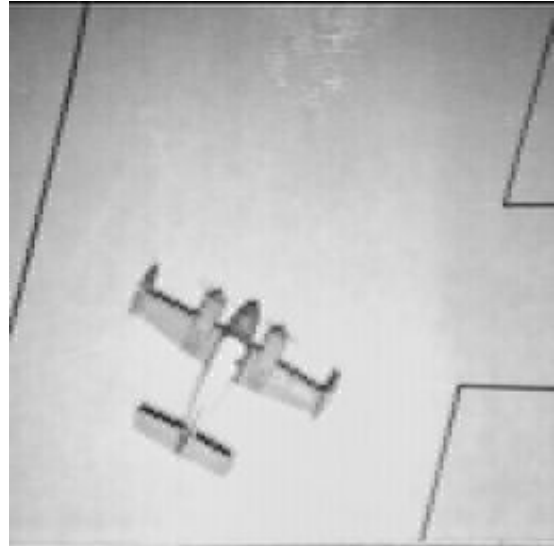
(d)

Fonte: O autor (2026).

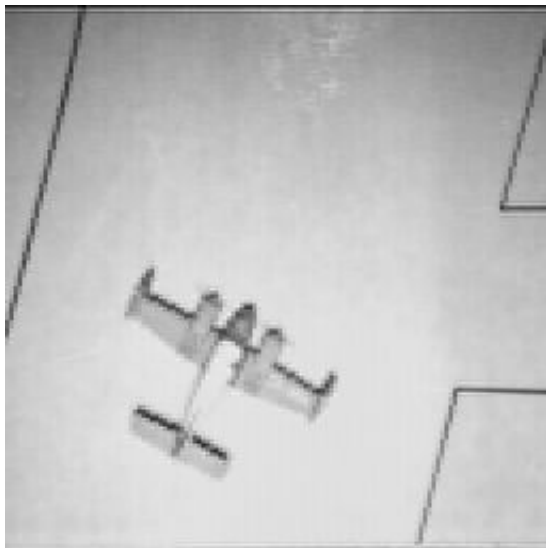
Figura 16 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.1.11: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$.



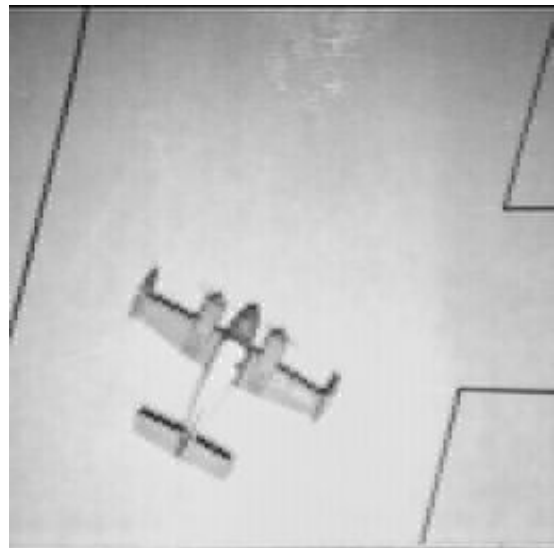
(a)



(b)



(c)



(d)

Fonte: O autor (2026).

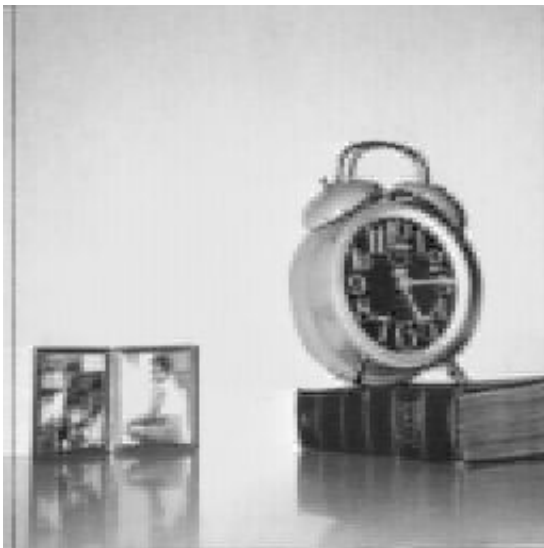
Figura 17 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.1.12: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_4^{(1)}$ e (d) aproximação $\hat{C}_4^{(2)}$.



(a)



(b)



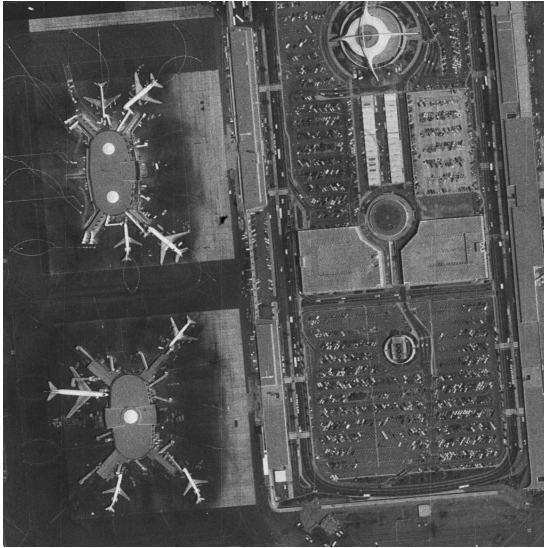
(c)



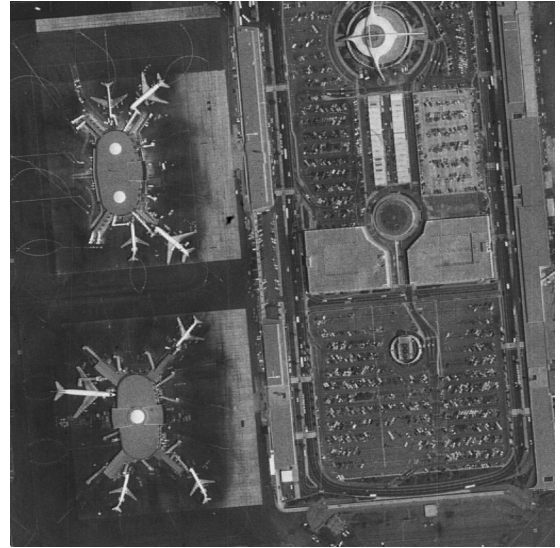
(d)

Fonte: O autor (2026).

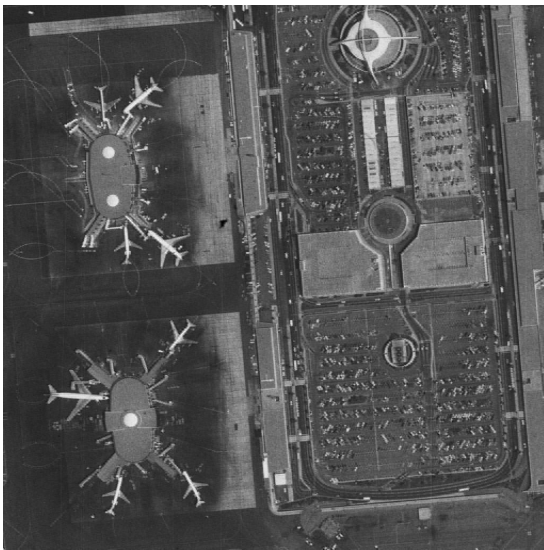
Figura 18 – Comparação da compressão de imagens ($N = 4$) para a imagem 5.3.02: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_4^{(1)}$ e (d) aproximação $\hat{C}_4^{(2)}$.



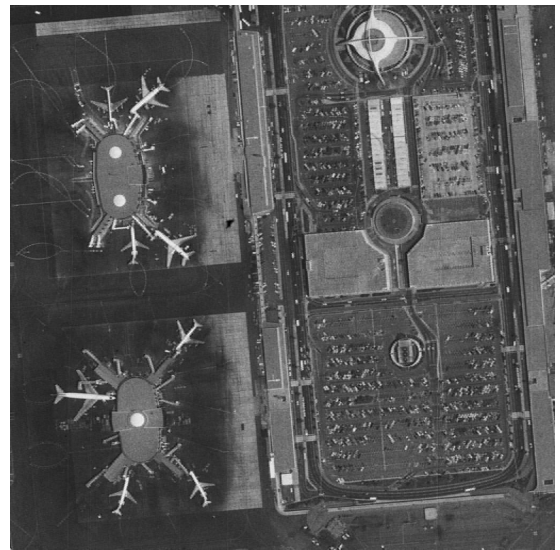
(a)



(b)



(c)



(d)

Fonte: O autor (2026).

Figura 19 – Comparação da compressão de imagens ($N = 4$) para a imagem 7.1.01: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_4^{(1)}$ e (d) aproximação $\hat{\mathbf{C}}_4^{(2)}$.



(a)



(b)



(c)



(d)

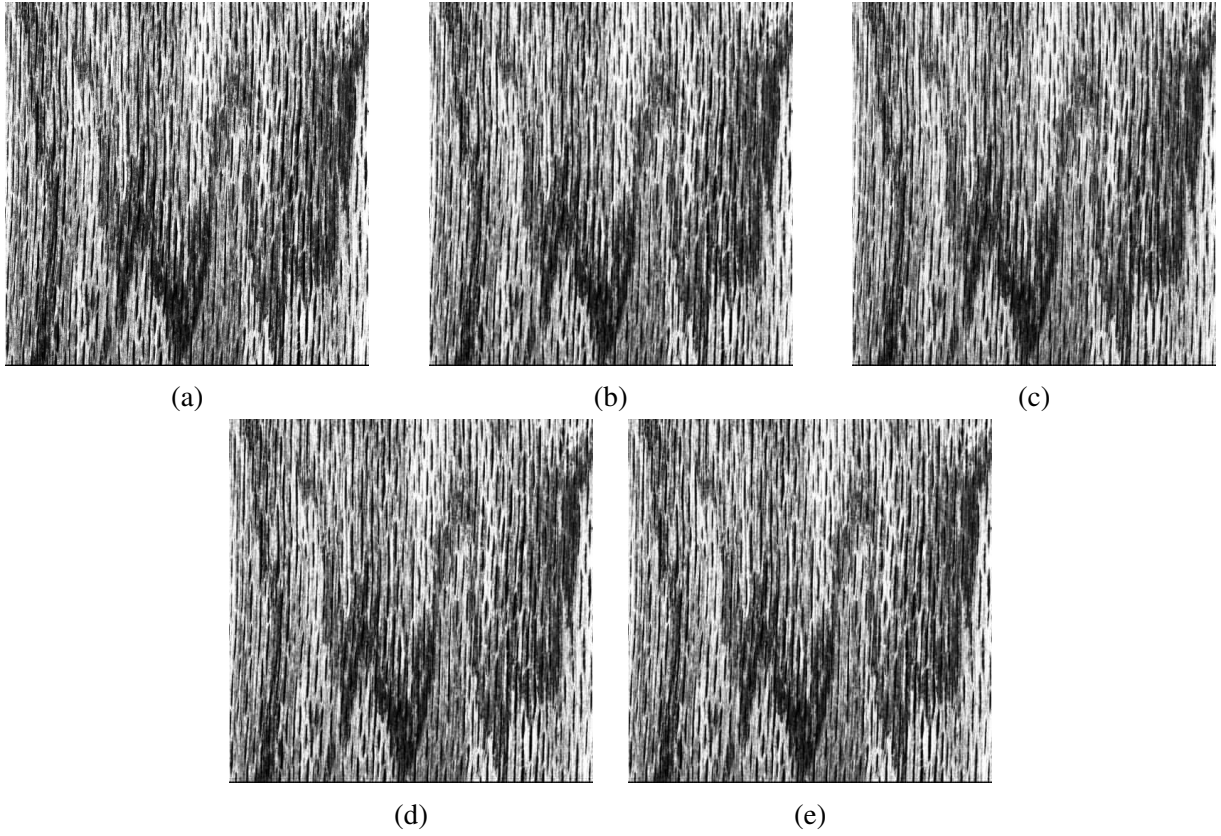
Fonte: O autor (2026).

Tabela 7 – Métricas de qualidade da compressão de imagens para ($N = 4$, $r = 4$).

Imagem	Transformada	PSNR (dB)	SSIM	MSE	UQI
1.2.09	DCT	18.539	0.7924	910.270	0.9101
	$\hat{\mathbf{C}}_4^{(1)}$	17.947	0.7714	1043.115	0.8965
	$\hat{\mathbf{C}}_4^{(2)}$	18.552	0.7937	907.671	0.9106
1.3.09	DCT	33.299	0.8852	30.418	0.9716
	$\hat{\mathbf{C}}_4^{(1)}$	31.740	0.8657	43.555	0.9588
	$\hat{\mathbf{C}}_4^{(2)}$	33.218	0.8849	30.992	0.9710
1.4.09	DCT	27.068	0.7496	127.733	0.9633
	$\hat{\mathbf{C}}_4^{(1)}$	26.276	0.7365	153.292	0.9556
	$\hat{\mathbf{C}}_4^{(2)}$	27.046	0.7503	128.382	0.9631
3.2.25	DCT	29.186	0.8130	78.435	0.9429
	$\hat{\mathbf{C}}_4^{(1)}$	28.728	0.7993	87.160	0.9361
	$\hat{\mathbf{C}}_4^{(2)}$	29.186	0.8132	78.427	0.9429
5.1.09	DCT	31.065	0.7836	50.882	0.9658
	$\hat{\mathbf{C}}_4^{(1)}$	30.571	0.7730	57.012	0.9615
	$\hat{\mathbf{C}}_4^{(2)}$	31.046	0.7836	51.108	0.9656
5.1.11	DCT	32.739	0.9463	34.609	0.9839
	$\hat{\mathbf{C}}_4^{(1)}$	30.660	0.9348	55.851	0.9738
	$\hat{\mathbf{C}}_4^{(2)}$	32.477	0.9453	36.759	0.9829
5.1.12	DCT	29.553	0.9385	72.068	0.9889
	$\hat{\mathbf{C}}_4^{(1)}$	28.150	0.9258	99.562	0.9846
	$\hat{\mathbf{C}}_4^{(2)}$	29.469	0.9385	73.487	0.9887
5.3.02	DCT	28.760	0.8199	86.512	0.9644
	$\hat{\mathbf{C}}_4^{(1)}$	28.005	0.8057	102.933	0.9573
	$\hat{\mathbf{C}}_4^{(2)}$	28.744	0.8203	86.834	0.9643
7.1.01	DCT	33.747	0.8990	27.441	0.9809
	$\hat{\mathbf{C}}_4^{(1)}$	32.817	0.8828	33.992	0.9762
	$\hat{\mathbf{C}}_4^{(2)}$	33.724	0.8989	27.587	0.9808

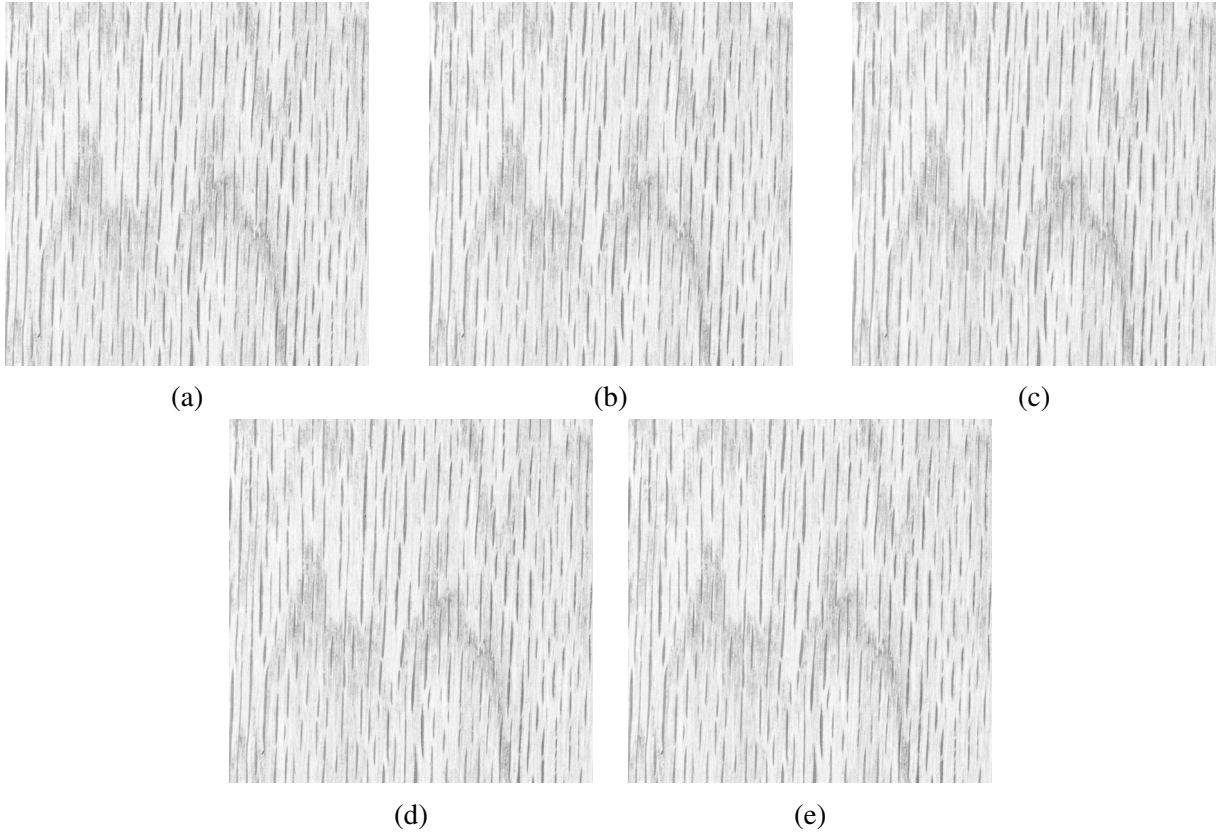
Fonte: O autor (2026).

Figura 20 – Comparação da compressão de imagens ($N = 8$) para a imagem 1.2.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$.



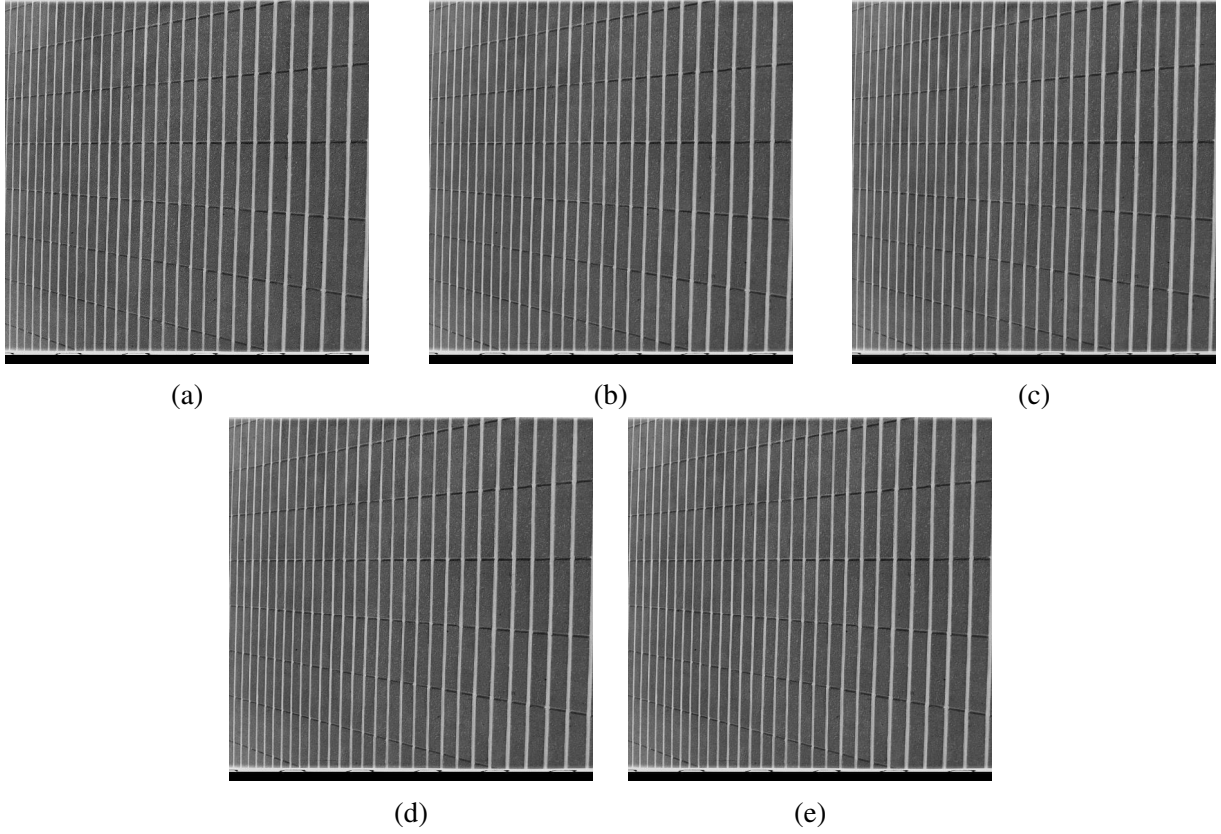
Fonte: O autor (2026).

Figura 21 – Comparação da compressão de imagens ($N = 8$) para a imagem 1.3.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_8^{(1)}$; (d) aproximação $\hat{\mathbf{C}}_8^{(2)}$ e (e) aproximação $\hat{\mathbf{C}}_8^{(4)}$.



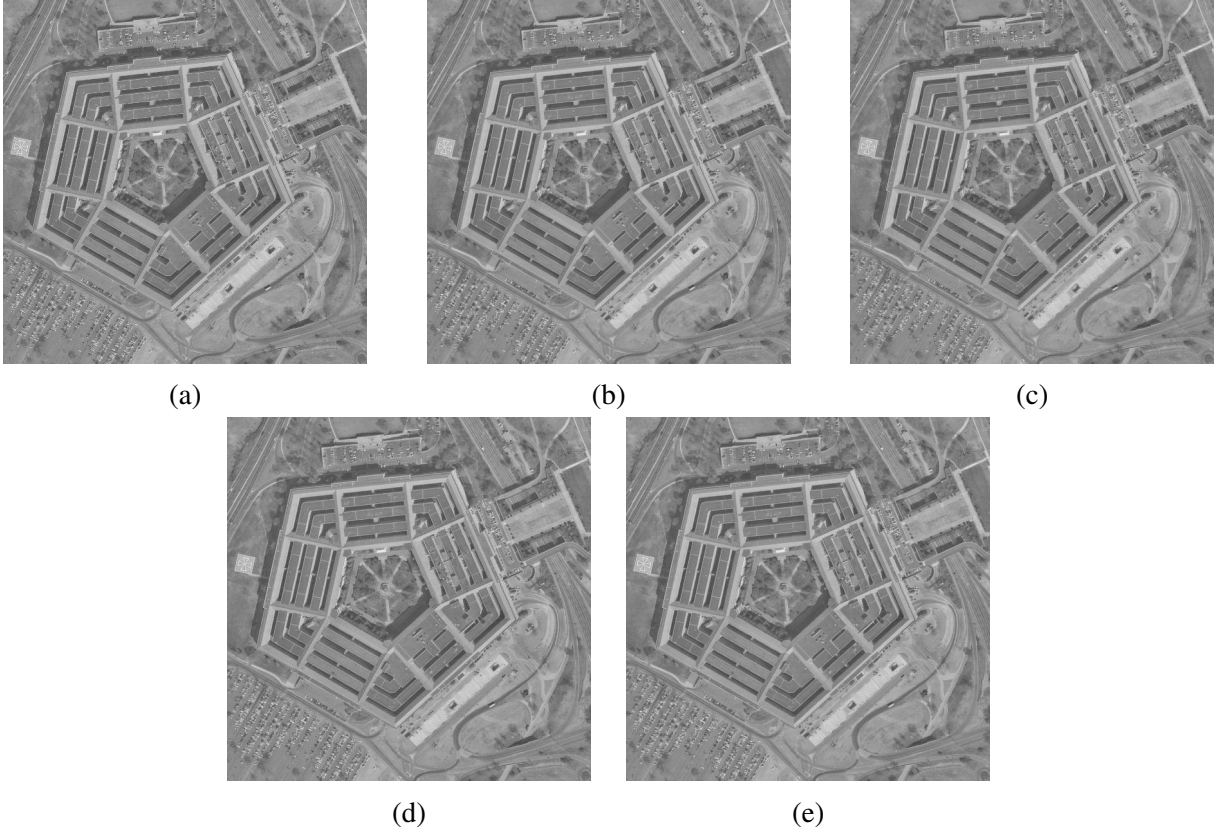
Fonte: O autor (2026).

Figura 22 – Comparação da compressão de imagens ($N = 8$) para a imagem 1.4.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_8^{(1)}$; (d) aproximação $\hat{\mathbf{C}}_8^{(2)}$ e (e) aproximação $\hat{\mathbf{C}}_8^{(4)}$.



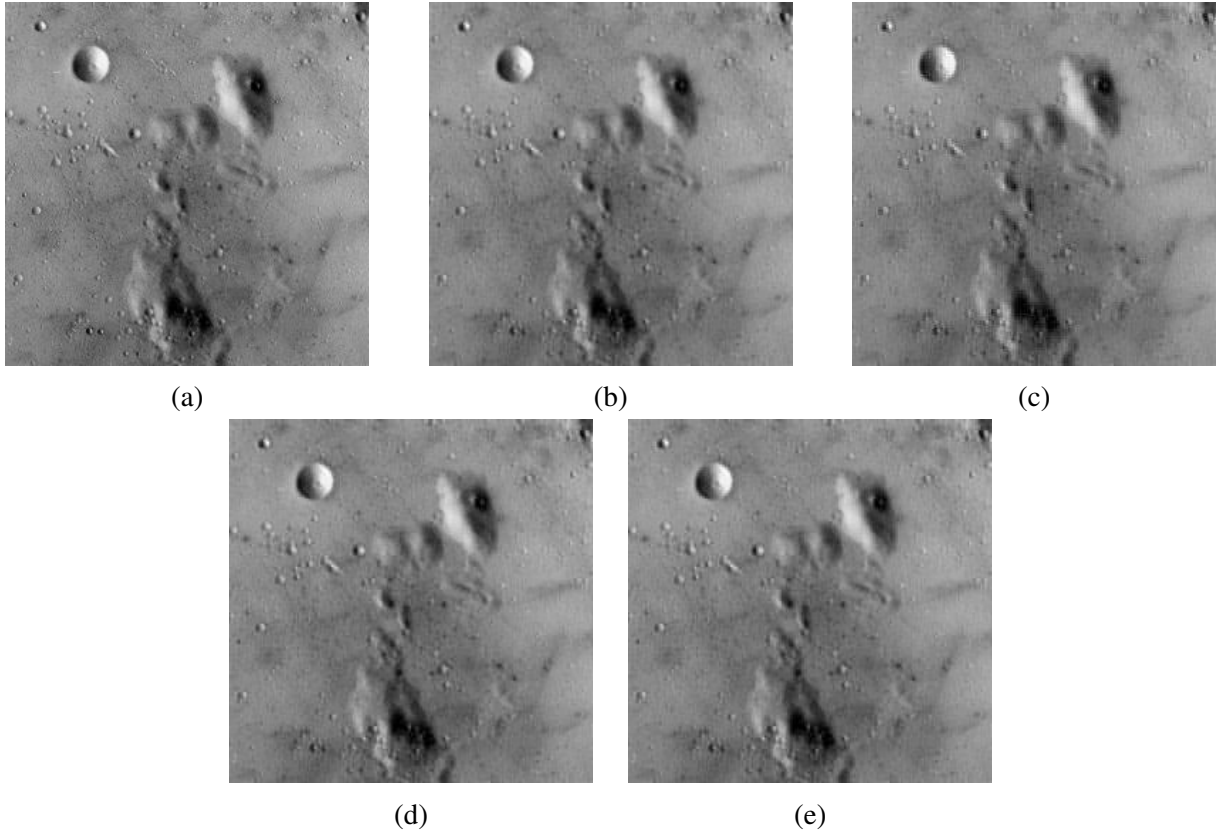
Fonte: O autor (2026).

Figura 23 – Comparação da compressão de imagens ($N = 8$) para a imagem 3.2.25: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_8^{(1)}$; (d) aproximação $\hat{\mathbf{C}}_8^{(2)}$ e (e) aproximação $\hat{\mathbf{C}}_8^{(4)}$.



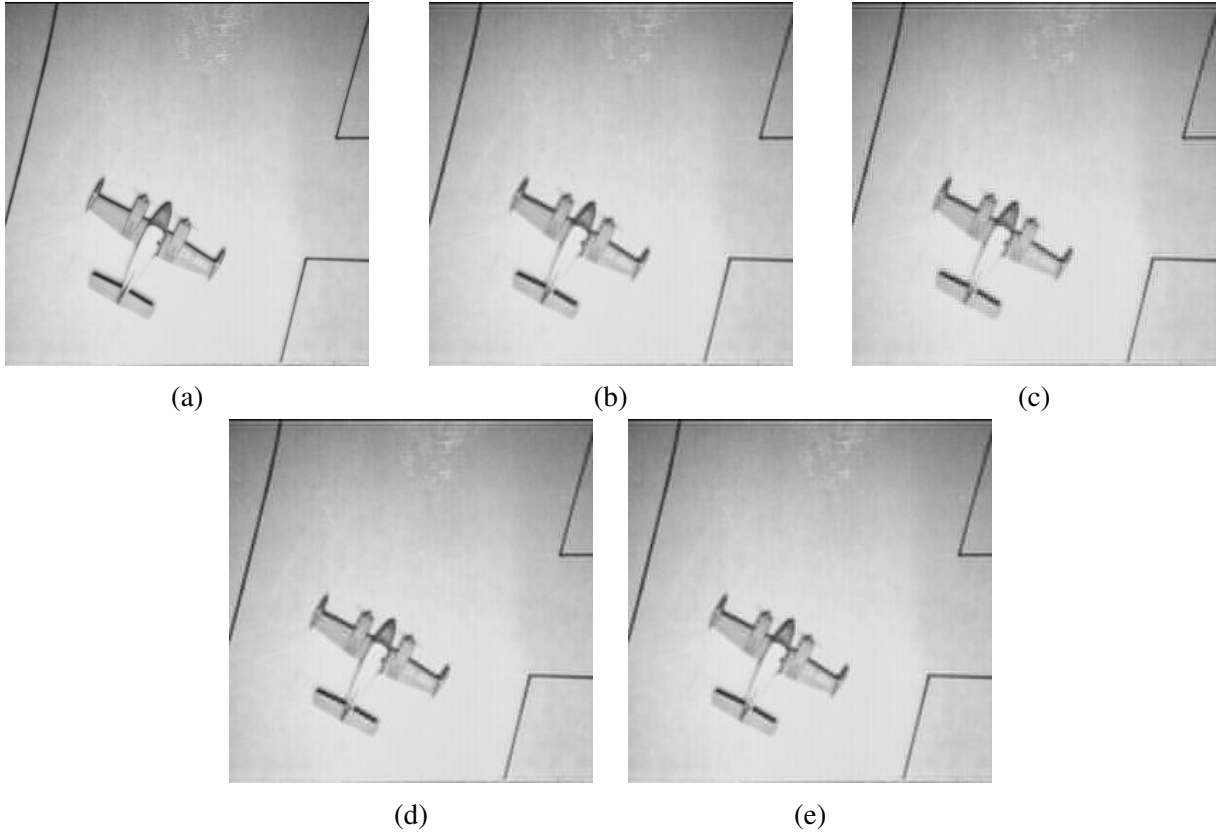
Fonte: O autor (2026).

Figura 24 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.1.09: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$.



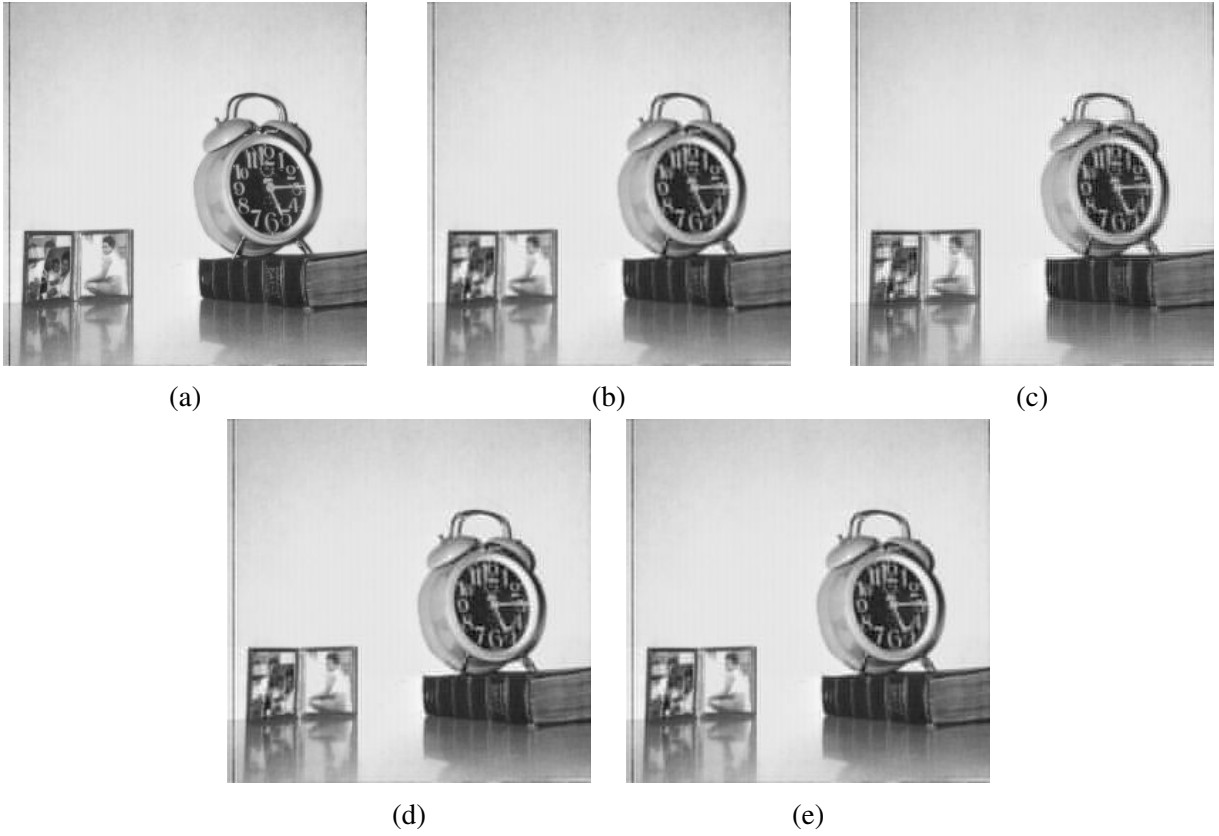
Fonte: O autor (2026).

Figura 25 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.1.11: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_8^{(1)}$; (d) aproximação $\hat{\mathbf{C}}_8^{(2)}$ e (e) aproximação $\hat{\mathbf{C}}_8^{(4)}$.



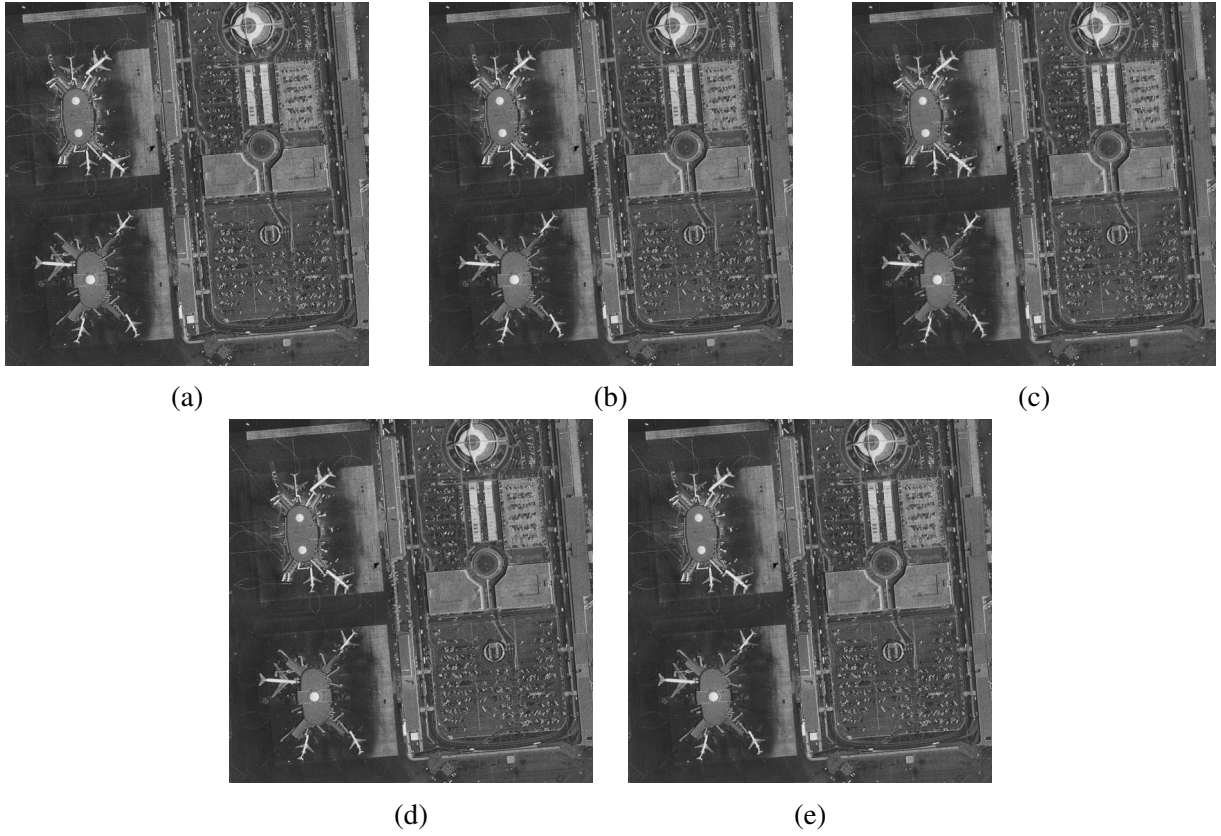
Fonte: O autor (2026).

Figura 26 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.1.12: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$.



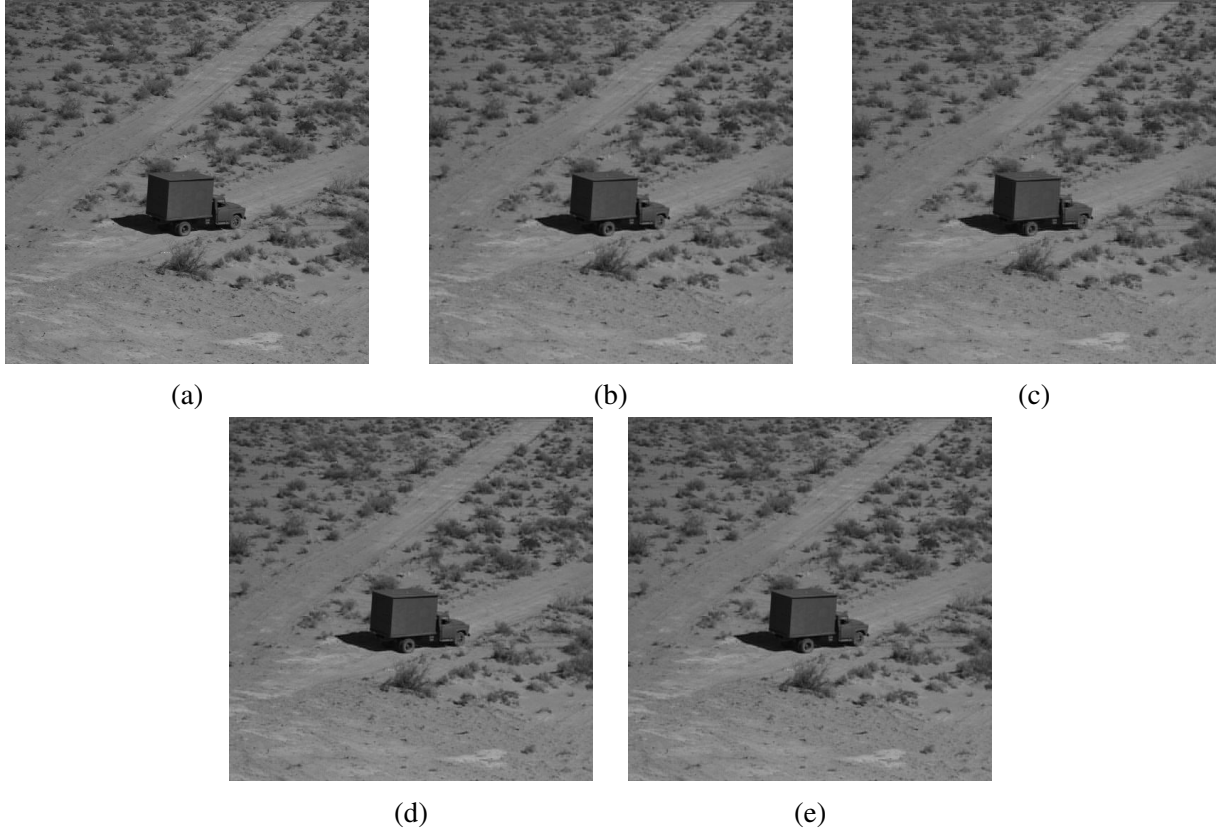
Fonte: O autor (2026).

Figura 27 – Comparação da compressão de imagens ($N = 8$) para a imagem 5.3.02: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{C}_8^{(1)}$; (d) aproximação $\hat{C}_8^{(2)}$ e (e) aproximação $\hat{C}_8^{(4)}$.



Fonte: O autor (2026).

Figura 28 – Comparação da compressão de imagens ($N = 8$) para a imagem 7.1.01: (a) imagem original e imagens reconstruídas pela (b) DCT exata; (c) aproximação $\hat{\mathbf{C}}_8^{(1)}$; (d) aproximação $\hat{\mathbf{C}}_8^{(2)}$ e (e) aproximação $\hat{\mathbf{C}}_8^{(4)}$.



Fonte: O autor (2026).

Tabela 8 – Métricas de qualidade da compressão de imagens para ($N = 8$, $r = 16$).

Imagem	Transformada	PSNR (dB)	SSIM	MSE	UQI
1.2.09	DCT	22.776	0.9147	343.147	0.9679
	$\hat{\mathbf{C}}_8^{(1)}$	21.042	0.8809	511.510	0.9512
	$\hat{\mathbf{C}}_8^{(2)}$	22.421	0.9091	372.333	0.9653
	$\hat{\mathbf{C}}_8^{(4)}$	22.616	0.9121	356.015	0.9667
1.3.09	DCT	35.482	0.9104	18.405	0.9830
	$\hat{\mathbf{C}}_8^{(1)}$	32.879	0.8756	33.508	0.9686
	$\hat{\mathbf{C}}_8^{(2)}$	35.205	0.9064	19.613	0.9819
	$\hat{\mathbf{C}}_8^{(4)}$	35.082	0.9070	20.176	0.9813
1.4.09	DCT	28.165	0.7710	99.220	0.9716
	$\hat{\mathbf{C}}_8^{(1)}$	26.741	0.7379	137.706	0.9601
	$\hat{\mathbf{C}}_8^{(2)}$	28.078	0.7662	101.221	0.9711
	$\hat{\mathbf{C}}_8^{(4)}$	28.073	0.7679	101.333	0.9710
3.2.25	DCT	29.739	0.8261	69.046	0.9501
	$\hat{\mathbf{C}}_8^{(1)}$	28.967	0.7992	82.486	0.9398
	$\hat{\mathbf{C}}_8^{(2)}$	29.552	0.8205	72.091	0.9481
	$\hat{\mathbf{C}}_8^{(4)}$	29.651	0.8236	70.470	0.9490
5.1.09	DCT	31.953	0.8091	41.474	0.9723
	$\hat{\mathbf{C}}_8^{(1)}$	31.193	0.7919	49.405	0.9668
	$\hat{\mathbf{C}}_8^{(2)}$	31.806	0.8049	42.901	0.9714
	$\hat{\mathbf{C}}_8^{(4)}$	31.863	0.8071	42.341	0.9717
5.1.11	DCT	33.827	0.9480	26.939	0.9875
	$\hat{\mathbf{C}}_8^{(1)}$	31.087	0.9235	50.624	0.9763
	$\hat{\mathbf{C}}_8^{(2)}$	33.706	0.9442	27.700	0.9872
	$\hat{\mathbf{C}}_8^{(4)}$	33.594	0.9477	28.427	0.9868
5.1.12	DCT	31.147	0.9497	49.927	0.9923
	$\hat{\mathbf{C}}_8^{(1)}$	29.138	0.9244	79.296	0.9877
	$\hat{\mathbf{C}}_8^{(2)}$	30.837	0.9449	53.627	0.9918
	$\hat{\mathbf{C}}_8^{(4)}$	30.925	0.9479	52.549	0.9919
5.3.02	DCT	29.927	0.8420	66.120	0.9730
	$\hat{\mathbf{C}}_8^{(1)}$	28.622	0.8147	89.313	0.9632
	$\hat{\mathbf{C}}_8^{(2)}$	29.693	0.8366	69.786	0.9716
	$\hat{\mathbf{C}}_8^{(4)}$	29.803	0.8400	68.047	0.9722
7.1.01	DCT	34.037	0.9023	25.667	0.9822
	$\hat{\mathbf{C}}_8^{(1)}$	32.645	0.8741	35.365	0.9753
	$\hat{\mathbf{C}}_8^{(2)}$	33.749	0.8972	27.424	0.9810
	$\hat{\mathbf{C}}_8^{(4)}$	33.868	0.8997	26.683	0.9815

Fonte: O autor (2026).

8.3.3 Análise dos resultados

As imagens reconstruídas a partir do experimento de compressão são perceptualmente indistinguíveis das originais sob inspeção visual, reproduzindo comportamento competitivo ao reportado na literatura para aproximações de baixa complexidade computacional (Oliveira *et al.*, 2018; Tablada *et al.*, 2014). As matrizes aproximadas propostas oferecem desempenho competitivo à DCT exata, com destaque para $\hat{\mathbf{C}}_8^{(2)}$ e $\hat{\mathbf{C}}_8^{(4)}$. Conforme a justificativa apresentada na Seção 2.3, isso indica que o rastreamento visual pode ser executado com qualidade equivalente à da DCT exata, porém com custo computacional inferior. Portanto, o resultado qualitativo é uma evidência da adequação das aproximações propostas para aplicações de compressão de imagens em sistemas de rastreamento em linha de produção.

9 CONCLUSÃO

Este trabalho investigou a construção de aproximações de baixa complexidade para a DCT de comprimentos $N = 4$ e $N = 8$, para aplicação em sistemas de rastreamento visual em linhas de produção industrial. A abordagem proposta baseou-se em duas etapas: (i) obtenção de aproximações da DFT de comprimentos $4N$ por meio de um problema de minimização da diferença angular entre os elementos da matriz exata e candidatos restritos a conjuntos de inteiros gaussianos $\mathcal{P}'_p$ e (ii) conversão matricial de $\hat{\mathbf{F}}_{4N}$ para $\hat{\mathbf{C}}_N$, derivada da adaptação do Teorema 4 (Blahut, 2010) à forma matricial.

Do ponto de vista metodológico, o trabalho apresenta três contribuições principais. Primeiro, foi proposto um método de minimização entre argumentos complexos como um problema de otimização não-linear sobre espaços de busca $\mathcal{P}'_p$ para obtenção de aproximações da DFT. Segundo, foi implementado um procedimento computacional de conversão entre matrizes aproximadas da DFT e da DCT. Terceiro, proposição de cinco aproximações da DCT livres de multiplicação — $\hat{\mathbf{C}}_4^{(1)}$, $\hat{\mathbf{C}}_4^{(2)}$, $\hat{\mathbf{C}}_8^{(1)}$, $\hat{\mathbf{C}}_8^{(2)}$ e $\hat{\mathbf{C}}_8^{(4)}$ — com fatorações por matrizes esparsas.

A avaliação das aproximações obtidas, segundo as figuras de mérito descritas na Seção 5.5, evidenciou desempenho competitivo da aproximação $\hat{\mathbf{C}}_8^{(1)}$ e desempenho superior das aproximações $\hat{\mathbf{C}}_8^{(2)}$ e $\hat{\mathbf{C}}_8^{(4)}$ em relação aos métodos em estado da arte, ao melhor do conhecimento do autor. O experimento de compressão padrão JPEG com retenção de 25% dos coeficientes corroborou qualitativamente os resultados numéricos: as imagens reconstruídas pelas matrizes propostas são perceptualmente indistinguíveis das originais. Tal resultado reforça a adequação das aproximações para cenários de processamento de imagens em dispositivos com restrições energéticas.

Do ponto de vista aplicado, as aproximações propostas são candidatas à substituição direta do núcleo espectral de métodos de rastreamento visual, como o descrito por Li *et al.* (2013), conforme indicado no fluxo metodológico da Figura 4. Ao viabilizar o processamento de imagens em dispositivos embarcados, o método contribui para a redução do custo operacional do monitoramento da produção em tempo real, fortalecendo os objetivos clássicos de desempenho discutidos no Capítulo 1.

Este trabalho de conclusão de curso é o desdobramento do trabalho de Iniciação Científica intitulado “*Algoritmos eficientes para processamento de imagens em linha de produção*”, desenvolvido no *Industrial Signal Processing Lab* (ISPL), do Centro Acadêmico do Agreste no período 2024–2025 sob orientação do Prof. Dr. R. J. Cintra.

9.1 PERSPECTIVAS DE TRABALHOS FUTUROS

A continuação deste trabalho inclui os seguintes desdobramentos:

- Estender o método para dimensões $N = 16, 32, 64$, para aplicação em padrões de codifi-

cação de vídeo, como o HEVC;

- Investigar funções-objeto alternativas que produzam aproximações ortogonais com desempenho competitivo;
- Submeter os resultados estendidos a periódico.

REFERÊNCIAS

- AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transform. **IEEE Transactions on Computers**, C-23, n. 1, p. 90–93, 1974.
- AHMED, N.; RAO, K. R. **Orthogonal Transforms for Digital Signal Processing**. New York: Springer-Verlag, 1975.
- BARNES, R. M. **Motion and Time Study: Design and Measurement of Work**. 7. ed. New York: John Wiley & Sons, 1980.
- BAYER, F. M.; CINTRA, R. J. Image compression via a fast dct approximation. **IEEE Latin America Transactions**, v. 8, n. 6, p. 708–713, 2010.
- BEYERER, J.; LEÓN, F. P.; FRESE, C. **Machine Vision: Automated Visual Inspection: Theory, Practice and Applications**. Berlin: Springer, 2016. ISBN 978-3-662-47794-6.
- BLAHUT, R. E. **Fast Algorithms for Signal Processing**. 2nd. ed. [S.l.]: Cambridge University Press, 2010.
- BOUGUEZEL, S.; AHMAD, M.; SWAMY, M. Low-complexity 8×8 transform for image compression. **Electron Lett**, Institution of Engineering and Technology (IET), v. 44, n. 21, p. 1249–1250, 2008.
- BRACEWELL, R. N. **The Hartley Transform**. New York: Oxford University Press, 1986.
- BRACEWELL, R. N. **The Fourier Transform and Its Applications**. 3rd. ed. New York, NY: McGraw-Hill, 2000.
- BRAILSFORD, S.; CHURILOV, L.; DANGERFIELD, B. **Discrete-Event Simulation and System Dynamics for Management Decision Making**. [S.l.]: Wiley, 2014. Eds.
- BRIGGS, W. L.; HENSON, V. E. **The DFT: An Owner's Manual for the Discrete Fourier Transform**. Philadelphia: SIAM, 1995.
- BRIGHAM, E. O. **The Fast Fourier Transform and Its Applications**. 1st. ed. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- BRITANAK, V.; YIP, P. C.; RAO, K. R. **Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations**. [S.l.]: Academic Press, 2007. ISBN 978-0123736246.
- CINTRA, R. J.; BAYER, F. M. A DCT approximation for image compression. **IEEE Signal Processing Letters**, v. 18, n. 10, p. 579–582, 2011.
- CINTRA, R. J.; BAYER, F. M. DCT-like transform for image compression requires 14 additions only. **Electronics Letters**, v. 47, n. 8, p. 483–484, 2011.
- CLARKE, R. J. Relation between the Karhunen-Loève and cosine transforms. **IEE Proceedings F - Communications, Radar and Signal Processing**, v. 128, n. 6, p. 359–360, 1981.
- COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. **Mathematics of Computation**, v. 19, n. 90, p. 297–301, 1965.

GALL, D. L. MPEG: A video compression standard for multimedia applications. **Communications of the ACM**, ACM, v. 34, n. 4, p. 46–58, 1991.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 4th. ed. New York: Pearson Education, 2018. ISBN 978-0133356724.

GOYAL, V. K. Theoretical foundations of transform coding. **IEEE Signal Processing Magazine**, v. 18, n. 5, p. 9–21, 2001.

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

HE, Q. P. *et al.* Statistical process monitoring for iot-enabled cybermanufacturing: Opportunities and challenges. In: **IFAC-PapersOnLine**. [S.l.: s.n.], 2017. v. 50, n. 1, p. 14946–14951.

HEIDEMAN, M. T. **Multiplicative Complexity, Convolution, and the DFT**. Tese (Doutorado) — Rice University, Houston, TX, 1988.

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

HIGHAM, N. J. Computing real square roots of a real matrix. **Linear Algebra and its Applications**, v. 88–89, p. 405–430, 1987.

HOTELLING, H. Analysis of a complex of statistical variables into principal components. **Journal of Educational Psychology**, v. 24, n. 6, p. 417–441, 1933.

HUYNH-THU, Q.; GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. **Electronics Letters**, v. 44, n. 13, p. 800–801, 2008.

ISARD, M.; BLAKE, A. Condensation—conditional density propagation for visual tracking. **International Journal of Computer Vision**, Springer, v. 29, n. 1, p. 5–28, 1998.

JAIN, A. K. **Fundamentals of Digital Image Processing**. Englewood Cliffs, NJ: Prentice Hall, 1989.

JOHANESA, T. V. A.; EQUETER, L.; MAHMOUDI, S. A. Survey on AI applications for product quality control and predictive maintenance in industry 4.0. **Electronics**, MDPI, v. 13, n. 5, p. 976, 2024.

KAGERMANN, H.; WAHLSTER, W.; HELBIG, J. Recommendations for implementing the strategic initiative INDUSTRIE 4.0. **Final Report of the Industrie 4.0 Working Group**, aca-tech – National Academy of Science and Engineering, p. 1–82, 2013.

KALSOOM, T. *et al.* Advances in sensor technologies in the era of smart factory and industry 4.0. **Sensors**, MDPI, v. 20, n. 23, p. 6783, 2020.

KATTO, J.; YASUDA, Y. Performance evaluation of subband coding and optimization of its filter coefficients. **Journal of Visual Communication and Image Representation**, v. 2, n. 4, p. 303–313, 1991.

KAY, S. M. **Fundamentals of Statistical Signal Processing: Estimation Theory**. Upper Saddle River, NJ: Prentice Hall, 2012. v. 1.

KUSIAK, A. Smart manufacturing. **International Journal of Production Research**, Taylor & Francis, v. 56, n. 1-2, p. 508–517, 2018.

LECOMPTE, J.; LEGOFF, O.; HASCOET, J.-Y. Technological form defects identification using discrete cosine transform method. **The International Journal of Advanced Manufacturing Technology**, Springer, v. 51, n. 9-12, p. 1033–1044, 2010.

LENGWEHASATIT, K.; ORTEGA, A. Scalable variable complexity approximate forward DCT. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 14, n. 11, p. 1236–1248, 2004.

LEVITIN, A. **Introduction to the Design and Analysis of Algorithms**. 3rd. ed. Boston, MA: Pearson, 2012.

LI, X. *et al.* Incremental learning of 3D-DCT compact representations for robust visual tracking. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 35, n. 4, p. 863–881, 2013.

LOEFFLER, C.; LIGTENBERG, A.; MOSCHYTZ, G. S. Practical fast 1-D DCT algorithms with 11 multiplications. p. 988–991, 1989.

MADANAYAKE, A. *et al.* Low-power VLSI architectures for DCT/DWT: Precision vs approximation for HD video, biomedical, and smart antenna applications. **IEEE Circuits and Systems Magazine**, IEEE, v. 15, n. 1, p. 25–47, February 2015.

MALAMAS, E. N. *et al.* A survey on industrial vision systems, applications and tools. **Image and Vision Computing**, Elsevier, v. 21, n. 2, p. 171–188, 2003.

MARVASTI-ZADEH, S. M. *et al.* Deep learning for visual tracking: A comprehensive survey. **IEEE Transactions on Intelligent Transportation Systems**, v. 23, n. 5, p. 3943–3968, 2022.

MONOSTORI, L. *et al.* Cyber-physical systems in manufacturing. **CIRP Annals**, Elsevier, v. 65, n. 2, p. 621–641, 2016.

MONTGOMERY, D. C. **Statistical Quality Control: A Modern Introduction**. 7. ed. Hoboken, NJ: John Wiley & Sons, 2012.

OAKLAND, J. S. **Total Quality Management and Operational Excellence: Text with Cases**. 4. ed. London: Routledge, 2014.

OLIVEIRA, R. S. *et al.* Low-complexity 8-point DCT approximation based on angle similarity for image and video coding. **Multidimensional Systems and Signal Processing**, v. 30, p. 1363–1394, 2018.

OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-Time Signal Processing**. 3rd. ed. [S.l.]: Pearson, 2010.

OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-Time Signal Processing**. 3rd. ed. Upper Saddle River, NJ: Pearson, 2014. ISBN 978-0131988422.

PARHI, K. K. **VLSI Digital Signal Processing Systems**. [S.l.]: Wiley, 1999.

PEARSON, K. On lines and planes of closest fit to systems of points in space. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, v. 2, n. 11, p. 559–572, 1901.

PERNG, D.-B.; CHEN, S.-H. Directional textures auto-inspection using discrete cosine transform. **International Journal of Production Research**, Taylor & Francis, v. 49, n. 23, p. 7171–7187, 2011.

PORTELLA, L.; BAYER, F. M.; CINTRA, R. J. Multiplierless DFT approximation based on the prime factor algorithm. **IEEE Transactions on Signal Processing**, p. 5273–5285, 2025.

RAO, K. R.; YIP, P. **Discrete Cosine Transform: Algorithms, Advantages, Applications**. Boston, MA: Academic Press, 1990. ISBN 978-0125802031.

RAO, K. R.; YIP, P. C. **The Transform and Data Compression Handbook**. 1st. ed. Boca Raton, FL: CRC Press, 2018.

RAY, W.; DRIVER, R. Further decomposition of the Karhunen-Loève series representation of a stationary random process. **IEEE Trans Inf Theory**, Institute of Electrical and Electronics Engineers (IEEE), v. 16, n. 6, p. 663–668, nov. 1970. ISSN 1557-9654.

RICHARDSON, I. E. **H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia**. Chichester: John Wiley & Sons, 2003.

SAYOOD, K. **Introduction to Data Compression**. 5. ed. Cambridge, MA: Morgan Kaufmann, 2017.

SEBER, G. A. F. **A matrix handbook for statisticians**. Hoboken, N.J.: Wiley-Interscience, 2008. (Wiley series in probability and statistics). Includes bibliographical references (p. 529–545) and index. ISBN 9780470226780.

SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Operations Management**. 6. ed. Harlow, UK: Pearson Education, 2010.

SMITH, J. O. **Mathematics of the Discrete Fourier Transform (DFT): With Audio Applications**. 2nd. ed. Stanford, CA: W3K Publishing, 2002. Disponível em: <<https://ccrma.stanford.edu/~jos/mdft/>>.

STRANG, G. **Linear Algebra and Its Applications**. 2nd. ed. New York, NY: Academic Press, 1980.

SULLIVAN, G. J. *et al.* Overview of the high efficiency video coding (HEVC) standard. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 22, n. 12, p. 1649–1668, 2012.

TABLADA, C. *et al.* Low-complexity DCT approximations for image and video coding. **Signal Processing: Image Communication**, v. 29, n. 10, p. 1127–1147, 2014.

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

THOBEN, K.-D.; WIESNER, S.; WUEST, T. “Industrie 4.0” and smart manufacturing – a review of research issues and application examples. **International Journal of Automation Technology**, v. 11, n. 1, p. 4–16, 2017.

USC-SIPI. **The USC-SIPI Image Database**. 2024. <<https://sipi.usc.edu/database/>>. Acesso em: 15 jan. 2025.

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

WANG, Z.; BOVIK, A. C. A universal image quality index. **IEEE Signal Processing Letters**, v. 9, n. 3, p. 81–84, 2002.

WANG, Z.; BOVIK, A. C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. **IEEE Signal Processing Magazine**, v. 26, n. 1, p. 98–117, 2009.

WANG, Z. *et al.* Image quality assessment: From error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600–612, 2004.

WEESE, M. L. *et al.* Statistical learning methods applied to process monitoring: An overview and perspective. **Journal of Quality Technology**, Taylor & Francis, v. 48, n. 1, p. 4–27, 2016.

WIEGAND, T. *et al.* Overview of the H.264/AVC video coding standard. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 13, n. 7, p. 560–576, 2003.

WOODALL, W. H. Controversies and contradictions in statistical process control. **Journal of Quality Technology**, Taylor & Francis, v. 32, n. 4, p. 341–350, 2000.

YILMAZ, A.; JAVED, O.; SHAH, M. Object tracking: A survey. **ACM Computing Surveys**, v. 38, n. 4, p. 1–45, 2006.

ZHONG, R. Y. *et al.* Intelligent manufacturing in the context of industry 4.0: A review. **Engineering**, Elsevier, v. 3, n. 5, p. 616–630, 2017.

APÊNDICE A – APROXIMAÇÕES DA DFT

Neste apêndice são apresentadas explicitamente as aproximações da DFT obtidas por meio do método descrito na Seção 6.2. As matrizes aqui listadas correspondem às soluções encontradas pelo algoritmo de otimização proposto, considerando os espaços de busca $\mathcal{P}'_p$ com $p = 1, 2, 4$. Desse modo, a notação adotada para as matrizes aproximadas é $\widehat{\mathbf{F}}_N^{(p)}$, em que N indica a dimensão da transformada e o sobrescrito (p) identifica o conjunto de soluções ao qual a matriz pertence.

APROXIMAÇÕES DA DFT DE TAMANHO $N = 16$

Aproximação $\widehat{\mathbf{F}}_{16}^{(1)}$

$$\widehat{\mathbf{F}}_{16}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1-j & 1-j & 1-j & -1-j & -1-j & -1-j & -1-j & -1 & -1+j & -1+j & -1+j & -1+j & 1j & 1+j & 1+j & 1+j \\ 1 & 1-j & -1j & -1-j & -1 & -1+j & 1j & 1+j & 1 & 1-j & -1j & -1-j & -1 & -1+j & 1j & 1+j \\ 1 & 1-j & -1-j & -1+j & 1j & 1+j & 1-j & -1-j & -1 & -1+j & 1+j & 1-j & -1j & -1-j & -1+j & 1+j \\ 1 & -1j & -1 & 1j & 1 & -1j & -1 & 1j & 1 & -1j & -1 & 1j & 1 & -1j & -1 & 1j \\ 1 & -1-j & -1+j & 1+j & -1j & -1+j & 1+j & 1-j & -1 & 1+j & 1-j & -1-j & 1j & 1-j & -1-j & -1+j \\ 1 & -1-j & 1j & 1-j & -1 & 1+j & -1j & -1+j & 1 & -1-j & 1j & 1-j & -1 & 1+j & -1j & -1+j \\ 1 & -1-j & 1+j & -1-j & 1j & 1-j & -1+j & 1-j & -1 & 1+j & -1-j & 1+j & -1j & -1+j & 1-j & -1+j \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -1+j & 1-j & -1+j & -1j & 1+j & -1-j & 1+j & -1 & 1-j & -1+j & 1-j & 1j & -1-j & 1+j & -1-j \\ 1 & -1+j & -1j & 1+j & -1 & 1-j & 1j & -1-j & 1 & -1+j & -1j & 1+j & -1 & 1-j & 1j & -1-j \\ 1 & -1+j & -1-j & 1-j & 1j & -1-j & 1-j & 1+j & -1 & 1-j & 1+j & -1+j & -1j & 1+j & -1+j & -1-j \\ 1 & 1j & -1 & -1j & 1 & 1j & -1 & -1j & 1 & 1j & -1 & -1j & 1 & 1j & -1 & -1j \\ 1 & 1+j & -1+j & -1-j & -1j & 1-j & 1+j & -1+j & -1 & -1-j & 1-j & 1+j & 1j & -1+j & -1-j & 1-j \\ 1 & 1+j & 1j & -1+j & -1 & -1-j & -1j & 1-j & 1 & 1+j & 1j & -1+j & -1 & -1-j & -1j & 1-j \\ 1 & 1+j & 1+j & 1+j & 1j & -1+j & -1+j & -1+j & -1 & -1-j & -1-j & -1-j & -1j & 1-j & 1-j & 1-j \end{bmatrix}$$

Aproximação $\hat{\mathbf{F}}_{16}^{(2)}$

$$\hat{\mathbf{F}}_{16}^{(2)} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2-1j & 2-2j & 1-2j & -2j & -1-2j & -2-2j & -2-1j & -2 & -2+1j & -2+2j & -1+2j & 2j & 1+2j & 2+2j & 2+1j \\ 2 & 2-2j & -2j & -2-2j & -2 & -2+2j & 2j & 2+2j & 2 & 2-2j & -2j & -2-2j & -2 & -2+2j & 2j & 2+2j \\ 2 & 1-2j & -2-2j & -2+1j & 2j & 2+1j & 2-2j & -1-2j & -2 & -1+2j & 2+2j & 2-1j & -2j & -2-1j & -2+2j & 1+2j \\ 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j \\ 2 & -1-2j & -2+2j & 2+1j & -2j & -2+1j & 2+2j & 1-2j & -2 & 1+2j & 2-2j & -2-1j & 2j & 2-1j & -2-2j & -1+2j \\ 2 & -2-2j & 2j & 2-2j & -2 & 2+2j & -2j & -2+2j & 2 & -2-2j & 2j & 2-2j & -2 & 2+2j & -2j & -2+2j \\ 2 & -2-1j & 2+2j & -1-2j & 2j & 1-2j & -2+2j & 2-1j & -2 & 2+1j & -2-2j & 1+2j & -2j & -1+2j & 2-2j & -2+1j \\ 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 \\ 2 & -2+1j & 2-2j & -1+2j & -2j & 1+2j & -2-2j & 2+1j & -2 & 2-1j & -2+2j & 1-2j & 2j & -1-2j & 2+2j & -2-1j \\ 2 & -2+2j & -2j & 2+2j & -2 & 2-2j & 2j & -2-2j & 2 & -2+2j & -2j & 2+2j & -2 & 2-2j & 2j & -2-2j \\ 2 & -1+2j & -2-2j & 2-1j & 2j & 2-1j & 2-2j & 1+2j & -2 & 1-2j & 2+2j & -2+1j & -2j & 2+1j & -2+2j & -1-2j \\ 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j \\ 2 & 1+2j & -2+2j & -2-1j & -2j & 2-1j & 2+2j & -1+2j & -2 & -1-2j & 2-2j & 2+1j & 2j & -2+1j & -2-2j & 1-2j \\ 2 & 2+2j & 2j & -2+2j & -2 & -2-2j & -2j & 2-2j & 2 & 2+2j & 2j & -2+2j & -2 & -2-2j & -2j & 2-2j \\ 2 & 2+1j & 2+2j & 1+2j & 2j & -1+2j & -2+2j & -2+1j & -2 & -2-1j & -2-2j & -1-2j & -2j & 1-2j & 2-2j & 2-1j \end{bmatrix}$$

APROXIMAÇÕES DA DFT DE TAMANHO $N = 32$ **Aproximação $\hat{\mathbf{F}}_{32}^{(1)}$**

$$\hat{\mathbf{F}}_{32}^{(1)} = \left[\begin{array}{c|c} \mathbf{P}_{11}^{(1)} & \mathbf{P}_{12}^{(1)} \\ \hline \mathbf{P}_{21}^{(1)} & \mathbf{P}_{22}^{(1)} \end{array} \right], \quad (\text{A.1})$$

em que

$$\mathbf{P}_{11}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1-j & 1-j & 1-j & 1-j & 1-j & -j & -j & -j & -1-j & -1-j & -1-j & -1-j & -1-j & -1 \\ 1 & 1-j & 1-j & 1-j & -j & -1-j & -1-j & -1-j & -1 & -1+j & -1+j & -1+j & j & 1+j & 1+j & 1+j \\ 1 & 1-j & 1-j & -j & -1-j & -1 & -1+j & -1+j & j & 1+j & 1+j & 1 & 1-j & -j & -1-j & -1-j \\ 1 & 1-j & -j & -1-j & -1 & -1+j & j & 1+j & 1 & 1-j & -j & -1-j & -1 & -1+j & j & 1+j \\ 1 & 1-j & -1-j & -1 & -1+j & j & 1+j & 1-j & -j & -1-j & -1+j & j & 1+j & 1 & 1-j & -1-j \\ 1 & 1-j & -1-j & -1+j & j & 1+j & 1-j & -1-j & -1 & -1+j & 1+j & 1-j & -j & -1-j & -1+j & 1+j \\ 1 & -j & -1-j & -1+j & 1+j & 1-j & -1-j & -1 & j & 1 & 1-j & -1-j & -1+j & 1+j & 1-j & -j \\ 1 & -j & -1 & j & 1 & -j & -1 & j & 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & -j & -1+j & 1+j & 1-j & -1-j & -1+j & 1 & -j & -1 & 1+j & 1-j & -1-j & -1+j & 1+j & -j \\ 1 & -1-j & -1+j & 1+j & -j & -1+j & 1+j & 1-j & -1 & 1+j & 1-j & -1-j & j & 1-j & -1-j & -1+j \\ 1 & -1-j & -1+j & 1 & -1-j & j & 1-j & -1-j & j & 1-j & -1-j & j & 1-j & -1 & 1+j & 1-j \\ 1 & -1-j & j & 1-j & -1 & 1+j & -j & -1+j & 1 & -1-j & j & 1-j & -1 & 1+j & -j & -1+j \\ 1 & -1-j & 1+j & -j & -1+j & 1 & -1-j & 1+j & -j & -1+j & 1-j & -1 & 1+j & -j & -1+j & 1-j \\ 1 & -1-j & 1+j & -1-j & j & 1-j & -1+j & 1-j & -1 & 1+j & -1-j & 1+j & -j & -1+j & 1-j & -1+j \\ 1 & -1 & 1+j & -1-j & 1+j & -1-j & 1+j & -j & j & -j & -1+j & 1-j & -1+j & 1-j & -1+j & 1 \end{bmatrix}$$

Aproximação $\hat{\mathbf{F}}_{32}^{(2)}$

$$\hat{\mathbf{F}}_{32}^{(2)} = \left[\begin{array}{c|c} \mathbf{P}_{11}^{(2)} & \mathbf{P}_{12}^{(2)} \\ \hline \mathbf{P}_{21}^{(2)} & \mathbf{P}_{22}^{(2)} \end{array} \right], \quad (\text{A.2})$$

em que

$$\mathbf{P}_{11}^{(2)} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2-j & 2-j & 2-2j & 1-2j & 1-2j & -2j & -2j & -2j & -1-2j & -1-2j & -2-2j & -2-j & -2-j & -2 \\ 2 & 2-j & 2-2j & 1-2j & -2j & -1-2j & -2-2j & -2-j & -2 & -2+j & -2+2j & -1+2j & 2j & 1+2j & 2+2j & 2+j \\ 2 & 2-j & 1-2j & -2j & -2-2j & -2 & -2+j & -1+2j & 2j & 1+2j & 2+j & 2 & 2-2j & -2j & -1-2j & -2-j \\ 2 & 2-2j & -2j & -2-2j & -2 & -2+2j & 2j & 2+2j & 2 & 2-2j & -2j & -2-2j & -2 & -2+2j & 2j & 2+2j \\ 2 & 1-2j & -1-2j & -2 & -2+2j & 2j & 2+j & 2-j & -2j & -2-j & -2+j & 2j & 2+2j & 2 & 1-2j & -1-2j \\ 2 & 1-2j & -2-2j & -2+j & 2j & 2+j & 2-2j & -1-2j & -2 & -1+2j & 2+2j & 2-j & -2j & -2-j & -2+2j & 1+2j \\ 2 & -2j & -2-j & -1+2j & 2+2j & 2-j & -1-2j & -2 & 2j & 2 & 1-2j & -2-j & -2+2j & 1+2j & 2-j & -2j \\ 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j \\ 2 & -2j & -2+j & 1+2j & 2-2j & -2-j & -1+2j & 2 & -2j & -2 & 1+2j & 2-j & -2-2j & -1+2j & 2+j & -2j \\ 2 & -1-2j & -2+2j & 2+j & -2j & -2+j & 2+2j & 1-2j & -2 & 1+2j & 2-2j & -2-j & 2j & 2-j & -2-2j & -1+2j \\ 2 & -1-2j & -1+2j & 2 & -2-2j & 2j & 2-j & -2-j & 2j & 2-j & -2-j & 2j & 2-2j & -2 & 1+2j & 1-2j \\ 2 & -2-2j & 2j & 2-2j & -2 & 2+2j & -2j & -2+2j & 2 & -2-2j & 2j & 2-2j & -2 & 2+2j & -2j & -2+2j \\ 2 & -2-j & 1+2j & -2j & -2+2j & 2 & -2-j & 1+2j & -2j & -1+2j & 2-j & -2 & 2+2j & -2j & -1+2j & 2-j \\ 2 & -2-j & 2+2j & -1-2j & 2j & 1-2j & -2+2j & 2-j & -2 & 2+j & -2-2j & 1+2j & -2j & -1+2j & 2-2j & -2+j \\ 2 & -2 & 2+j & -2-j & 2+2j & -1-2j & 1+2j & -2j & 2j & -2j & -1+2j & 1-2j & -2+2j & 2-j & -2+j & 2 \end{bmatrix}$$

$$\mathbf{P}_{12}^{(2)} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ -2 & -2 & -2+j & -2+j & -2+2j & -1+2j & -1+2j & 2j & 2j & 2j & 1+2j & 1+2j & 2+2j & 2+j & 2+j & 2 \\ 2 & 2-j & 2-2j & 1-2j & -2j & -1-2j & -2-2j & -2-j & -2 & -2+j & -2+2j & -1+2j & 2j & 1+2j & 2+2j & 2+j \\ -2 & -2+j & -1+2j & 2j & 2+2j & 2 & 2-j & 1-2j & -2j & -1-2j & -2-j & -2 & -2+2j & 2j & 1+2j & 2+j \\ 2 & 2-2j & -2j & -2-2j & -2 & -2+2j & 2j & 2+2j & 2 & 2-2j & -2j & -2-2j & -2 & -2+2j & 2j & 2+2j \\ -2 & -1+2j & 1+2j & 2 & 2-2j & -2j & -2-j & -2+j & 2j & 2+j & 2-j & -2j & -2-2j & -2 & -1+2j & 1+2j \\ 2 & 1-2j & -2-2j & -2+j & 2j & 2+j & 2-2j & -1-2j & -2 & -1+2j & 2+2j & 2-j & -2j & -2-j & -2+2j & 1+2j \\ -2 & 2j & 2+j & 1-2j & -2-2j & -2+j & 1+2j & 2 & -2j & -2 & -1+2j & 2+j & 2-2j & -1-2j & -2+j & 2j \\ 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j & 2 & -2j & -2 & 2j \\ -2 & 2j & 2-j & -1-2j & -2+2j & 2+j & 1-2j & -2 & 2j & 2 & -1-2j & -2+j & 2+2j & 1-2j & -2-j & 2j \\ 2 & -1-2j & -2+2j & 2+j & -2j & -2+j & 2+2j & 1-2j & -2 & 1+2j & 2-2j & -2-j & 2j & 2-j & -2-2j & -1+2j \\ -2 & 1+2j & 1-2j & -2 & 2+2j & -2j & -2+j & 2+j & -2j & -2+j & 2+j & -2j & -2+2j & 2 & -1-2j & -1+2j \\ 2 & -2-2j & 2j & 2-2j & -2 & 2+2j & -2j & -2+2j & 2 & -2-2j & 2j & 2-2j & -2 & 2+2j & -2j & -2+2j \\ -2 & 2+j & -1-2j & 2j & 2-2j & -2 & 2+j & -1-2j & 2j & 1-2j & -2+j & 2 & -2-2j & 2j & 1-2j & -2+j \\ 2 & -2-j & 2+2j & -1-2j & 2j & 1-2j & -2+2j & 2-j & -2 & 2+j & -2-2j & 1+2j & -2j & -1+2j & 2-2j & -2+j \\ -2 & 2 & -2-j & 2+j & -2-2j & 1+2j & -1-2j & 2j & -2j & 2j & 1-2j & -1+2j & 2-2j & -2+j & 2-j & -2 \end{bmatrix}$$

$$\mathbf{P}_{21}^{(2)} = \begin{bmatrix} 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 \\ 2 & -2 & 2-j & -2+j & 2-2j & -1+2j & 1-2j & 2j & -2j & 2j & -1-2j & 1+2j & -2-2j & 2+j & -2-j & 2 \\ 2 & -2+j & 2-2j & -1+2j & -2j & 1+2j & -2-2j & 2+j & -2 & 2-j & -2+2j & 1-2j & 2j & -1-2j & 2+2j & -2-j \\ 2 & -2+j & 1-2j & 2j & -2-2j & 2 & -2+j & 1-2j & 2j & -1-2j & 2+j & -2 & 2-2j & 2j & -1-2j & 2+j \\ 2 & -2+2j & -2j & 2+2j & -2 & 2-2j & 2j & -2-2j & 2 & -2+2j & -2j & 2+2j & -2 & 2-2j & 2j & -2-2j \\ 2 & -1+2j & -1-2j & 2 & -2+2j & -2j & 2+j & -2+j & -2j & 2+j & -2+j & -2j & 2+2j & -2 & 1-2j & 1+2j \\ 2 & -1+2j & -2-2j & 2-j & 2j & -2-j & 2-2j & 1+2j & -2 & 1-2j & 2+2j & -2+j & -2j & 2+j & -2+2j & -1-2j \\ 2 & 2j & -2-j & 1-2j & 2+2j & -2+j & -1-2j & 2 & 2j & -2 & 1-2j & 2+j & -2+2j & -1-2j & 2-j & 2j \\ 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j \\ 2 & 2j & -2+j & -1-2j & 2-2j & 2+j & -1+2j & -2 & -2j & 2 & 1+2j & -2+j & -2-2j & 1-2j & 2+j & 2j \\ 2 & 1+2j & -2+2j & -2-j & -2j & 2-j & 2+2j & -1+2j & -2 & -1-2j & 2-2j & 2+j & -2+j & -2-2j & 1-2j & 2-j \\ 2 & 1+2j & -1+2j & -2 & -2-2j & -2j & 2-j & 2+j & 2j & -2+j & -2-j & -2j & 2-2j & 2 & 1+2j & -1+2j \\ 2 & 2+2j & 2j & -2+2j & -2 & -2-2j & -2j & 2-2j & 2 & 2+2j & 2j & -2+2j & -2 & -2-2j & -2j & 2-2j \\ 2 & 2+j & 1+2j & 2j & -2+2j & -2 & -2-j & -1-2j & -2j & 1-2j & 2-j & 2 & 2+2j & 2j & -1+2j & -2+j \\ 2 & 2+j & 2+2j & 1+2j & 2j & -1+2j & -2+2j & -2+j & -2 & -2-j & -2-2j & -1-2j & -2j & 1-2j & 2-2j & 2-j \\ 2 & 2 & 2+j & 2+j & 2+2j & 1+2j & 1+2j & 2j & 2j & 2j & -1+2j & -1+2j & -2+2j & -2+j & -2+j & -2 \end{bmatrix}$$

$$\mathbf{P}_{22}^{(2)} = \begin{bmatrix} 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 & 2 & -2 \\ -2 & 2 & -2+j & 2-j & -2+2j & 1-2j & -1+2j & -2j & 2j & -2j & 1+2j & -1-2j & 2+2j & -2-j & 2+j & -2 \\ 2 & -2+j & 2-2j & -1+2j & -2j & 1+2j & -2-2j & 2+j & -2 & 2-j & -2+2j & 1-2j & 2j & -1-2j & 2+2j & -2-j \\ -2 & 2-j & -1+2j & -2j & 2+2j & -2 & 2-j & -1+2j & -2j & 1+2j & -2-j & 2 & -2+2j & -2j & 1+2j & -2-j \\ 2 & -2+2j & -2j & 2+2j & -2 & 2-2j & 2j & -2-2j & 2 & -2+2j & -2j & 2+2j & -2 & 2-2j & 2j & -2-2j \\ -2 & 1-2j & 1+2j & -2 & 2-2j & 2j & -2-j & 2-j & 2j & -2-j & 2-j & 2j & -2-2j & 2 & -1+2j & -1-2j \\ 2 & -1+2j & -2-2j & 2-j & 2j & -2-j & 2-2j & 1+2j & -2 & 1-2j & 2+2j & -2+j & -2j & 2+j & -2+2j & -1-2j \\ -2 & -2j & 2+j & -1+2j & -2-2j & 2-j & 1+2j & -2 & -2j & 2 & -1+2j & -2-j & 2-2j & 1+2j & -2+j & -2j \\ 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j & 2 & 2j & -2 & -2j \\ -2 & -2j & 2-j & 1+2j & -2+2j & -2-j & 1-2j & 2 & 2j & -2 & -1-2j & 2-j & 2+2j & -1+2j & -2-j & -2j \\ 2 & 1+2j & -2+2j & -2-j & -2j & 2-j & 2+2j & -1+2j & -2 & -1-2j & 2-2j & 2+j & 2j & -2+j & -2-2j & 1-2j \\ -2 & -1-2j & 1-2j & 2 & 2+2j & 2j & -2+j & -2-j & -2j & 2-j & 2+j & 2j & -2+2j & -2 & -1-2j & 1-2j \\ 2 & 2+2j & 2j & -2+2j & -2 & -2-2j & -2j & 2-2j & 2 & 2+2j & 2j & -2+2j & -2 & -2-2j & -2j & 2-2j \\ -2 & -2-j & -1-2j & -2j & 2-2j & 2 & 2+j & 1+2j & 2j & -1+2j & -2+j & -2 & -2-2j & -2j & 1-2j & 2-j \\ 2 & 2+j & 2+2j & 1+2j & 2j & -1+2j & -2+2j & -2+j & -2 & -2-j & -2-2j & -1-2j & -2j & 1-2j & 2-2j & 2-j \\ -2 & -2 & -2-j & -2-j & -2-2j & -1-2j & -1-2j & -2j & -2j & -2j & 1-2j & 1-2j & 2-2j & 2-j & 2-j & 2 \end{bmatrix}$$

Aproximação $\hat{\mathbf{F}}_{32}^{(4)}$

$$\hat{\mathbf{F}}_{32}^{(4)} = \begin{bmatrix} \mathbf{P}_{11}^{(4)} & \mathbf{P}_{12}^{(4)} \\ \mathbf{P}_{21}^{(4)} & \mathbf{P}_{22}^{(4)} \end{bmatrix}, \quad (\text{A.3})$$

em que

$$\mathbf{P}_{11}^{(4)} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 4-j & 4-2j & 4-2j & 4-4j & 2-4j & 2-4j & 1-4j & -4j & -1-4j & -2-4j & -2-4j & -4-4j & -4-2j & -4-j & -4-j \\ 4 & 4-2j & 4-4j & 2-4j & -4j & -2-4j & -4-4j & -4-2j & -4 & -4+2j & -4+4j & -2+4j & 4j & 2+4j & 4+4j & 4+2j \\ 4 & 4-2j & 2-4j & -1-4j & -4-4j & -4-j & -4+2j & -2+4j & 4j & 2+4j & 4+2j & 4-j & 4-4j & 1-4j & -2-4j & -4-2j \\ 4 & 4-4j & -4j & -4-4j & -4 & -4+4j & 4j & 4+4j & 4 & 4-4j & -4j & -4-4j & -4 & -4+4j & 4j & 4+4j \\ 4 & 2-4j & -2-4j & -4-j & -4+4j & 1+4j & 4+2j & 4-2j & -4j & -4-2j & -4+2j & -1+4j & 4+4j & 4-j & 2-4j & -2-4j \\ 4 & 2-4j & -4-4j & -4+2j & 4j & 4+2j & 4-4j & -2-4j & -4 & -2+4j & 4+4j & 4-2j & -4j & -4-2j & -4+4j & 2+4j \\ 4 & 1-4j & -4-2j & -2+4j & 4+4j & 4-2j & -2-4j & -4+j & 4j & 4+j & 2-4j & -4-2j & -4+4j & 2+4j & 4-2j & -1-4j \\ 4 & -4j & -4 & 4j & 4 & -4j & -4 & 4j & 4 & -4j & -4 & 4j & 4 & -4j & -4 & 4j \\ 4 & -1-4j & -4+2j & 2+4j & 4-4j & -4-2j & -2+4j & 4+j & -4j & -4+j & 2+4j & 4-2j & -4-4j & -2+4j & 4+2j & 1-4j \\ 4 & -2-4j & -4+4j & 4+2j & -4j & -4+2j & 4+4j & 2-4j & -4 & 2+4j & 4-4j & -4-2j & 4j & 4-2j & -4-4j & -2+4j \\ 4 & -2-4j & -2+4j & 4-j & -4-4j & -1+4j & 4-2j & -2-4j & 4j & 4-2j & -4-2j & 1+4j & 4-4j & -4-j & 2+4j & 2-4j \\ 4 & -4-4j & 4j & 4-4j & -4 & 4+4j & -4j & -4+4j & 4 & -4-4j & 4j & 4-4j & -4 & 4+4j & -4j & -4+4j \\ 4 & -4-2j & 2+4j & 1-4j & -4+4j & 4-j & -4-2j & 2+4j & -4j & -2+4j & 4-2j & -4-j & 4+4j & -1-4j & -2+4j & 4-2j \\ 4 & -4-2j & 4+4j & -2-4j & 4j & 2-4j & -4+4j & 4-2j & -4 & 4+2j & -4-4j & 2+4j & -4j & -2+4j & 4-4j & -4+2j \\ 4 & -4-j & 4+2j & -4-2j & 4+4j & -2-4j & 2+4j & -1-4j & 4j & 1-4j & -2+4j & 2-4j & -4+4j & 4-2j & -4+2j & 4-j \end{bmatrix}$$

$$\mathbf{P}_{12}^{(4)} = \begin{bmatrix} 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ -4 & -4+j & -4+2j & -4+2j & -4+4j & -2+4j & -2+4j & -1+4j & 4j & 1+4j & 2+4j & 2+4j & 4+4j & 4+2j & 4+2j & 4+j \\ 4 & 4-2j & 4-4j & 2-4j & -4j & -2-4j & -4-4j & -4-2j & -4 & -4+2j & -4+4j & -2+4j & 4j & 2+4j & 4+4j & 4+2j \\ -4 & -4+2j & -2+4j & 1+4j & 4+4j & 4+j & 4-2j & 2-4j & -4j & -2-4j & -4-2j & -4+j & -4+4j & -1+4j & 2+4j & 4+2j \\ 4 & 4-4j & -4j & -4-4j & -4 & -4+4j & 4j & 4+4j & 4 & 4-4j & -4j & -4-4j & -4 & -4+4j & 4j & 4+4j \\ -4 & -2+4j & 2+4j & 4+j & 4-4j & -1-4j & -4-2j & -4+2j & 4j & 4+2j & 4-2j & 1-4j & -4-4j & -4+j & -2+4j & 2+4j \\ 4 & 2-4j & -4-4j & -4+2j & 4j & 4+2j & 4-4j & -2-4j & -4 & -2+4j & 4+4j & 4-2j & -4j & -4-2j & -4+4j & 2+4j \\ -4 & -1+4j & 4+2j & 2-4j & -4-4j & -4+2j & 2+4j & 4-j & -4j & -4-j & -2+4j & 4+2j & 4-4j & -2-4j & -4+2j & 1+4j \\ 4 & -4j & -4 & 4j & 4 & -4j & -4 & 4j & 4 & -4j & -4 & 4j & 4 & -4j & -4 & 4j \\ -4 & 1+4j & 4-2j & -2-4j & -4+4j & 4+2j & 2-4j & -4-j & 4j & 4-j & -2-4j & -4+2j & 4+4j & 2-4j & -4-2j & -1+4j \\ 4 & -2-4j & -4+4j & 4+2j & -4j & -4+2j & 4+4j & 2-4j & -4 & 2+4j & 4-4j & -4-2j & 4j & 4-2j & -4-4j & -2+4j \\ -4 & 2+4j & 2-4j & -4+j & 4+4j & 1-4j & -4+2j & 4+2j & -4j & -4+2j & 4+2j & -1-4j & -4+4j & 4+j & -2-4j & -2+4j \\ 4 & -4-4j & 4j & 4-4j & -4 & 4+4j & -4j & -4+4j & 4 & -4-4j & 4j & 4-4j & -4 & 4+4j & -4j & -4+4j \\ -4 & 4+2j & -2-4j & -1+4j & 4-4j & -4+j & 4+2j & -2-4j & 4j & 2-4j & -4+2j & 4+j & -4-4j & 1+4j & 2-4j & -4+2j \\ 4 & -4-2j & 4+4j & -2-4j & 4j & 2-4j & -4+4j & 4-2j & -4 & 4+2j & -4-4j & 2+4j & -4j & -2+4j & 4-4j & -4+2j \\ -4 & 4+j & -4-2j & 4+2j & -4-4j & 2+4j & -2-4j & 1+4j & -4j & -1+4j & 2-4j & -2+4j & 4-4j & -4+2j & 4-2j & -4+j \end{bmatrix}$$

$$\mathbf{P}_{21}^{(4)} = \begin{bmatrix} 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 \\ 4 & -4+j & 4-2j & -4+2j & 4-4j & -2+4j & 2-4j & -1+4j & -4j & 1+4j & -2-4j & 2+4j & -4-4j & 4+2j & -4-2j & 4+j \\ 4 & -4+2j & 4-4j & -2+4j & -4j & 2+4j & -4-4j & 4+2j & -4 & 4-2j & -4+4j & 2-4j & 4j & -2-4j & 4+4j & -4-2j \\ 4 & -4+2j & 2-4j & 1+4j & -4-4j & 4+j & -4+2j & 2-4j & 4j & -2-4j & 4+2j & -4+j & 4-4j & -1+4j & -2-4j & 4+2j \\ 4 & -4+4j & -4j & 4+4j & -4 & 4-4j & 4j & -4-4j & 4 & -4+4j & -4j & 4+4j & -4 & 4-4j & 4j & -4-4j \\ 4 & -2+4j & -2-4j & 4+j & -4+4j & -1-4j & 4+2j & -4+2j & -4j & 4+2j & -4+2j & 1-4j & 4+4j & -4+j & 2-4j & 2+4j \\ 4 & -2+4j & -4-4j & 4-2j & 4j & -4-2j & 4-4j & 2+4j & -4 & 2-4j & 4+4j & -4+2j & -4j & 4+2j & -4+4j & -2-4j \\ 4 & -1+4j & -4-2j & 2-4j & 4+4j & -4+2j & -2-4j & 4-j & 4j & -4-j & 2-4j & 4+2j & -4+4j & -2-4j & 4-2j & 1+4j \\ 4 & 4j & -4 & -4j & 4 & 4j & -4 & -4j & 4 & 4j & -4 & -4j & 4 & 4j & -4 & -4j \\ 4 & 1+4j & -4+2j & -2-4j & 4-4j & 4+2j & -2+4j & -4-j & -4j & 4-j & 2+4j & -4+2j & -4-4j & 2-4j & 4+2j & -1+4j \\ 4 & 2+4j & -4+4j & -4-2j & -4j & 4-2j & 4+4j & -2+4j & -4 & -2-4j & 4-4j & 4+2j & 4j & -4+2j & -4-4j & 2-4j \\ 4 & 2+4j & -2+4j & -4+j & -4-4j & 1-4j & 4-2j & 4+2j & 4j & -4+2j & -4-2j & -1-4j & 4-4j & 4+j & 2+4j & -2+4j \\ 4 & 4+4j & 4j & -4+4j & -4 & -4-4j & -4j & 4-4j & 4 & 4+4j & 4j & -4+4j & -4 & -4-4j & -4j & 4-4j \\ 4 & 4+2j & 2+4j & -1+4j & -4+4j & -4+j & 4-2j & -2-4j & -4j & 2-4j & 4-2j & 4+j & 4+4j & 1+4j & -2+4j & -4+2j \\ 4 & 4+2j & 4+4j & 2+4j & 4j & -2+4j & -4+4j & -4+2j & -4 & -4-2j & -4-4j & -2-4j & -4j & 2-4j & 4-4j & 4-2j \\ 4 & 4+j & 4+2j & 4+2j & 4+4j & 2+4j & 2+4j & 1+4j & 4j & -1+4j & -2+4j & -2+4j & -4+4j & -4+2j & -4+2j & -4+j \end{bmatrix}$$

$$\mathbf{P}_{22}^{(4)} = \begin{bmatrix} 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 & 4 & -4 \\ -4 & 4-j & -4+2j & 4-2j & -4+4j & 2-4j & -2+4j & 1-4j & 4j & -1-4j & 2+4j & -2-4j & 4+4j & -4-2j & 4+2j & -4-j \\ 4 & -4+2j & 4-4j & -2+4j & -4j & 2+4j & -4-4j & 4+2j & -4 & 4-2j & -4+4j & 2-4j & 4j & -2-4j & 4+4j & -4-2j \\ -4 & 4-2j & -2+4j & -1-4j & 4+4j & -4-j & 4-2j & -2+4j & -4j & 2+4j & -4-2j & 4-j & -4+4j & 1-4j & 2+4j & -4-2j \\ 4 & -4+4j & -4j & 4+4j & -4 & 4-4j & 4j & -4-4j & 4 & -4+4j & -4j & 4+4j & -4 & 4-4j & 4j & -4-4j \\ -4 & 2-4j & 2+4j & -4-j & 4-4j & 1+4j & -4-2j & 4-2j & 4j & -4-2j & 4-2j & -1+4j & -4-4j & 4-j & -2+4j & -2-4j \\ 4 & -2+4j & -4-4j & 4-2j & 4j & -4-2j & 4-4j & 2+4j & -4 & 2-4j & 4+4j & -4+2j & -4j & 4+2j & -4+4j & -2-4j \\ -4 & 1-4j & 4+2j & -2+4j & -4-4j & 4-2j & 2+4j & -4+j & -4j & 4+j & -2+4j & -4-2j & 4-4j & 2+4j & -4+2j & -1-4j \\ 4 & 4j & -4 & -4j & 4 & 4j & -4 & -4j & 4 & 4j & -4 & -4j & 4 & 4j & -4 & -4j \\ -4 & -1-4j & 4-2j & 2+4j & -4+4j & -4-2j & 2-4j & 4+j & 4j & -4+j & -2-4j & 4-2j & 4+4j & -2+4j & -4-2j & 1-4j \\ 4 & 2+4j & -4+4j & -4-2j & -4j & 4-2j & 4+4j & -2+4j & -4 & -2-4j & 4-4j & 4+2j & 4j & -4+2j & -4-4j & 2-4j \\ -4 & -2-4j & 2-4j & 4-j & 4+4j & -1+4j & -4+2j & -4-2j & -4j & 4-2j & 4+2j & 1+4j & -4+4j & -4-j & -2-4j & 2-4j \\ 4 & 4+4j & 4j & -4+4j & -4 & -4-4j & -4j & 4-4j & 4 & 4+4j & 4j & -4+4j & -4 & -4-4j & -4j & 4-4j \\ -4 & -4-2j & -2-4j & 1-4j & 4-4j & 4-j & 4+2j & 2+4j & 4j & -2+4j & -4+2j & -4-j & -4-4j & -1-4j & 2-4j & 4-2j \\ 4 & 4+2j & 4+4j & 2+4j & 4j & -2+4j & -4+4j & -4+2j & -4 & -4-2j & -4-4j & -2-4j & -4j & 2-4j & 4-4j & 4-2j \\ -4 & -4-j & -4-2j & -4-2j & -4-4j & -2-4j & -2-4j & -1-4j & -4j & 1-4j & 2-4j & 2-4j & 4-4j & 4-2j & 4-2j & 4-j \end{bmatrix}$$