



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

GERALDO JOSÉ TRAVASSOS DE ARRUDA FILHO

**ESTRATÉGIAS PARA MELHORIA DE UM CODEC DE IMAGEM BASEADO NA  
TRANSFORMADA DISCRETA MANOBRÁVEL DO COSSENO**

Recife

2019

GERALDO JOSÉ TRAVASSOS DE ARRUDA FILHO

**ESTRATÉGIAS PARA MELHORIA DE UM CODEC DE IMAGEM BASEADO NA  
TRANSFORMADA DISCRETA MANOBRÁVEL DO COSSENO**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

**Área de Concentração:** Comunicações.

**Orientador:** Prof. Dr. Juliano Bandeira Lima

Recife

2019

Catálogo na fonte  
Bibliotecária Maria Luiza de Moura Ferreira, CRB-4 / 1469

A779e Arruda Filho, Geraldo José Travassos de.  
Estratégias para melhoria de um CODEC de imagem baseado na transformada discreta manobrável do cosseno / Geraldo José Travassos de Arruda Filho. - 2019.  
85 folhas, il., tab., abr., sigl. e símb.

Orientador: Prof. Dr. Juliano Bandeira Lima.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Elétrica, 2019.

Inclui Referências e Apêndice.

1. Engenharia Elétrica. 2. Transformada discreta do cosseno. 3. Transformada discreta manobrável do cosseno. 4. Codificação de imagem. 5. JPEG. I. Lima, Juliano Bandeira (Orientador). II. Título.

UFPE

621.3 CDD (22. ed.)

BCTG/2019-367

GERALDO JOSÉ TRAVASSOS DE ARRUDA FILHO

**ESTRATÉGIAS PARA MELHORIA DE UM CODEC DE IMAGEM BASEADO NA  
TRANSFORMADA DISCRETA MANOBRÁVEL DO COSSENO**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica.

**Área de Concentração:** Comunicações.

Aprovado em: 31 / 07 / 2019.

---

Prof. Dr. Juliano Bandeira Lima (Orientador)  
Universidade Federal de Pernambuco

---

Prof. Dr. Marcelo Cabral Cavalcanti (Coordenador do PPGEE)  
Universidade Federal de Pernambuco

---

Prof. Dr. João Marcelo Xavier Natario Teixeira (Examinador Externo)  
Universidade Federal de Pernambuco

---

Prof. Dr. Carlos Alexandre Barros de Mello (Examinador Externo)  
Universidade Federal de Pernambuco

*Aos meus pais.*

## AGRADECIMENTOS

Primeiramente gostaria de agradecer aos meus pais, Geraldo e Nadja, que nunca pouparam esforços para me proporcionar a melhor educação que estava ao alcance. Sem todo o apoio que recebi, não teria chegado onde estou hoje.

À minha esposa, Rafaella, por toda ajuda, compreensão e paciência ao longo das tantas horas que precisei dedicar a esta pesquisa durante os últimos dois anos.

Ao meu irmão Eduardo e a toda a minha família que, por sempre valorizarem minhas capacidades, compõem minha rede de apoio e me fornecem a segurança necessária para encarar e superar novos desafios.

Ao meu orientador Prof. Dr. Juliano Bandeira Lima, por todas as horas dedicadas a este trabalho, pela disponibilidade, pelos ensinamentos e pelo grande espírito de parceria com que me proporcionou a realização desta pesquisa.

A todos os amigos que acompanharam minha trajetória até aqui. Em especial, àqueles que mais contribuíram com a construção e manutenção do meu interesse pela ciência e pela eletrônica: Acácio, José Neto, Kleber e Vítor.

A tantos professores que, desde minha primeira infância, contribuíram com paciência e dedicação com a minha formação.

A todos os que fazem a Universidade Federal de Pernambuco que, durante tantos anos, e apesar das dificuldades, me proporcionou uma formação acadêmica de excelência.

À Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco pelo suporte financeiro recebido durante os primeiros meses deste trabalho.

## RESUMO

Esta dissertação de mestrado aborda estratégias baseadas na transformada discreta manobrável do cosseno (SDCT) para compressão com perdas de imagens estáticas. A referida transformada corresponde a uma generalização da DCT bidimensional obtida após a rotação, aos pares, de vetores de base desta última. A formação dos referidos pares é orientada pelo fato de a DCT bidimensional corresponder a uma das possíveis GFTs (transformadas de Fourier sobre grafos) associadas a grafos em grade, que por sua vez são comumente empregados para modelar a correlação entre um pixel e seus vizinhos nas imagens digitais. Nos capítulos iniciais, faz-se uma revisão bibliográfica sobre a SDCT, incluindo uma breve explicação sobre o processamento de sinais sobre grafos e a transformada de Fourier sobre grafos. Também são revisados os principais algoritmos propostos na literatura para a codificação de imagens utilizando esta transformada: SDCT-BT (árvore binária), SDCT-AM (minimização alternada), SDCT-1 (um único ângulo por bloco) e  $S^3$ DCT (esparsificadora de subespaços). A principal contribuição do presente trabalho foi a proposição de melhorias no sentido de obter um codificador de imagem competitivo utilizando a SDCT. Além disso, o desempenho de um CODEC (codificador e decodificador) de imagens completamente funcional baseado na SDCT é comparado pela primeira vez com uma alternativa já consolidada e amplamente utilizada em que a DCT é empregada neste mesmo sentido: a compressão de imagens padrão JPEG. Após a análise dos *tradeoffs* obtidos em termos de distorção, taxa e complexidade operacional, conclui-se que, no presente estado de desenvolvimento, apenas o algoritmo da SDCT-1 entrega resultados que possam justificar a sua utilização em substituição ao que é proposto pelo padrão JPEG. Nossos resultados sugerem que a transformada discreta manobrável do cosseno poderá ser eficientemente utilizada em trabalhos futuros para melhorar o desempenho de CODECs de imagens multidimensionais e *light fields* do tipo *lenslet*.

Palavras-chave: Transformada discreta do cosseno. Transformada discreta manobrável do cosseno. Codificação de imagem. JPEG.

## ABSTRACT

This dissertation addresses strategies for static image lossy compression based on the steerable discrete cosine transform (SDCT). The mentioned transform can be understood as a generalization of the bidimensional DCT transform and consists of rotations applied to pairs of its basis vectors. The choice of these pairs is oriented by the fact that the bidimensional DCT corresponds to one possible choice for the GFT (graph Fourier transform) for the grid graph, which are commonly used for modeling pixel neighborhood correlations in digital images. On the first chapters, we review the literature on SDCT and make a brief explanation about signal processing over graphs and Fourier transform over graphs. We also go over the most common algorithms suggested in literature for image coding using this transform: SDCT-BT (binary tree), SDCT-AM (alternated minimization), SDCT-1 (single angle per block) and  $S^3$ DCT (subspace sparsifying). The main contribution of this work lies on proposed improvements that may allow creating a competitive image encoder using SDCT. Furthermore, we also compare for the first time the performance of a fully functional SDCT-based image CODEC (encoder and decoder) and another DCT-based solution largely used for the same purpose: the JPEG image compression standard. The tradeoffs obtained concerning distortion, tax and operational complexity are analysed and the conclusion indicates that, in the present development state, only the SDCT-1 delivers results that may justify its use in substitution to JPEG. Our results suggest that Steerable Discrete Cosine Transform may be efficiently applied in further studies to improve lenslet light fields and multidimensional images CODEC performance.

Keywords: Discrete cosine transform. Steerable cosine transform. Image coding. JPEG.

## LISTA DE FIGURAS

Figura 1 – Exemplos de sinais sobre grafos. . . . .	24
Figura 2 – Grafo caminho $\mathcal{P}_4$ . . . . .	25
Figura 3 – Grafo grade $\mathcal{P}_4 \square \mathcal{P}_4$ . . . . .	26
Figura 4 – Em representação matricial, para $N = 8$ , são mostrados os vetores de base da 2D-DCT (a) e da SDCT para uma rotação de $\pi/2$ radianos (b). . . . .	28
Figura 5 – Rotação por um ângulo ótimo. . . . .	29
Figura 6 – Exemplo de árvore binária de decisões na SDCT-BT. . . . .	36
Figura 7 – Curvas RD para o CODEC baseado nos algoritmos originais (Peak Signal-to-Noise Ratio - PSNR). . . . .	42
Figura 8 – Curvas RD para o CODEC baseado nos algoritmos originais (Structural Similarity Index - SSIM). . . . .	42
Figura 9 – Cabeçalho do arquivo codificado contendo metadados necessários para decodificação. . . . .	43
Figura 10 – Os três tipos de ordenação zigue-zague utilizadas para: (a) coeficientes quantizados, (b) coeficientes quantizados da $S^3$ DCT e (c) valores dos ângulos. . . . .	47
Figura 11 – Curvas RD para o CODEC após melhorias no codificador aritmético (Peak Signal-to-Noise Ratio - PSNR). . . . .	48
Figura 12 – Curvas RD para o CODEC melhorias no codificador aritmético (Structural Similarity Index - SSIM). . . . .	48
Figura 13 – Vetores de base da 2D-DCT rotacionados pela SDCT-1 para diferentes valores de $\theta$ . . . . .	53
Figura 14 – Curvas RD para a imagem Boat (Peak Signal-to-Noise Ratio - PSNR). . . . .	57
Figura 15 – Curvas RD para a imagem F16 (Peak Signal-to-Noise Ratio - PSNR). . . . .	57
Figura 16 – Curvas RD para a imagem Boat (Structural Similarity Index - SSIM). . . . .	58
Figura 17 – Curvas RD para a imagem F16 (Structural Similarity Index - SSIM). . . . .	59
Figura 18 – Parte da imagem F16 após compressão a taxa de 1,05 bpp. . . . .	60
Figura 19 – Parte da imagem F16 após compressão a taxa de 1,05 bpp. . . . .	61
Figura 20 – Parte da imagem F16 após compressão a taxa de 0,5 bpp. . . . .	62
Figura 21 – Parte da imagem F16 após compressão a taxa de 0,5 bpp. . . . .	63
Figura 22 – Histograma de escolha de ângulos para a SDCT-1: o ângulo nulo foi escolhido em 40,7 % dos blocos processados. . . . .	66
Figura 23 – Histograma de escolha de ângulos para a SDCT-BT: o ângulo nulo foi escolhido em 41 % das sub-bandas criadas pelo algoritmo. . . . .	67
Figura 24 – Histograma de escolha de ângulos para a SDCT-AM. . . . .	67
Figura 25 – Histograma de escolha de ângulos para a $S^3$ DCT. . . . .	68

Figura 26 – Histograma de quantidade de sub-bandas por bloco para a SDCT-BT: uma única sub-banda é utilizada em 45,31 % dos blocos processados. . . . .	68
Figura 27 – Histograma de quantidade de sub-bandas por bloco para a SDCT-AM: a 2D-DCT usual (0 sub-bandas) é escolhida em 99,02 % dos blocos processados.	69
Figura 28 – Histograma de quantidade de sub-bandas por bloco para a S <sup>3</sup> DCT: a 2D-DCT usual (0 sub-bandas) é escolhida em 95,29 % dos blocos processados. . . . .	69

## LISTA DE TABELAS

Tabela 1	– Tempo para completar a seguinte tarefa: codificar e decodificar a imagem F16 utilizando cada um dos algoritmos disponíveis (SDCT-1, SDCT-AM, SDCT-BT, S <sup>3</sup> DCT, DCT e JPEG) dezoito vezes (dezoito níveis de qualidade), com 3 bits para quantização dos ângulos, e utilizando até 8 processos em paralelo.	44
Tabela 2	– Métrica de Bjontegaard considerando PSNR (BD-Rate): variações percentuais médias nas taxas de bits graças ao CABAC.	46
Tabela 3	– Tempo para completar a seguinte tarefa: codificar e decodificar a imagem F16 utilizando cada um dos algoritmos disponíveis (SDCT-1, SDCT-AM, SDCT-BT, S <sup>3</sup> DCT, DCT e JPEG) dezoito vezes (dezoito níveis de qualidade), com 3 bits para quantização dos ângulos, e utilizando até 8 processos em paralelo.	47
Tabela 4	– Métrica de Bjontegaard considerando PSNR (BD-Rate): variações percentuais médias na taxa de bits devido à utilização de quantização com ponderação baseada no Sistema Visual Humano (HVS).	50
Tabela 5	– Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits com a utilização de DPCM e deslocamento de faixa.	51
Tabela 6	– Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits com a nova faixa para escolha dos ângulos quantizados.	52
Tabela 7	– Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits com a utilização de apenas 4 níveis de quantização para os ângulos.	53
Tabela 8	– Variações percentuais médias nos tempos de codificação e decodificação devido à redução de 8 para 4 níveis de quantização.	54
Tabela 9	– Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits em relação à DCT usual.	56
Tabela 10	– Tempo de processamento necessário para codificar a imagem F16.	64
Tabela 11	– Tempo de processamento necessário para decodificar a imagem F16.	65
Tabela 12	– Métricas de Bjontegaard (BD-Rate, BD-PSNR, BD-SSIM): resultados médios para todas as 886 imagens do banco UCID.	66
Tabela 13	– Quadro comparativo geral com as particularidades de cada algoritmo estudado.	70

## LISTA DE ABREVIATURAS E SIGLAS

DFT	Transformada discreta de Fourier / Discrete Fourier transform
DHT	Transformada discreta de Hartley / Discrete Hartley transform
DTT	Transformadas discretas trigonométricas / Discrete trigonometric transforms
DCT	Transformada discreta do cosseno / Discrete cosine transform
DST	Transformada discreta do seno / Discrete sine transform
2D-DCT	Transformada discreta bidimensional do cosseno / Two-dimensional discrete cosine transform
DDCT	Transformada discreta direcional do cosseno / Directional discrete cosine transform
SDCT	Transformada discreta manobrável do cosseno / Steerable discrete cosine transform
S <sup>3</sup> DCT	Transformada discreta manobrável do cosseno esparsificadora de subespaços / Subspace sparsifying steerable discrete cosine transform
SDCT-AM	Transformada discreta manobrável do cosseno com minimização alternada / Alternated minimization steerable discrete cosine transform
SDCT-BT	Transformada discreta manobrável do cosseno com árvore binária / Binary tree steerable discrete cosine transform
GSP	Processamento de sinais sobre grafos / Graph signal processing
GFT	Transformada de Fourier sobre grafos / Graph Fourier transform
RD	Taxa-distorção / Rate-distortion
PSNR	Relação sinal-ruído de pico / Peak Signal-to-Noise Ratio
SSIM	Índice de similaridade estrutural / Structural Similarity Index

## LISTA DE SÍMBOLOS

$\mathbb{R}$	conjunto dos números reais
$\mathbb{Z}$	conjunto dos números inteiros
$(\cdot)^T$	transposição de linhas por colunas do argumento
$\otimes$	produto de Kronecker
$\text{sen}(\cdot)$	retorna o seno de seu argumento
$\text{cos}(\cdot)$	retorna o cosseno de seu argumento
$\text{arctan}(\cdot)$	retorna o arco tangente de seu argumento
$\text{cas}(\cdot)$	retorna $\text{cos}(\cdot) + \text{sen}(\cdot)$
$\text{arred}(\cdot)$	retorna o inteiro mais próximo de seu argumento
$\text{card}(\cdot)$	retorna a cardinalidade de seu argumento
$\min_{x \in D} f(x)$	minimiza a função $f(x)$ para $x$ assumindo algum valor no domínio $D$
$\mathbf{V}^{(i)}$	$i$ -ésima coluna da matriz $\mathbf{V}$
$\mathbf{V}(n)$	$n$ -ésima componente do vetor $\mathbf{V}$
$\mathcal{G}_1 \square \mathcal{G}_2$	produto cartesiano dos grafos $\mathcal{G}_1$ e $\mathcal{G}_2$
$L(\mathcal{G})$	matriz laplaciana do grafo $\mathcal{G}$
$\vec{\mathbf{0}}$	vetor nulo
$\ \cdot\ _i$	norma $\ell_i$
$\mathbf{A} \circ \mathbf{B}$	produto matricial de Hadamard entre $\mathbf{A}$ e $\mathbf{B}$ : $(\mathbf{A} \circ \mathbf{B})_{i,j} = \mathbf{A}_{i,j} \mathbf{B}_{i,j}$

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
1.1	TRANSFORMADAS DISCRETAS DIRECIONAIS E CODIFICAÇÃO DE IMAGEM	17
1.2	MOTIVAÇÃO	19
1.3	OBJETIVOS	20
1.4	ESTRUTURA DA DISSERTAÇÃO E CONTRIBUIÇÕES	20
<b>2</b>	<b>A TRANSFORMADA DISCRETA MANOBRÁVEL DO COSSENO</b>	<b>22</b>
2.1	PROCESSAMENTO DE SINAIS SOBRE GRAFOS	22
<b>2.1.1</b>	<b>Fundamentos de Teoria dos Grafos</b>	<b>22</b>
<b>2.1.2</b>	<b>Sinais e Transformada de Fourier sobre Grafos</b>	<b>23</b>
2.2	DEFINIÇÃO DA SDCT	26
<b>2.2.1</b>	<b>Ângulo Ótimo</b>	<b>27</b>
<b>3</b>	<b>CODIFICAÇÃO DE IMAGEM BASEADA NA SDCT</b>	<b>30</b>
3.1	MODELO RD	30
3.2	MINIMIZAÇÃO ALTERNADA: SDCT-AM	32
3.3	UTILIZANDO UM ÂNGULO ÚNICO: SDCT-1	34
3.4	ÁRVORE BINÁRIA PARA ESTRUTURA DE ÂNGULOS: SDCT-BT	35
3.5	ESPARSIFICAÇÃO DE SUBESPAÇOS: S <sup>3</sup> DCT	37
3.6	CODEC DE COMPRESSÃO DE IMAGENS BASEADO NA SDCT	38
<b>4</b>	<b>MELHORIAS NA CODIFICAÇÃO DE IMAGEM BASEADA NA SDCT</b>	<b>39</b>
4.1	CRIAÇÃO DE UM CODEC COMPLETO BASEADO EM SDCT	39
<b>4.1.1</b>	<b>Meta informações para o decodificador</b>	<b>41</b>
<b>4.1.2</b>	<b>Otimizações para tempo de execução</b>	<b>43</b>
<b>4.1.3</b>	<b>Context Based Arithmetic Coding (CABAC)</b>	<b>44</b>
4.2	MELHORIAS INCREMENTAIS EM DESEMPENHO RD	49
<b>4.2.1</b>	<b>Tabela de Quantização Baseada no Sistema Visual Humano (HVS)</b>	<b>49</b>
<b>4.2.2</b>	<b>Deslocamento de níveis de intensidade e codificação diferencial (DPCM)</b>	<b>50</b>
<b>4.2.3</b>	<b>Nova faixa para escolha dos ângulos de rotação</b>	<b>51</b>
<b>4.2.4</b>	<b>Redução na quantidade de níveis de quantização</b>	<b>52</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>55</b>
5.1	MÉTRICA DE BJONTEGAARD - PSNR	56
5.2	COMPARATIVO SSIM (STRUCTURAL SIMILARITY INDEX)	58
5.3	COMPARAÇÃO VISUAL SUBJETIVA	59
5.4	COMPARATIVO DE TEMPO DE CODIFICAÇÃO E DECODIFICAÇÃO	64
5.5	DESEMPENHO MÉDIO SOBRE IMAGENS DO BANCO UCID	65
5.6	QUADRO COMPARATIVO GERAL	70

<b>6</b>	<b>CONCLUSÕES . . . . .</b>	<b>71</b>
6.1	CONSIDERAÇÕES FINAIS . . . . .	71
6.2	TRABALHOS FUTUROS . . . . .	72
	<b>REFERÊNCIAS . . . . .</b>	<b>74</b>
	<b>APÊNDICE A – IMAGENS UTILIZADAS NESTE TRABALHO . . .</b>	<b>80</b>

## 1 INTRODUÇÃO

As transformadas estão entre as ferramentas matemáticas mais úteis em aplicações da Engenharia e, em particular, do processamento digital de sinais (SCHAFER; OPPENHEIM, 2010; POULARIKAS, 2018). Nesse contexto, a transformada mais conhecida é a transformada discreta de Fourier (DFT, do inglês *discrete Fourier transform*, a qual teve a sua aplicabilidade prática viabilizada a partir da concepção de algoritmos rápidos para o seu cálculo (COOLEY; TUKEY, 1965; BLAHUT, 2010). Com a DFT, um sinal discreto e com comprimento finito  $N$  é representado na base de funções exponenciais discretas  $\{v^{(k)}\}$ ,  $k = 0, 1, \dots, N - 1$ , em que

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

de maneira que essa transformada pode ser vista como um equivalente discreto da transformada de Fourier originalmente definida para sinais de variável contínua. Em meio à vasta gama de aplicações da DFT, merecem destaque aquelas proporcionadas pela propriedade de convolução cíclica dessa transformada, a qual permite que sejam eficientemente calculadas convoluções lineares entre um sinal e um filtro (SCHAFER; OPPENHEIM, 2010, p. 660).

Transformadas discretas de outros tipos também têm sido definidas. A transformada discreta de Hartley (DHT, do inglês *discrete Hartley transform*) (BRACEWELL, 1983), por exemplo, tem suas funções de base obtidas a partir da função *cas* (cosseno e seno, ou *cosine and sine*), isto é,  $\text{cas}(\cdot) = \cos(\cdot) + \text{sen}(\cdot)$ . Equivalentemente, a DHT de um sinal corresponde à diferença entre a parte real e a parte imaginária da DFT do mesmo sinal; assim, diferentemente da DFT, a DHT de um sinal real é também real. Isso torna essa transformada útil, por exemplo, em aplicações relacionadas ao processamento de sinais no domínio óptico (AZIM; GUENNEC; MAURY, 2018; HAJOMER; YANG; HU, 2018).

Outra família bem conhecida de transformadas é a das transformadas discretas trigonométricas (DTT, do inglês *discrete trigonometric transforms*), a qual é constituída por 8 transformadas discretas do cosseno (DCT, do inglês *discrete cosine transform*) e 8 transformadas discretas do seno (DST, do inglês *discrete sine transform*) (MARTUCCI, 1994; STRANG, 1999). Uma das maneiras de definir essas transformadas é considerar que o sinal cuja transformada se deseja calcular tem seu comprimento inicialmente duplicado, por meio de uma extensão simétrica, e, em seguida, quadruplicado<sup>1</sup> (em relação ao comprimento original) por meio de uma nova extensão simétrica. Daí, calcula-se a DFT dessa versão estendida do sinal e, após algumas simplificações associadas às simetrias e a inclusão de fatores de ponderação, obtêm-se transformadas cujo núcleo conserva apenas a função (discreta) do seno ou a do cosseno. Como quatro tipos de

<sup>1</sup> Na verdade, o tamanho pode ser efetivamente quadruplicado ou *quase-quadruplicado*, uma vez que as extensões simétricas realizadas podem ou não repetir as amostras que se encontram na última posição do sinal a ser estendido.

extensões simétricas são admissíveis (dois tipos de extensão par e dois de extensão ímpar), 16 transformadas distintas são obtidas.

Dentre outras transformadas, têm sido ainda consideradas a transformada discreta de wavelet (HEIL; WALNUT, 1989; BAIG et al., 2019), a transformada discreta de Hilbert (CIZEK, 1970; BISWAL; BISWAL; MALIK, 2018) e a transformada discreta da Hadamard (AHMED; RAO, 2012; ZHENG; HUANG, 2018). Além disso, generalizações dessas transformadas e das transformadas acima mencionadas também têm sido propostas. Nessa linha, têm sido consideradas transformadas discretas fracionárias, as quais possuem como operador matricial associado uma potência não-inteira do operador matricial da respectiva transformada ordinária (CANDAN; KUTAY; OZAKTAS, 2000; PEI; YEH, 2001; HSUE; CHANG, 2015; ANNABY; RUSHDI; NEHARY, 2016; SERBES, 2017); assim, se se denotar por  $M$  a matriz de uma certa transformada discreta, a matriz da transformada fracionária correspondente será dada por  $M^a$ , em que  $a \in \mathbb{R}$  (o conjunto dos números reais). Transformadas fracionárias têm sido empregadas, por exemplo, em análise tempo-frequência e cifragem de imagens (LU; XIAO; WEI, 2016; LIU et al., 2015). Transformadas discretas também podem ser definidas sobre estruturas algébricas finitas, como anéis e corpos finitos (BLAHUT, 2010; CAMPELLO DE SOUZA et al., 1998; LIMA; CAMPELLO DE SOUZA, 2011; LIMA; CAMPELLO DE SOUZA, 2016). Nesses casos, o núcleo da transformada guarda, normalmente, uma analogia com o da respectiva transformada definida sobre números reais ou complexos; por outro lado, os cenários de aplicação são diversos, estendendo-se da codificação de canal à criptografia (BLAHUT, 2003; LIMA; MADEIRO; SALES, 2015; FEKRI et al., 2006).

Uma generalização para transformadas discretas mais recentemente proposta é a que se identifica como transformada de Fourier sobre grafos (GFT, do inglês *graph Fourier transform*) (SHUMAN et al., 2013; ORTEGA et al., 2018). Essa é uma das operações fundamentais do chamado processamento de sinais sobre grafos (GSP, do inglês *graph signal processing*) (ORTEGA et al., 2018), em que são considerados sinais que correspondem a um mapeamento nos vértices de um grafo que modela um certo domínio irregular. Uma rede de sensores espalhados arbitrariamente por uma determinada região geográfica, por exemplo, pode ser representada por um grafo cujos vértices estão associados aos sensores (e suas respectivas posições), cujas arestas e seus pesos são inferidos utilizando algum critério plausível para o problema em questão (proximidade geográfica, por exemplo) e sobre o qual se encontram amostras de um sinal com valores que representam o que os sensores estão medindo (temperatura, por exemplo). A ideia do GSP é que todo o processamento (transformações, filtragem, amostragem etc.) leve em conta a topologia do grafo, em vez de considerar apenas domínios regulares como o tempo e o espaço discretos. A GFT consiste na representação de um sinal sobre um grafo numa base formada por autovetores de um operador que é construído segundo as especificidades de cada grafo (normalmente a matriz de adjacência ou o Laplaciano); tal representação pode ser dita generalizada porque, para grafos com topologias particulares, a transformada coincide com transformadas discretas conhecidas. De modo mais específico, a GFT para um grafo em anel

direcionado coincide com a DFT (FRACASTORO; MAGLI, 2017) e, para um grafo em caminho, coincide com a DCT do tipo 2 (FRACASTORO; FOSSON; MAGLI, 2017).

## 1.1 TRANSFORMADAS DISCRETAS DIRECIONAIS E CODIFICAÇÃO DE IMAGEM

Uma das aplicações em que as transformadas discretas desempenham um papel de destaque é a compressão de sinais. Neste cenário, a ferramenta mais largamente empregada é a DCT do tipo 2<sup>2</sup> (AHMED; NATARAJAN; RAO, 1974; GONZALEZ; WOODS, 2017), que possui a importante propriedade de compactação de energia (SCHAFER; OPPENHEIM, 2010); noutras palavras, a aplicação de uma DCT bidimensional a sinais em geral e, em particular, a sinais de imagens naturais em formato digital, cujas amostras apresentam alta correlação espacial, produz uma representação em que os coeficientes de ordens mais baixas (baixas frequências) concentram um percentual significativo da energia total do sinal. Isso é decorrente das extensões simétricas usadas na construção dessa transformada, que, diferentemente do que acontece na definição de outras transformadas discretas, eliminam certas transições abruptas de valor entre amostras subsequentes do sinal (MARTUCCI, 1994). Tal artifício favorece a referida compactação e permite que se obtenha uma versão mais *enxuta* do sinal por meio da preservação apenas dos coeficientes mais representativos para sua reconstrução (DCT inversa). Naturalmente, tal procedimento se encontra associado à compressão com perdas, já que os coeficientes menos representativos a cada bloco de imagem que é transformado são anulados e mesmo os mais significativos são aproximados e quantizados.

A estratégia de compressão descrita é bastante eficiente quando blocos de imagem com descontinuidades predominantemente nas direções horizontal e vertical são tratados; isto é, são conseguidas boas relações (quanto menor melhor) entre o tamanho da imagem após a compressão (espaço necessário para armazenamento num disco) e alguma métrica de similaridade entre sua versão reconstruída com perda e a sua versão original. Tal fenômeno é associado ao fato de, sendo a DCT uma transformada separável, a aplicação de sua versão bidimensional equivaler à aplicação inicial de DCTs unidimensionais às colunas da estrutura bidimensional cuja transformada se esteja calculando e, depois, à aplicação de novas DCTs unidimensionais às linhas da mesma estrutura (já com as colunas transformadas). Porém, quando um bloco de imagem possui significativas descontinuidades em direções arbitrárias, tem-se mostrado que o uso da DCT se torna menos eficiente (KIM; NARANG; ORTEGA, 2012). Diante disso, têm sido propostos ajustes que incorporam informação direcional à definição da DCT, com o propósito de adaptá-la às referidas descontinuidades e melhorar sua eficiência na compressão (ZENG; FU, 2008; CHANG; GIROD, 2008; KAMISLI; LIM, 2009; COHEN et al., 2010; DRÉMEAU et al., 2010).

Em Zeng e Fu (2008), por exemplo, é descrita a DCT direcional (DDCT, do inglês *directional discrete cosine transform*). Nessa solução, na primeira etapa de cálculo da DCT

<sup>2</sup> Deste ponto em diante no texto, essa transformada será identificada pelo acrônimo DCT apenas.

bidimensional (DCT unidimensional aplicada às colunas), a transformada pode ser aplicada numa direção diferente da vertical e da horizontal. Os coeficientes produzidos nesta etapa são organizados apropriadamente, de modo que a segunda transformada possa ser aplicada aos coeficientes que estão melhor alinhados uns com os outros. Em Chang e Girod (2008), os autores introduziram novas direções para a primeira etapa de transformação e propuseram um novo método de varredura em zigue-zague dos coeficientes. Em Kamisli e Lim (2009), sugere-se não aplicar as DCTs unidimensionais da segunda etapa (linhas) ou aplicá-las apenas aos coeficientes DC gerados na primeira etapa de transformação (COHEN et al., 2010). Em Drémeau et al. (2010), a DDCT é aperfeiçoada usando suportes de base local anisotrópica, em que a base ideal é selecionada explorando a estrutura em árvore binária de um dicionário. Entretanto, os métodos baseados na DDCT têm algumas restrições. Em particular, eles requerem DCTs unidimensionais de comprimentos distintos, alguns dos quais são muito curtos e nem sempre são uma potência de 2; além disso, as DCTs da segunda etapa (linhas) nem sempre podem ser aplicadas a coeficientes de frequências AC similares (YEO et al., 2012).

Uma solução diferente das fundamentadas na DDCT é a apresentada em Xu, Xu e Wu (2007), em que se define a chamada DCT baseada em *lifting*. Essa transformada é construída levando em conta atributos de direção de cada bloco da imagem; tais atributos são incorporados à definição da DCT por meio de operações primárias direcionais que atuam numa primeira etapa de transformação unidimensional (linhas ou colunas). Embora a DCT baseada em *lifting* possa ser aplicada ao longo de qualquer direção, ela se estende além dos limites de cada bloco sendo processado, pois parâmetros relacionados à adaptação de direção (na transição de um bloco para o outro) também precisam ser considerados.

Numa série de trabalhos recentes, Fracastoro *et al.* apresentaram soluções para codificação de imagem baseadas na transformada discreta manobrável do cosseno (SDCT, do inglês *steerable discrete cosine transform*) (FRACASTORO; MAGLI, 2015; FRACASTORO; MAGLI, 2016; FRACASTORO; FOSSON; MAGLI, 2017). Esta transformada corresponde a uma generalização da DCT bidimensional obtida após a rotação, aos pares, de vetores de base desta última. A formação dos referidos pares é orientada pelo fato de a DCT bidimensional corresponder a uma das possíveis GFTs associadas a grafos em grade, os quais são comumente empregados para modelar imagens digitais (num grafo deste tipo, cada vértice, com exceção daqueles que se encontram nas bordas da grade, é conectado por arestas com seus vizinhos imediatos horizontais e verticais). Mais precisamente, formam um par a ser rotacionado dois vetores de base da DCT bidimensional com tamanho  $N \times N$  que estão associados ao mesmo autovalor da Laplaciana do grafo em grade com igual tamanho. As rotações em questão produzem um efeito semelhante ao que já foi discutido: o de alinhar melhor os vetores usados na transformação (mudança de base) à direção das transições de intensidade presentes num bloco de imagem e conseguir, assim, uma representação mais compacta e que possa ser aproveitada por um codificador para obter um melhor desempenho a título de compressão.

A aplicação da SDCT pode levar à exata anulação de até  $N(N - 1)/2$  coeficientes dentre os  $N^2$  coeficientes de um bloco de imagem  $N \times N$  no domínio da transformada. Todavia, tal possibilidade exigiria que fosse armazenada a mesma quantidade de ângulos de rotação e, dessa forma, não se conseguiria tirar proveito, na codificação, da representação mais compacta proporcionada pela SDCT. Em função disso, os criadores da transformada propuseram algumas soluções sub-ótimas, mas que, de algum modo, viabilizam o efetivo emprego da ferramenta no cenário em questão. Em Fracastoro e Magli (2015), propõe-se a SDCT-1, em que a SDCT é construída considerando que, em vez de  $N(N - 1)/2$  ângulos possivelmente diferentes, um mesmo ângulo é usado para rotacionar todos os pares de vetores empregados na transformação de um certo bloco de imagem; este ângulo único é escolhido de um conjunto finito resultante da quantização de um intervalo contínuo de ângulos. Em Fracastoro e Magli (2016), é proposta a S<sup>3</sup>DCT (SDCT esparsificadora de subespaços, do inglês *subspace sparsifying SDCT*), que admite a seleção de um ângulo para cada par de vetores a ser rotacionado; tal seleção é, porém, feita a partir de um conjunto finito de ângulos resultantes de uma quantização como a que se acabou de mencionar. Por fim, em Fracastoro, Fosson e Magli (2017), são apresentadas a SDCT-BT (SDCT - árvore binária, do inglês *SDCT - binary tree*) e a SDCT-AM (SDCT - minimização alternada, do inglês *SDCT - alternated minimization*). Na primeira, usa-se um algoritmo que permite a realização de buscas exaustivas ao longo de todos os valores de ângulos quantizados disponíveis, de modo semelhante ao que é realizado no algoritmo da SDCT-1, ao mesmo tempo em que permite a utilização de mais de um ângulo por bloco. Na segunda, são definidos grupos de pares de vetores que serão rotacionados por um mesmo ângulo. Algumas das soluções acima descritas foram combinadas com outras operações necessárias à composição de um codificador de imagem completo, a saber, quantização dos coeficientes no domínio da transformada e codificação de entropia. Os resultados obtidos demonstram que, em geral, codificadores baseados na SDCT têm desempenho melhor em termos de taxa-distorção que aqueles baseados na DCT e na DDCT.

## 1.2 MOTIVAÇÃO

Apesar de estar documentada na literatura a obtenção de ganhos significativos na compressão de imagens utilizando SDCT quando comparada a soluções empregando a DCT (FRACASTORO; MAGLI, 2015; FRACASTORO; MAGLI, 2016; FRACASTORO; FOSSON; MAGLI, 2017; FRACASTORO; MAGLI, 2017), não havia sido documentados, ao melhor do nosso conhecimento, experimentos em que um CODEC de imagem completamente funcional empregando a SDCT fosse comparado com alternativas já consolidadas e amplamente utilizadas em que a DCT é empregada neste mesmo sentido de forma eficiente, como por exemplo aquela proposta pelo padrão JPEG (PENNEBAKER, 1993). Além disso, não foram encontradas também considerações a respeito dos *tradeoffs* obtidos com o acréscimo na complexidade computacional imposta pelos algoritmos de otimização RD empregados para escolha dos ângulos de rotação utilizados pela SDCT.

Por outro lado, ao mesmo tempo em que enxergamos a necessidade de obtenção de resultados mais realistas - entenda-se, mais próximos de um cenário de aplicação final - para um CODEC de imagem baseado em SDCT, era possível visualizar possibilidades de melhorias nos algoritmos. Entre as possibilidades, destacava-se a necessidade de propor um esquema para compressão dos dados adicionais gerados pelos ângulos de rotação.

### 1.3 OBJETIVOS

O objetivo geral deste trabalho é investigar o desempenho do emprego da SDCT para compressão de imagens, bem como propor melhorias no sentido de torná-la competitiva quando comparada às soluções que empregam a 2D-DCT usual. Os objetivos específicos são os seguintes:

1. Criação de um CODEC de imagem completamente funcional baseado na SDCT.
2. Proposta de um método adequado para compressão adicional dos ângulos de rotação.
3. Comparação dos resultados obtidos com o desempenho da codificação JPEG.
4. Análise dos *tradeoffs* obtidos entre desempenho RD e complexidade.

### 1.4 ESTRUTURA DA DISSERTAÇÃO E CONTRIBUIÇÕES

Esta dissertação está organizada da seguinte forma<sup>3</sup>:

- No Capítulo 1, é feita uma introdução ao tema central do trabalho, provendo a contextualização fundamental sobre a qual a pesquisa foi realizada. São apresentados também os principais motivos para o seu desenvolvimento e seus objetivos.
- No Capítulo 2, é feita uma revisão bibliográfica dos conceitos matemáticos preliminares relacionados ao processamento de sinais sobre grafos (GSP) e transformada de Fourier sobre grafos (GFT), que por sua vez é utilizada como referência para revisar a definição formal da transformada manobrável do cosseno (SDCT).
- No Capítulo 3, é feita uma revisão sobre o conceito de modelo RD (do inglês *rate distortion*) e são descritos os algoritmos e esquemas para codificação de imagens estáticas baseada na SDCT que têm sido propostos na literatura.
- No Capítulo 4, são apresentadas as principais contribuições desta dissertação: (i) descrição de um CODEC de imagem completamente funcional baseado nos algoritmos já propostos na literatura, (ii) comparação dos resultados com o JPEG, (iii) proposta de utilização de um codificador aritmético já descrito na literatura, com modificações que permitiram

<sup>3</sup> Ao longo desta seção, as principais contribuições desta dissertação são indicadas por meio de uma numeração em algarismos romanos.

a sua utilização também para compressão dos ângulos, (iv) utilização de codificação diferencial para os coeficientes DC, (v) proposta de um esquema para utilização de tabela de quantização baseada no sistema visual humano no CODEC SDCT e (vi) exploração de simetrias para escolha da faixa de ângulos a ser utilizada em cada algoritmo SDCT.

- No Capítulo 5, são apresentados os principais resultados obtidos pelo CODEC SDCT após as melhorias propostas nesta dissertação. O desempenho do CODEC é comparado com aquele que seria obtido pelo emprego da 2D-DCT usual, tanto do ponto de vista de taxa-distorção como do ponto de vista de complexidade computacional.
- No Capítulo 6, são apresentadas as considerações conclusivas desta dissertação, com a indicação de oportunidades para a realização de pesquisas futuras.

## 2 A TRANSFORMADA DISCRETA MANOBRÁVEL DO COSSENO

Neste capítulo, é feita uma revisão sobre a transformada discreta manobrável do cosseno (SDCT). Como o estabelecimento dessa ferramenta está relacionado à transformada de Fourier definida sobre grafos, faz-se, inicialmente, uma sucinta explanação acerca do chamado processamento de sinais sobre grafos (GSP). O capítulo é iniciado com a apresentação de alguns dos principais fundamentos acerca da teoria dos grafos, a qual, juntamente com conceitos oriundos do processamento de sinais clássico, viabiliza o GSP; a transformada de Fourier sobre grafos (GFT, do inglês *graph Fourier transform*) é definida e particularidades relacionadas à GFT para grafos caminho e grade são discutidas. Em seguida, a SDCT é apresentada e algumas de suas principais propriedades são discutidas.

### 2.1 PROCESSAMENTO DE SINAIS SOBRE GRAFOS

Nas duas subseções seguintes são apresentados alguns dos principais fundamentos sobre teoria dos grafos, incluindo os conceitos de sinais e transformada de Fourier sobre grafos.

#### 2.1.1 Fundamentos de Teoria dos Grafos

Um grafo corresponde a uma representação gráfica de uma relação em um conjunto. É usualmente denotado como  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , em que  $\mathcal{V}$  é o conjunto de vértices e  $\mathcal{E}$  é o conjunto de arestas, que correspondem a pares não ordenados de elementos distintos de  $\mathcal{V}$  (DIESTEL, 1997; BONDY; MURTY, 1976). Grafos têm aplicação em Engenharias (HARRISON, 2016; SLUSARCZYK et al., 2017; CIONINI et al., 2017; LU; LIU; YAN, 2018), Computação (ZHENG; DUNAGAN; KAPOOR, 2011; KOTENKO; DOYNIKOVA; CHECHULIN, 2014) e Matemática (SHI et al., 2017; KUZIAK; YERO; RODRIGUEZ-VELAZQUEZ, 2013; FRANCIS et al., 2017), por exemplo.

A matriz de adjacência  $\mathbf{A}(\mathcal{G})$  de um grafo  $\mathcal{G}$  com  $N$  vértices tem sua componente na  $i$ -ésima linha e na  $j$ -ésima coluna,  $i, j = 1, 2, \dots, N$ , dada por (DIESTEL, 1997)

$$a_{ij} = \begin{cases} 1, & \text{se } \{v_i, v_j\} \in \mathcal{E}, \\ 0, & \text{caso contrário.} \end{cases} \quad (2.1)$$

O grau de um vértice  $v_i$  é denotado por  $d(v_i)$  e corresponde à quantidade de arestas adjacentes a ele (DIESTEL, 1997). A matriz de grau  $\mathbf{D}(\mathcal{G})$  é uma matriz diagonal cuja componente na  $i$ -ésima linha e na  $i$ -ésima coluna,  $i = 1, 2, \dots, N$ , é dada por  $d(v_i)$ .

A matriz Laplaciana de um grafo  $\mathcal{G}$  é definida como  $\mathbf{L}(\mathcal{G}) = \mathbf{D}(\mathcal{G}) - \mathbf{A}(\mathcal{G})$  (CHUNG, 1997). Como a matriz de adjacência é simétrica e real (assumindo que se está considerando apenas grafos não-direcionados e com arestas com pesos reais) e  $\mathbf{D}(\mathcal{G})$  é uma matriz diagonal,

$\mathbf{L}(\mathcal{G})$  é uma matriz simétrica e real e, portanto, é diagonalizável por uma matriz ortogonal  $\mathbf{U}^{-1} = \mathbf{U}^T \in \mathbb{R}^{N \times N}$ , em que  $T$  indica a transposição de linhas por colunas do argumento, tal que

$$\mathbf{L}(\mathcal{G}) = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}; \quad (2.2)$$

na última equação, as colunas  $\mathbf{u}^{(k)}$ ,  $k = 0, 1, \dots, N - 1$ , da matriz  $\mathbf{U}$  são autovetores de  $\mathbf{L}(\mathcal{G})$  e  $\mathbf{\Lambda} \in \mathbb{R}^{N \times N}$  é uma matriz diagonal contendo os respectivos autovalores,  $\lambda_0, \dots, \lambda_{N-1}$  (SHUMAN et al., 2013).

### 2.1.2 Sinais e Transformada de Fourier sobre Grafos

O processamento de sinais sobre grafos tem como foco a representação e o processamento de sinais discretos cujo domínio subjacente pode ser modelado como um grafo (ORTEGA et al., 2018). Nesse contexto, um sinal sobre um grafo é definido como um mapeamento nos vértices do grafo. Mais especificamente, seja  $\mathbf{f}$  um sinal sobre o grafo  $\mathcal{G}$ , em que cada componente  $f(n)$ ,  $n = 0, 1, \dots, N - 1$ , é associada ao vértice  $v_i \in \mathcal{V}$ . O sinal sobre o grafo pode ser escrito vetorialmente como

$$\mathbf{f} = [f(0), f(1), \dots, f(N - 1)] \in \mathbb{R}^N. \quad (2.3)$$

Na Figura 1, são exibidos exemplos de sinais sobre grafos (a rotulação dos vértices é omitida; pode-se assumir que a amostra de sinal  $f(n)$  é atribuída ao vértice  $v_n$ ). O grafo em ciclo  $\mathcal{C}_N$ , mostrado na Figura 1a, pode ser empregado para modelar um eixo de tempo discreto com duração finita (e cíclica); os valores do sinal são indicados escrevendo seu valor numérico ao lado do respectivo vértice. No grafo de Dürer mostrado na Figura 1b, barras verticais com alturas que indicam o valor das amostras de sinal são usadas. A Figura 1c ilustra um sinal definido por uma rede de sensores; as arestas do grafo podem ser ponderadas pelo inverso da distância Euclidiana entre os vértices e uma escala de cores é usada para indicar os valores do sinal.

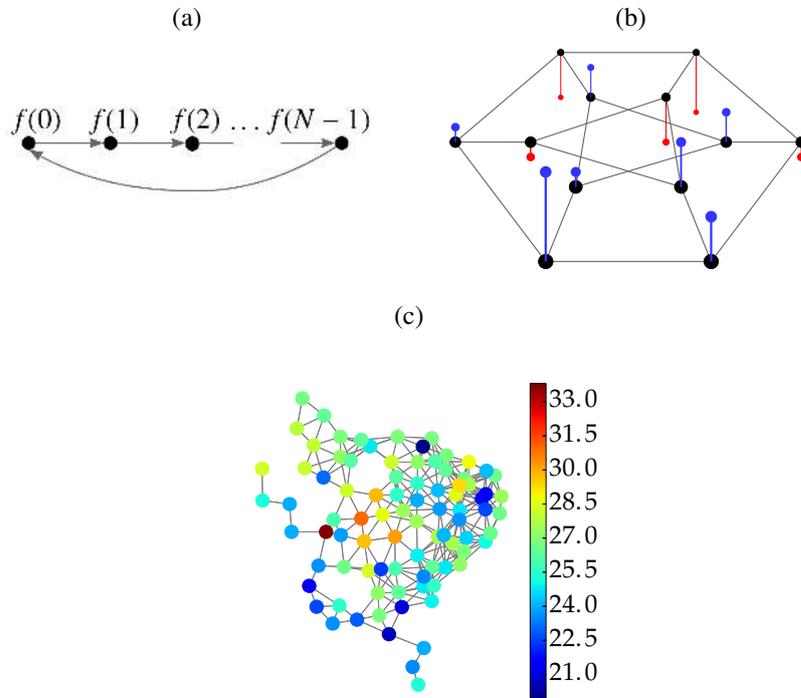
A transformada clássica de Fourier (FT, do inglês *Fourier transform*) consiste basicamente em expandir uma função  $f$  em termos de exponenciais complexas  $e^{2\pi i \xi t}$ , que são autofunções do operador unidimensional de Laplace  $\frac{d^2}{dt^2}$ . No GSP, define-se uma transformada de Fourier sobre grafos que, em certo sentido, é análoga à FT. Isso é feito expandindo  $\mathbf{f} \in \mathbb{R}^N$  em termos dos autovetores do Laplaciano do grafo, isto é,

$$\hat{f}(\lambda_k) := \langle \mathbf{f}, \mathbf{u}^{(k)} \rangle = \sum_{n=0}^{N-1} f(n) [u^{(k)}(n)]^*, \quad (2.4)$$

$k = 0, 1, \dots, N - 1$ . Matricialmente, tem-se

$$\hat{\mathbf{f}} = \mathbf{U}^{-1}\mathbf{f}. \quad (2.5)$$

Figura 1 – Exemplos de sinais sobre grafos: (a) um grafo em ciclo direcionado, (b) um grafo de Dürer não direcionado e (c) um grafo de cidades da região Nordeste do Brasil sobre o qual foi definido um sinal de medições de temperatura a partir de 1º de Fevereiro de 2012 (Banco de Dados Meteorológicos para Ensino e Pesquisa, BDMEP; disponível em: <http://www.inmet.gov.br/portal/index.php?r=bdmep/bdmep>).



Fonte: Ribeiro (2018).

A GFT inversa é dada por

$$f(n) = \sum_{k=0}^{N-1} \hat{f}(\lambda_k) u^{(k)}(n), \quad (2.6)$$

$n = 0, 1, \dots, N - 1$ , ou, matricialmente,

$$\mathbf{f} = \mathbf{U}\hat{\mathbf{f}}. \quad (2.7)$$

Então, a definição de uma GFT depende, naturalmente, da topologia do grafo sobre o qual as amostras de determinado sinal são dispostas. Além disso, grafos com topologias específicas podem ter GFTs associadas que coincidem com transformadas discretas conhecidas. Por exemplo, a transformada discreta do cosseno (DCT, do inglês *discrete cosine transform*) corresponde à GFT que se obtém quando se considera um grafo caminho  $\mathcal{P}_N$  com  $N$  vértices (veja o grafo  $\mathcal{P}_4$  na Figura 2). De fato, mostra-se que os autovetores de  $\mathbf{L}(\mathcal{P}_N)$  são iguais aos vetores de base  $\mathbf{v}^{(k)}$ ,  $k = 0, 1, \dots, N - 1$ , da DCT do tipo 2 (STRANG, 1999; MARTUCCI, 1994), os quais têm suas componentes dadas por

$$v^{(k)}(n) = \cos\left(\frac{\pi k}{N} \left(n + \frac{1}{2}\right)\right), \quad n = 0, 1, \dots, N - 1. \quad (2.8)$$

Figura 2 – Grafo caminho  $\mathcal{P}_4$ .

Fonte: Fracastoro, Fosson e Magli (2017).

Cada  $\mathbf{v}^{(k)}$  é um autovetor de  $\mathbf{L}(\mathcal{P}_N)$  associado ao autovalor

$$\lambda_k = 4\text{sen}^2\left(\frac{\pi k}{2N}\right), \quad (2.9)$$

o qual possui multiplicidade igual a 1. Consequentemente, os vetores de base da DCT compõem a única base ortonormal de  $\mathbb{R}^N$  formada por autovetores de  $\mathbf{L}(\mathcal{P}_N)$ .

O produto cartesiano entre dois grafos caminho corresponde a um grafo grade  $\mathcal{P}_N \square \mathcal{P}_N$ , com  $N^2$  vértices (veja na Figura 3 um exemplo para  $N = 4$ ). Uma das possíveis bases de  $\mathbb{R}^{N \times N}$  formada por autovetores de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$  corresponde ao conjunto de vetores de base da 2D-DCT (ZHANG; FLORÊNCIO, 2013). Assim, a 2D-DCT coincide com uma GFT para o grafo em questão. O que há de diferente com relação ao caso unidimensional, associado ao grafo caminho, é a multiplicidade dos autovalores associados. Mais especificamente, de acordo com Merris (1994) e Merris (1998), os autovalores de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$  são todas as possíveis somas

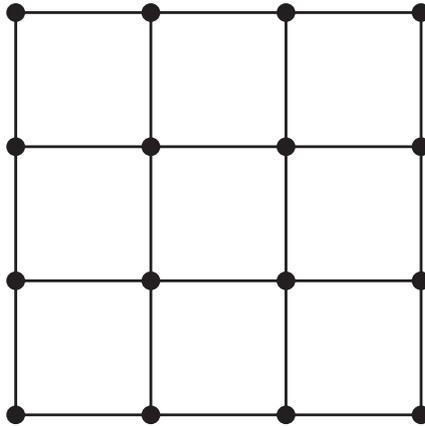
$$\lambda_{k,l} = \lambda_k + \lambda_l = 4\text{sen}^2\left(\frac{\pi k}{2N}\right) + 4\text{sen}^2\left(\frac{\pi l}{2N}\right) \quad (2.10)$$

e os autovetores correspondentes são determinados pelo produto de Kronecker entre os respectivos autovetores de  $\mathcal{P}_N$ , isto é,

$$\mathbf{v}^{(k,l)} = \mathbf{v}^{(k)} \otimes \mathbf{v}^{(l)}, \quad k, l = 0, 1, \dots, N - 1. \quad (2.11)$$

Considerando (2.10), observa-se que alguns autovalores de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$  aparecem repetidos devido à simetria  $\lambda_{k,l} = \lambda_{l,k}$ , para  $k \neq l$ . Assim, os referidos autovalores têm as seguintes multiplicidades algébricas:

- O autovalor  $\lambda_{k,l} = 4$  possui multiplicidade  $N - 1$  e corresponde a todos os autovalores  $\lambda_{k,N-k}$  com  $1 \leq k \leq N - 1$ ;
- O autovalor  $\lambda_{k,l}$  é igual a  $\lambda_{l,k}$ , para  $k \neq l$ , devido à simetria,  $\lambda_{k,l} = \lambda_k + \lambda_l = \lambda_l + \lambda_k = \lambda_{l,k}$ , e possui multiplicidade igual a 2;

Figura 3 – Grafo grade  $\mathcal{P}_4 \square \mathcal{P}_4$ .

Fonte: Fracastoro, Fosson e Magli (2017).

- O autovalor  $\lambda_{k,l}$  possui multiplicidade igual a 1, quando  $k = l$  e  $k \neq \frac{N}{2}$ . Logo, no espectro de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$ , há  $N - 1$  autovalores com multiplicidade algébrica igual a 1.

De acordo com (2.11), os autovetores de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$  são obtidos a partir de produtos de Kronecker; como tal operação é não-comutativa, os autovetores  $\mathbf{v}^{(k,l)}$  e  $\mathbf{v}^{(l,k)}$  são linearmente independentes, ainda que estejam associados ao mesmo autovalor. Portanto, a multiplicidade geométrica desses autovalores coincide com a algébrica, o que significa que há autoespaços com dimensão maior que 1 e que os vetores de base da 2D-DCT não constituem a única base de  $\mathbb{R}^{N \times N}$  formada por autovetores de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$ .

## 2.2 DEFINIÇÃO DA SDCT

O fato de os vetores de base da 2D-DCT não constituírem a única base de  $\mathbb{R}^{N \times N}$  formada por autovetores de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$  sugere que novas bases formadas por autovetores do Laplaciano do referido grafo podem ser construídas. Este fato foi explorado por Fracastoro *et al.* (FRACASTORO; MAGLI, 2015; FRACASTORO; FOSSON; MAGLI, 2017) para definir a transformada discreta manobrável do cosseno. Mais especificamente, o que se faz é considerar pares de vetores de base da 2D-DCT associados a um mesmo autovalor e impor uma rotação no espaço bidimensional ao qual cada um desses pares pertence. Assim, considere os vetores de base  $\mathbf{v}^{(k,l)}$  e  $\mathbf{v}^{(l,k)}$  da 2D-DCT, os quais se encontram associados ao autovalor  $\lambda_{k,l}$  de  $\mathbf{L}(\mathcal{P}_N \square \mathcal{P}_N)$ . Respectivas versões rotacionadas  $\mathbf{v}^{(k,l)'}$  e  $\mathbf{v}^{(l,k)'}$  desses vetores são obtidas por

$$\begin{bmatrix} \mathbf{v}^{(k,l)'} \\ \mathbf{v}^{(l,k)'} \end{bmatrix} = \begin{bmatrix} \cos\theta_{k,l} & \text{sen}\theta_{k,l} \\ -\text{sen}\theta_{k,l} & \cos\theta_{k,l} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{v}^{(k,l)} \\ \mathbf{v}^{(l,k)} \end{bmatrix}, \quad (2.12)$$

em que  $\theta_{k,l}$  é um ângulo em  $[0, 2\pi]$ ; observe que, naturalmente, os vetores de base da 2D-DCT são bidimensionais, de modo que, na última equação, para que o produto matricial no lado direito da igualdade esteja coerente, supõe-se que  $\mathbf{v}^{(k,l)}$  e  $\mathbf{v}^{(l,k)}$  têm suas componentes rearranjadas de forma unidimensional, isto é, em vetores-linha com  $N^2$  componentes.

Dentre os  $N - 1$  autovetores com autovalor associado  $\lambda_{k,N-k} = 4, 2\lfloor(N - 1)/2\rfloor$  são rotacionados aos pares  $\mathbf{v}^{(k,N-k)}$  e  $\mathbf{v}^{(N-k,k)}$ ,  $k = 1, 2, \dots, \lceil N/2 \rceil - 1$ . Os autovetores com autovalor associado  $\lambda_{k,k}$ ,  $k = 0, 1, \dots, N - 1$  não são rotacionados. O efeito da rotação em (2.12) sobre os vetores de base da 2D-DCT pode ser melhor compreendido com o auxílio da Figura 4, em que estão destacados em verde os vetores de base correspondentes aos  $N - 1$  autovalores com multiplicidade algébrica 1 e em azul um daqueles correspondentes ao autovalor com multiplicidade algébrica 4. Os  $N$  vetores de base na diagonal principal são os únicos que não são rotacionados pela SDCT, os demais são rotacionados aos pares, em que um dos integrantes do par está localizado abaixo da diagonal principal e o outro acima. Em vermelho é destacado um desses pares de vetores de base correspondente a um autovalor com multiplicidade 2 e em laranja um par relacionado ao autovalor com multiplicidade algébrica 4. A transformação cuja base é o conjunto formado por todos os pares de vetores rotacionados  $\mathbf{v}^{(k,l)'} e \mathbf{v}^{(l,k)'}$ , em substituição aos respectivos pares de vetores originais  $\mathbf{v}^{(k,l)}$  e  $\mathbf{v}^{(l,k)}$ , é identificada como SDCT. O operador associado à SDCT pode ser expresso como

$$\mathbf{V}(\Theta) = \mathbf{R}(\Theta)\mathbf{V}; \quad (2.13)$$

na última equação,  $\Theta$  é um vetor com  $N(N - 1)/2$  componentes, contendo todos os ângulos empregados nas rotações dos pares de vetores, e  $\vec{\mathbf{0}}$  é o vetor nulo com mesmo comprimento;  $\mathbf{V} = \mathbf{V}(\vec{\mathbf{0}})$  é o operador associado à 2D-DCT (para manter coerência na escrita da equação matricial, pode-se assumir que  $\mathbf{V}$  é construída pela disposição, ao longo de suas linhas, de versões unidimensionais dos vetores de base da transformada ordinária, conforme já se descreveu anteriormente);  $\mathbf{R}(\Theta)$  é uma matriz de rotação esparsa, com os cossenos e senos dos ângulos de rotação tomados de  $\Theta$  dispostos segundo (2.12).

Para qualquer sinal  $\mathbf{f} \in \mathbb{R}^{N^2}$ , a SDCT é dada por

$$\hat{\mathbf{f}} = \mathbf{V}(\Theta)\mathbf{f} = \mathbf{R}(\Theta)\mathbf{V}\mathbf{f} = \mathbf{R}(\Theta)\mathbf{f}_{2\text{D-DCT}}, \quad (2.14)$$

em que  $\mathbf{f}_{2\text{D-DCT}} = \mathbf{V}\mathbf{f}$  denota a 2D-DCT de  $\mathbf{f}$ . Isso significa que as rotações impostas pela matriz  $\mathbf{R}(\Theta)$  aos vetores de base da 2D-DCT, para obtenção do operador  $\mathbf{V}(\Theta)$  da SDCT visando à sua aplicação a  $\mathbf{f}$ , podem, equivalentemente, ser impostas às componentes de  $\mathbf{f}_{2\text{D-DCT}}$ , as quais seriam inicialmente calculadas. Desse modo, dado  $\mathbf{f}_{2\text{D-DCT}}$ , a complexidade aritmética adicional para obtenção de  $\hat{\mathbf{f}}$  se deve ao produto entre  $\mathbf{R}(\Theta)$  e  $\mathbf{f}_{2\text{D-DCT}}$ , o qual envolve  $\mathcal{O}(N)$  operações de multiplicação e adição. Em suma, a complexidade aritmética total (assintótica) envolvida no cálculo de uma SDCT é a mesma envolvida no cálculo da 2D-DCT correspondente.

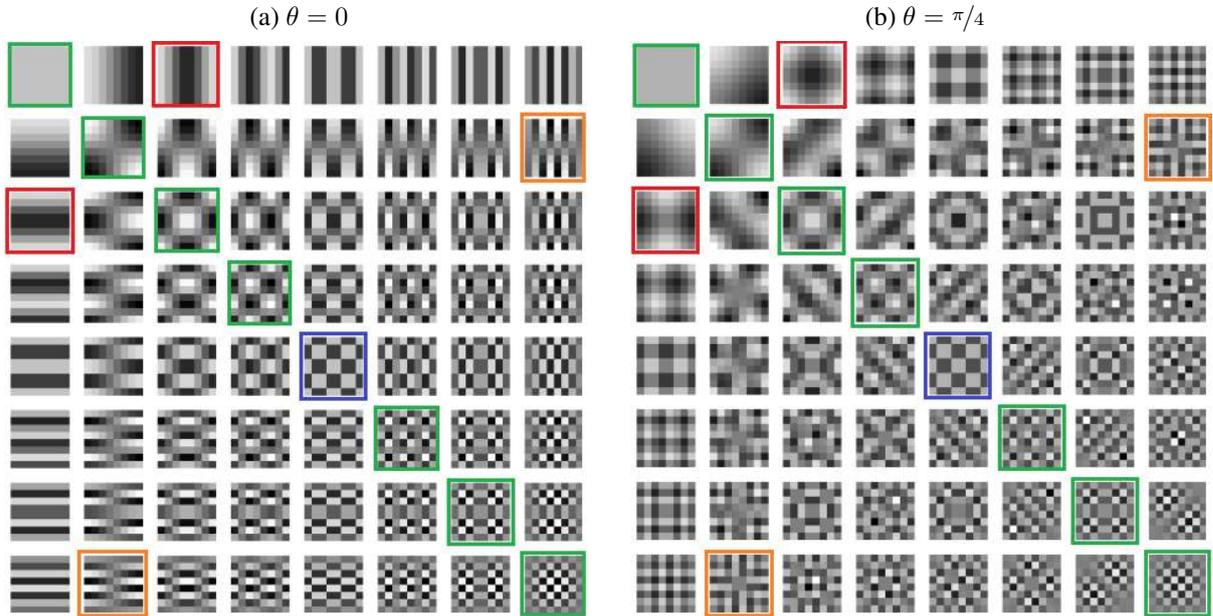
A SDCT inversa é calculada por

$$\mathbf{f} = \mathbf{V}^{-1}(\Theta)\hat{\mathbf{f}} = \mathbf{V}^{-1}\mathbf{R}^{-1}(\Theta)\hat{\mathbf{f}} = \mathbf{V}^T\mathbf{R}^T(\Theta)\hat{\mathbf{f}}.$$

### 2.2.1 Ângulo Ótimo

Conforme anteriormente indicado, os ângulos de rotação para definição de uma SDCT podem ser livremente escolhidos. No entanto, a escolha de ângulos específicos para rotacionar

Figura 4 – Em representação matricial, para  $N = 8$ , são mostrados os vetores de base da 2D-DCT (a) e da SDCT para uma rotação de  $\pi/2$  radianos (b).



Fonte: Baseado em Fracastoro, Fosson e Magli (2017).

cada par de coeficientes da 2D-DCT pode levar à anulação de um desses dois coeficientes no domínio da SDCT. Mais precisamente, considere que o coeficiente na posição  $(k, l)$  no domínio da 2D-DCT de  $f$  é denotado por  $c_{k,l}$ ; este coeficiente é obtido pela projeção de  $f$  no subespaço vetorial relacionado ao vetor de base  $\mathbf{v}^{(k,l)}$ . Utilizando o ângulo

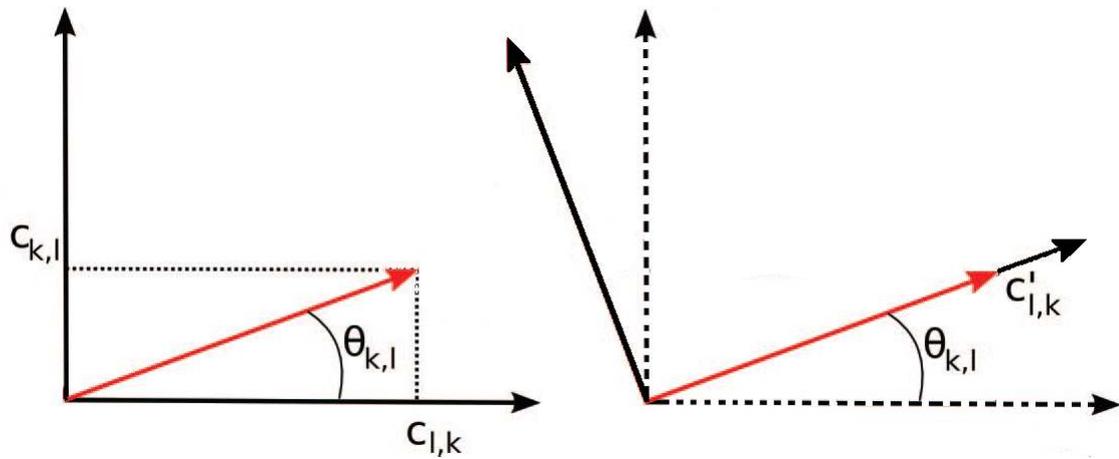
$$\theta_{k,l}^{\text{opt.}} = \arctan\left(\frac{c_{k,l}}{c_{l,k}}\right) \quad (2.15)$$

para rotacionar o par de coeficientes  $c_{k,l}$  e  $c_{l,k}$  (ou o par de vetores de base  $\mathbf{v}^{(k,l)}$  e  $\mathbf{v}^{(l,k)}$ ),  $k = 1, 2, \dots, N - 1$ ,  $l = 0, 1, \dots, k - 1$ , é possível verificar que os respectivos coeficientes no domínio da SDCT são dados por  $c'_{k,l} = 0$  e  $c'_{l,k} \neq 0$  (tem-se  $c'_{k,l} = c'_{l,k} = 0$  apenas quando  $c_{k,l} = c_{l,k} = 0$ ). O ângulo dado em (2.15) é denominado ângulo ótimo.

O efeito da rotação pelo ângulo ótimo pode ser melhor compreendido com o auxílio da Figura 5, em que o vetor na cor vermelha representa a projeção ortogonal do sinal  $f$  no espaço bidimensional gerado pelos dois autovetores que são representados na figura utilizando a cor preta. Ao rotacionar o par de autovetores utilizando o ângulo  $\theta_{k,l} = \theta_{k,l}^{\text{opt.}}$ , obtém-se o efeito de fazer coincidir a orientação de um dos autovetores com a orientação de  $f$  no autoespaço correspondente ao par.

Claramente, se cada par de coeficientes (ou de vetores) da 2D-DCT for rotacionado empregando o respectivo ângulo ótimo, a estrutura bidimensional  $\hat{f}$  corresponderá a uma matriz triangular inferior com dimensões  $N \times N$ . Assim, o uso de ângulos ótimos para definir uma SDCT sugere que essa transformada é capaz de proporcionar maior compactação de energia do que a 2D-DCT: além da concentração dos coeficientes com maior magnitude nas baixas

Figura 5 – Rotação por um ângulo ótimo.



Fonte: Baseado em (FRACASTORO; FOSSON; MAGLI, 2017)

frequências (canto superior esquerdo de  $\hat{f}$ ), o que também se consegue com o uso da transformada ordinária, tem-se que todos os coeficientes acima da diagonal principal de  $\hat{f}$  seriam iguais a zero. No próximo capítulo, explica-se como essa possibilidade tem sido explorada para realizar compressão de imagens digitais.

### 3 CODIFICAÇÃO DE IMAGEM BASEADA NA SDCT

Na última seção do Capítulo 2, foi apresentada a transformada discreta manobrável do cosseno. Em particular, foi discutida a possibilidade de aplicação dessa transformada empregando os chamados ângulos ótimos; isso faria com que o sinal bidimensional  $\mathbf{f}$ , com dimensões  $N \times N$ , fosse representado, no domínio da SDCT, por um sinal bidimensional  $\hat{\mathbf{f}}$ , com as mesmas dimensões, correspondente a uma matriz triangular inferior. A ideia mais natural seria explorar a compactação da energia dos elementos de  $\hat{\mathbf{f}}$  e a sua esparsidade para realizar uma compressão do sinal. Se as estruturas bidimensionais em questão corresponderem a blocos de uma imagem digital, poder-se-ia pensar em comprimir tais blocos e obter um codificador com desempenho melhor que aquele alcançado pelo uso da 2D-DCT no que diz respeito à eficiência de codificação. O fato é que, embora o uso da SDCT com ângulos ótimos leve à anulação de  $N(N - 1)/2$  coeficientes (que não precisariam ser informados ao decodificador) dentre os  $N^2$  coeficientes que compõem  $\hat{\mathbf{f}}_{2D-DCT}$  ou  $\hat{\mathbf{f}}$ , para recuperação de  $\mathbf{f}$ , seria necessário informar (ao decodificador) os  $N(N - 1)/2$  ângulos ótimos empregados nas rotações e nenhum ganho efetivo seria provido.

Pelo motivo descrito acima, o uso da SDCT para realizar compressão de imagens tem empregado estratégias sub-ótimas. Essas estratégias incluem soluções menos esparsas, isto é, em que um menor número de coeficientes é anulado a cada bloco de imagem, e soluções não-exatamente esparsas, isto é, em que muitos coeficientes são próximos de zero, mas não são exatamente zero. A sugestão é que soluções desse tipo produzam *tradeoffs* interessantes quando se compara o custo de se transmitir ao decodificador mais informações de controle (predominantemente, os valores dos ângulos de rotação utilizados) com o benefício provido por uma representação mais compacta do sinal. Essa possibilidade pode ser analisada a partir de uma perspectiva de taxa-distorção (RD, do inglês *rate-distortion*), conforme descrito na seção a seguir. Nas demais seções deste capítulo, são revisados os algoritmos e esquemas para codificação de imagens baseada na SDCT que têm sido propostos na literatura.

#### 3.1 MODELO RD

Conforme visto, para determinado bloco de imagem digital  $\mathbf{f}$  ( $N \times N$ ), a SDCT pode ser interpretada como uma versão parametrizável da DCT bidimensional. Esta parametrização ocorre por meio da escolha de  $P = N(N - 1)/2$  ângulos de rotação  $\theta_{k,l}$ , que são valores reais pertencentes ao intervalo semiaberto  $[0, 2\pi)$ . Estes ângulos podem ser agrupados, empregando alguma ordenação pré-definida  $\text{ord}(k, l) = \text{ord}(l, k) = i, i \in \{0, 1, \dots, P - 1\}$ , em um vetor unidimensional  $\Theta = (\theta_0, \theta_1, \dots, \theta_{P-1})$ . Para cada escolha de  $\Theta$ , obtém-se uma versão diferente  $\hat{\mathbf{f}}$  para os coeficientes da SDCT, apresentando diferentes graus de esparsidade. Tanto os elementos reais de  $\hat{\mathbf{f}}$  como os de  $\Theta$  podem ser quantizados respectivamente nos conjuntos finitos de valores reais  $\mathbf{Q}_{\hat{\mathbf{f}}} \subset \mathbb{R}$  e  $\mathbf{Q}_{\theta} \subset [0, 2\pi)$ . A versão quantizada dos coeficientes será denotada por

$\hat{\mathbf{f}}_Q$ , enquanto a notação  $\Theta$ , deste ponto em diante, será utilizada para a versão quantizada dos ângulos.

A escolha dos ângulos de rotação influencia na quantidade necessária de bits para codificar o bloco de imagem de forma direta, devido à quantidade de ângulos diferentes que precisarão ser informados ao decodificador e à quantidade de bits necessária para codificar cada um destes ângulos. Tal escolha também atua de forma indireta por meio de alterações provocadas na esparsidade de  $\hat{\mathbf{f}}$ . Além disso, para mapeamentos pré-determinados de quantização  $Q(\hat{\mathbf{f}}) : \mathbb{R}^{N^2} \rightarrow \mathbf{Q}_{\hat{\mathbf{f}}}^{N^2}$  e  $Q(\Theta) : [0, 2\pi)^P \rightarrow \mathbf{Q}_{\Theta}^P$ , as escolhas dos ângulos afetam a distorção, em relação ao bloco original  $\mathbf{f}$ , do bloco reconstruído  $\mathbf{f}^*$  após aplicação da SDCT inversa  $V(\Theta)^T$ . Neste ponto, percebe-se claramente o problema central no projeto de um codificador baseado na SDCT: a escolha de um conjunto adequado de ângulos que entregue o *tradeoff* desejado entre taxa e distorção (problema RD).

Como medida de distorção, Fracastoro, Fosson e Magli (2017) propõem o uso do erro de reconstrução dado por:

$$\mathcal{D}(\hat{\mathbf{f}}, \Theta) := \|\mathbf{f} - \mathbf{f}^*\|_2^2 = \|\mathbf{f} - V^T(\Theta)\hat{\mathbf{f}}_Q\|_2^2. \quad (3.1)$$

Para composição da taxa de bits  $\mathcal{R}$  considera-se dois fatores, sendo um referente aos coeficientes da transformada ( $\mathcal{R}_{\hat{\mathbf{f}}}$ ) e outro referente aos ângulos de rotação ( $\mathcal{R}_{\theta}$ ):

$$\mathcal{R}(\hat{\mathbf{f}}, \Theta) = \mathcal{R}_{\hat{\mathbf{f}}} + \mathcal{R}_{\theta}. \quad (3.2)$$

Para transformadas DCT, existe uma relação aproximadamente linear entre a taxa  $\mathcal{R}_{\hat{\mathbf{f}}}$  e a norma  $\ell_0$  de  $\hat{\mathbf{f}}$  (KIM; HE; MITRA, 2001 apud FRACASTORO; FOSSON; MAGLI, 2017), em que uma constante de proporcionalidade  $\alpha$  pode ser encontrada empiricamente, ou seja:

$$\mathcal{R}_{\hat{\mathbf{f}}} = \alpha \|\hat{\mathbf{f}}\|_0. \quad (3.3)$$

Já a taxa devido aos ângulos ( $\mathcal{R}_{\theta}$ ) precisa ser tratada de acordo com cada caso, conforme é visto nas próximas subseções.

O objetivo no problema RD é minimizar a distorção  $\mathcal{D}$ , porém sujeitando-se a uma restrição sobre a taxa  $\mathcal{R}$ , ou seja:

$$\min(\mathcal{D}), \text{ sujeito a } \mathcal{R} < \mathcal{R}_{\lambda}. \quad (3.4)$$

A tarefa de otimização descrita por (3.4) pode ser resolvida através do método dos multiplicadores de Lagrange (SULLIVAN; WIEGAND et al., 1998, p. 81):

$$\min(J), \text{ em que } J = \mathcal{D} + \lambda \mathcal{R}. \quad (3.5)$$

A solução de (3.5) sob um valor dado para o multiplicador  $\lambda$  corresponde a uma solução ótima sob o ponto de vista da distorção sujeita a uma determinada taxa  $\mathcal{R}_{\lambda}$  (3.4). Como a taxa final obtida

no processo de compressão está em grande medida relacionada ao passo de quantização utilizado para quantizar os coeficientes e ângulos, a escolha do fator  $\lambda$  deve levar em consideração o nível de quantização utilizado pelo codificador. Seja  $Q$  o passo de quantização utilizado, Sullivan, Wiegand et al. (1998, p. 86) levantam empiricamente que um fator  $\lambda$  ótimo é proporcional a  $Q^2$ .

### 3.2 MINIMIZAÇÃO ALTERNADA: SDCT-AM

Para modelar o problema da taxa referente aos ângulos ( $\mathcal{R}_\theta$ ), Fracastoro, Fosson e Magli (2017) propõem a sua divisão em grupos de acordo com os pares de vetores de base da 2D-DCT que sofrerão a mesma rotação, de forma que o vetor  $\Theta$  torna-se constante por partes. Seja  $s$  o número de tais grupos. Se  $s$  for menor que  $P$ , pode-se transmitir ao decodificador apenas  $s$  ângulos além de  $s$  valores indicando onde cada grupo termina. Então, assumindo que, em princípio, não há compressão para os ângulos e que são quantizados utilizando  $q_\theta = \text{card}(Q_\theta)$  valores por ângulo, o custo de transmissão de  $\Theta$  (em bits) pode ser calculado por

$$\mathcal{R}_\theta = s \lceil \log_2 q_\theta \rceil + s \lceil \log_2 P \rceil = \|\mathbf{B}\Theta\|_0 (\lceil \log_2 q_\theta \rceil + \lceil \log_2 P \rceil), \quad (3.6)$$

em que  $B \in \mathbb{R}^{P \times P}$  é o operador matricial de diferença dado por

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ -1 & 1 & 0 & \dots & 0 \\ 0 & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & \dots & -1 & 1 \end{bmatrix}, \quad (3.7)$$

de forma que a função lagrangeana  $J$  no problema de minimização em (3.5) se torna

$$\begin{aligned} J(\hat{\mathbf{f}}, \Theta) &= \mathcal{D}(\hat{\mathbf{f}}, \Theta) + \lambda(\mathcal{R}_{\hat{\mathbf{f}}} + \mathcal{R}_\theta) \\ &= \|\mathbf{f} - \mathbf{V}^T(\Theta)\hat{\mathbf{f}}_Q\|_2^2 + \lambda \left[ \alpha \|\hat{\mathbf{f}}_Q\|_0 + \|\mathbf{B}\Theta\|_0 (\lceil \log_2 q_\theta \rceil + \lceil \log_2 P \rceil) \right]. \end{aligned} \quad (3.8)$$

Técnicas de *hard thresholding* podem ser utilizadas para resolver o problema de minimização da equação 3.8 em relação aos coeficientes quantizados  $\hat{\mathbf{f}}$  (BLUMENSATH; DAVIES, 2008 apud FRACASTORO; FOSSON; MAGLI, 2017). De forma que  $\|\mathbf{f} - \mathbf{V}^T(\Theta)\hat{\mathbf{f}}_Q\|_2^2 + \lambda\alpha\|\hat{\mathbf{f}}_Q\|_0$  terá mínimo global em  $\hat{\mathbf{f}}_Q^{\text{opt}} = \mathcal{H}_{\sqrt{\lambda\alpha}}[\mathbf{V}(\Theta)\mathbf{f}]$  (FRACASTORO; FOSSON; MAGLI, 2017, p. 309), em que  $\mathcal{H}_{\sqrt{\lambda\alpha}} : \mathbb{R}^{N^2} \rightarrow \mathbb{R}^{N^2}$  é dado por:

$$\mathcal{H}_{\sqrt{\lambda\alpha}}[\mathbf{x}](i) = \begin{cases} 0, & \text{se } |x(i)| \leq \sqrt{\lambda\alpha}, \\ x(i), & \text{caso contrário;} \end{cases} \quad i = 0, 1, \dots, N^2 - 1. \quad (3.9)$$

Entretanto, esta solução leva em consideração apenas a contribuição da variável  $\hat{\mathbf{f}}_Q$ , considerada enquanto variável dependente do mapeamento realizado pelo quantizador  $Q : \mathbb{R}^{N^2} \rightarrow Q_{\hat{\mathbf{f}}}^{N^2}$ . Há que se considerar ainda a contribuição da variável independente  $\Theta$ . Fracastoro, Fosson e Magli (2017) propõem então o uso de minimização alternada nas variáveis isoladas como forma de obter uma solução parcial analítica para (3.5).

Inicialmente, assume-se  $\Theta$  fixo, de forma a utilizar a técnica de *hard thresholding* para obter uma solução de  $\min_{\hat{\mathbf{f}}_Q \in Q_{\hat{\mathbf{f}}}^{N^2}} J(\hat{\mathbf{f}}, \Theta)$ . Em seguida realiza-se minimização em relação à variável  $\Theta = [\theta_0, \theta_1, \dots, \theta_{P-1}]$ , feita separadamente para cada escalar  $\theta_i$ :

$$\min_{\theta_i \in Q_{\theta}} J(\hat{\mathbf{f}}, \Theta) = \min_{\theta_i \in Q_{\theta}} \left\{ \|\mathbf{f} - \mathbf{V}^T(\Theta)\hat{\mathbf{f}}_Q\|_2^2 + \lambda \|\mathbf{B}\Theta\|_0 (\lceil \log_2 q_{\theta} \rceil + \lceil \log_2 P \rceil) \right\}. \quad (3.10)$$

Em que o termo  $\|\mathbf{B}\Theta\|_0$  pode ser substituído por

$$\|\mathbf{B}\Theta\|_0 = \begin{cases} \|\theta_i - \theta_{i+1}\|_0 + \|\theta_i - \theta_{i-1}\|_0, & \text{se } i \in \{2, \dots, P-1\}, \\ \|\theta_1\|_0 + \|\theta_1 - \theta_2\|_0, & \text{se } i = 1, \\ \|\theta_P - \theta_{P-1}\|_0, & \text{se } i = P. \end{cases} \quad (3.11)$$

Para minimizar a parcela referente à distorção  $D(\hat{\mathbf{f}}, \Theta)$ , o primeiro passo é expandir a expressão em (3.1). Como  $V(\Theta)$  é ortogonal para qualquer  $\Theta$ :

$$\begin{aligned} \|\mathbf{f} - V^T(\Theta)\hat{\mathbf{f}}_Q\|_2^2 &= \|\mathbf{f}\|_2^2 - 2\mathbf{f}^T V^T(\Theta)\hat{\mathbf{f}}_Q + \|\hat{\mathbf{f}}_Q\|_2^2 \\ &= \|\mathbf{f}\|_2^2 - 2\mathbf{f}^T R^T(\Theta)V^T\hat{\mathbf{f}}_Q + \|\hat{\mathbf{f}}_Q\|_2^2 \\ &= \|\mathbf{f}\|_2^2 - 2\mathbf{f}^T \left[ \Delta + \tilde{R}^T(\Theta) \right] V^T\hat{\mathbf{f}}_Q + \|\hat{\mathbf{f}}_Q\|_2^2. \end{aligned} \quad (3.12)$$

Em (3.12),  $\mathbf{R}(\Theta)$  é decomposta em duas matrizes:  $\Delta \in \mathbb{R}^{N^2 \times N^2}$  é uma matriz diagonal correspondente aos autovetores que não sofrem rotação, enquanto  $\tilde{R}(\Theta) \in \mathbb{R}^{N^2 \times N^2}$  é uma matriz correspondente aos autovetores que são rotacionados. Fracastoro, Fosson e Magli (2017) definem ainda uma matriz  $\mathbf{W} = \mathbf{W}(\hat{\mathbf{f}}_Q) \in \mathbb{R}^{N^2 \times 2P}$ , tal que

$$\mathbf{V}\tilde{\mathbf{R}}(\Theta)\hat{\mathbf{f}}_Q = \mathbf{W}(\hat{\mathbf{f}}_Q) [\cos(\theta_0), \text{sen}(\theta_0), \dots, \cos(\theta_{P-1}), \text{sen}(\theta_{P-1})]^T.$$

Desta forma, o problema de encontrar o argumento que minimiza a expressão de distorção em (3.12) torna-se um problema mais compacto e analiticamente tratável:

$$\min_{\theta_i \in Q_{\theta}} D(\hat{\mathbf{f}}, \Theta) = \min_{\theta_i \in Q_{\theta}} -2\mathbf{f}^T \mathbf{W}(\hat{\mathbf{f}}_Q) (\cos(\theta_0), \text{sen}(\theta_0), \dots, \cos(\theta_{P-1}), \text{sen}(\theta_{P-1}))^T. \quad (3.13)$$

Este problema pode ser resolvido tomando a derivada parcial e igualando a zero:

$$\frac{\partial D(\hat{\mathbf{f}}, \Theta)}{\partial \Theta} = 2\mathbf{f}^T \mathbf{U}(\Theta) = \vec{\mathbf{0}}, \quad (3.14)$$

em que:  $\mathbf{U}^{(2i)} = \mathbf{W}^{(2i)} \text{sen}(\theta_i)$  e  $\mathbf{U}^{(2i+1)} = -\mathbf{W}^{(2i+1)} \cos(\theta_i)$  para  $i = 0, 1, \dots, P-1$ .

Minimizando para cada uma das variáveis  $\theta_i$  ficamos com

$$\begin{aligned} \frac{\partial D(\hat{\mathbf{f}}, \theta_i)}{\partial \theta_i} &= 2\mathbf{f}^T [\mathbf{U}^{(2i)} \text{sen}(\theta_i) + \mathbf{U}^{(2i+1)} \cos(\theta_i)] = \vec{\mathbf{0}} \\ \frac{\text{sen}(\theta_i)}{\cos(\theta_i)} &= \frac{\mathbf{f}^T \mathbf{U}^{(2i+1)}}{\mathbf{f}^T \mathbf{U}^{(2i)}} \\ \theta_i^{\text{opt-AM}} &= \arctan \left( \frac{\mathbf{f}^T \mathbf{U}^{(2i+1)}}{\mathbf{f}^T \mathbf{U}^{(2i)}} \right). \end{aligned} \quad (3.15)$$

O arco-tangente possui uma solução em  $[0, \pi)$ , que poderá ser um ponto de máximo ou um ponto de mínimo da distorção em relação à variável  $\theta_i$ . Compara-se então o valor desta distorção com os pontos extremos  $\theta_i = 0$  e  $\theta_i = \pi$  para verificar se é realmente um ponto de mínimo. Por fim, a solução deve ser mapeada pelo quantizador para um valor em  $Q_\theta$ .

Após encontradas as soluções ótimas para cada um dos ângulos com base na minimização da distorção, conforme (3.15), é realizada a tarefa de minimização voltada para  $\mathcal{R}_\theta$  que, seguindo o algoritmo SDCT-AM proposto por Fracastoro, Fosson e Magli (2017), consiste basicamente em escolher, entre duas ou três possibilidades, qual é a escolha mais adequada (no sentido de minimizar  $J$ ) para cada  $\theta_i$ . As três decisões possíveis são basicamente as seguintes:

1.  $\theta_i = \theta_i^{opt-AM}$ ;
2.  $\theta_i = \theta_{i-1}^{opt-AM}$ , se  $i > 0$ , ou  $\theta_i = 0$ , se  $i = 0$ ;
3. ou  $\theta_i = \theta_{i+1}^{opt-AM}$  se  $i < P - 1$ .

Em suma, o método de minimizações alternadas da SDCT-AM envolve o emprego de três minimizações independentes, sendo uma sobre a variável  $\hat{\mathbf{f}}_Q = Q(\hat{\mathbf{f}})$  (3.9) e duas sobre a variável  $\Theta$ . Em relação aos ângulos, uma primeira otimização é focada na redução da distorção (encontrando os ângulos ótimos  $\theta_i^{opt-AM}$ ) e uma segunda é focada no fator de taxa relativa aos ângulos (variando a quantidade  $s$  de conjuntos de ângulos distintos). Todo o processo é repetido iterativamente até que o novo valor encontrado para a função lagrangiana torne-se estável ou um número máximo de iterações seja realizado: o que ocorrer primeiro. Seja  $t$  o número da iteração e  $\hat{\mathbf{f}}_Q(t)$  e  $\Theta(t)$  os valores das variáveis obtidos ao final da mesma; de Fracastoro, Fosson e Magli (2017, Teorema 2), garante-se que existirá uma iteração  $t_0$  na qual  $J(\hat{\mathbf{f}}_Q(t), \Theta(t))$  da SDCT-AM estabilizará em um ponto de mínimo local.

### 3.3 UTILIZANDO UM ÂNGULO ÚNICO: SDCT-1

Para simplificar o problema de otimização de taxa e distorção na compressão de imagens utilizando a SDCT, Fracastoro e Magli (2015) propõem que uma restrição pode ser imposta sobre  $\Theta$  de forma que um único ângulo  $\theta_{SDCT1}$  seja utilizado para rotacionar todos os  $P$  pares de autovetores e ainda assim resultados melhores que os da DCT podem ser obtidos em termos de RD. Como apenas um ângulo de rotação precisa ser transmitido por bloco, o custo  $\mathcal{R}_\theta$  passa a ser simplesmente  $\lceil \log_2 q_\theta \rceil$  bits, em que  $q_\theta = \text{card}(Q_\theta)$ .

No algoritmo da SDCT-1, uma busca exaustiva do melhor resultado para otimizar uma função de custo  $J$  é realizada entre todas as possibilidades de núcleo para a SDCT-1:

$$\mathbf{V}_{SDCT1}(\theta^{SDCT1}) = \mathbf{V}(\Theta),$$

em que  $\Theta = [\theta_0, \theta_1, \dots, \theta_{P-1}]$  e  $\theta_0 = \theta_1 = \dots = \theta_{P-1} = \theta_{SDCT1}$ ,  $\theta_{SDCT1} \in Q_\theta$ .

Fracastoro e Magli (2015) propõem a utilização de uma forma de recuperar uma versão aproximada da imagem original com base nos  $M$  maiores coeficientes dentre os  $N^2$  obtidos após aplicação da SDCT sobre um bloco de imagem  $f$  de tamanho  $N \times N$ . Sejam os coeficientes  $\hat{f} = \mathbf{V}_{SDCT1}(\theta_{SDCT1})f$ . Então, uma versão aproximada  $\hat{f}^*$  de  $\hat{f}$  é obtida fazendo com que os  $N^2 - M$  menores coeficientes sejam aproximados por zero. Uma versão aproximada do bloco de imagem  $f$  pode ser então obtida como  $f^* = \mathbf{V}_{SDCT1}^T(\theta_{SDCT1})\hat{f}^*$ . Para encontrar o ângulo de rotação ótimo, é realizada uma busca exaustiva entre todos os ângulos de rotação possíveis de forma que uma função de custo dada por  $J = \|\hat{f}^*\|_2^2$  seja minimizada.

A distorção em termos relação sinal-ruído de pico, também expressa como PSNR (*Peak Signal-to-Noise Ratio*), é então calculada para a aproximação  $f^*$  em relação ao bloco de imagem original e comparada com a distorção obtida quando a mesma aproximação é obtida utilizando a transformada 2D-DCT usual. Os valores de distorção são comparados para o mesmo número  $M$  de coeficientes retidos entre as duas transformadas.

Para obtenção de um conjunto finito de ângulos  $Q_\theta$ , Fracastoro e Magli (2015) propõem uma quantização uniforme entre  $0^\circ$  e  $90^\circ$ . Resultados foram obtidos utilizando 4, 16, e 128 níveis de quantização para os ângulos. Eles foram comparados com os resultados da 2D-DCT usual, indicando ganhos de até 1,5 dB. Os resultados da SDCT-1 também foram comparados pelos autores com os resultados da transformada discreta direcionável do cosseno (DDCT) de Zeng e Fu (2008), obtendo desempenho numérico levemente superior (0,1 dB médio).

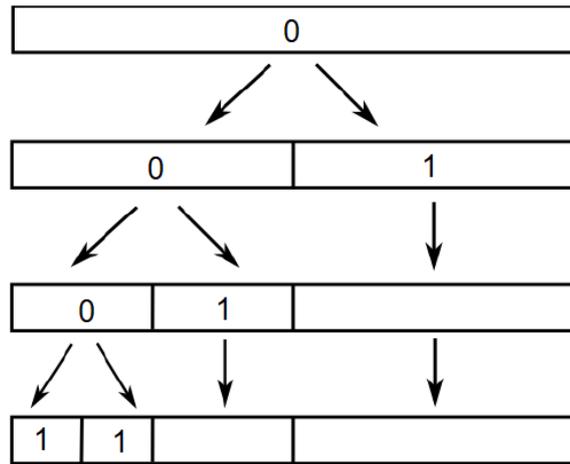
Apesar dos resultados positivos, a comparação feita em termos de número de coeficientes retidos não é exatamente justa, pois não necessariamente corresponderá aos resultados obtidos se as transformadas forem utilizadas em um CODEC de compressão de imagens e forem comparados em termos de taxa de bits. Isso foi corrigido pelos autores originais em Fracastoro, Fosson e Magli (2017) quando fizeram a comparação dos resultados da SDCT-1 no contexto da apresentação dos novos algoritmos SDCT-AM e SDCT-BT.

Neste caso, a função  $J$  foi substituída pela lagrangeana em (3.5) e a aproximação baseada nos  $M$  maiores coeficientes foi substituída por uma aproximação com base em quantização uniforme dos coeficientes  $\hat{f}$  utilizando um passo único de quantização  $Q$ , de forma semelhante ao que foi proposto para as demais transformadas do artigo. Neste caso, entretanto, o ângulo foi quantizado uniformemente entre  $0^\circ$  e  $180^\circ$ . Os resultados apresentados neste segundo artigo mantiveram-se positivos, com a SDCT-1 superando a 2D-DCT convencional, porém com resultados abaixo dos obtidos para os algoritmos mais complexos SDCT-AM e SDCT-BT, que utilizam mais de um ângulo por bloco.

### 3.4 ÁRVORE BINÁRIA PARA ESTRUTURA DE ÂNGULOS: SDCT-BT

Fracastoro, Fosson e Magli (2017) propõem um algoritmo que permite a utilização de buscas exaustivas ao longo de todos os  $q_\theta$  valores possíveis de ângulos quantizados, de modo

Figura 6 – Exemplo de árvore binária de decisões na SDCT-BT.



Fonte: Baseado em Fracastoro, Fosson e Magli (2017, Figura 6).

semelhante ao que é realizado no algoritmo da SDCT-1, ao mesmo tempo em que permite a utilização de mais de um ângulo por bloco. Chamado de SDCT-BT, utiliza uma estrutura de árvore binária para representação da distribuição dos diferentes valores de ângulos ao longo dos pares de autovetores a serem rotacionados.

Ao contrário do que é feito no caso da minimização alternada da SDCT-AM, na SDCT-BT é entendido que exista apenas uma variável independente  $\Theta$ . A cada escolha de  $\Theta$ ,  $\hat{f}_Q$  fica automaticamente determinado como  $\hat{f}_Q = Q[\mathbf{V}(\Theta)\mathbf{f}]$ , em que  $Q$  indica a operação de quantização em  $Q_{\hat{f}}$ . A função lagrangeana RD passa a ser então

$$J(\Theta) = \mathcal{D}(\Theta) + \lambda(\mathcal{R}_{\hat{f}} + \mathcal{R}_{\theta}). \quad (3.16)$$

A escolha dos ângulos na SDCT-BT através de busca exaustiva voltada à minimização de  $J$  ocorre de forma recursiva. Inicia-se com apenas um único ângulo para todos os pares de autovetores (uma única sub-banda) e o menor valor encontrado para  $J$  é registrado. Iterativamente, cada sub-banda disponível é dividida experimentalmente em duas novas com aproximadamente o mesmo tamanho. Uma nova busca é realizada dentre todas as  $q_{\theta}^2$  possibilidades de escolhas para a combinação de ângulos em cada uma das novas sub-bandas. Caso o novo valor ótimo encontrado para  $J$  seja melhor do que o anterior, a divisão é aceita e, em seguida, duas novas sub-divisões experimentais são testadas e a busca prossegue. Caso contrário, a divisão não é aceita e o algoritmo é encerrado.

A Figura 6 mostra um possível resultado do processo iterativo de decisões sobre a divisão em novas sub-bandas para a SDCT-BT. Neste exemplo, a primeira iteração do algoritmo encontrou uma função  $J$  melhor com duas sub-bandas ao invés de uma e, portanto, esta divisão foi aceita. Em seguida, o algoritmo foi novamente executado para cada uma das sub-bandas. Na sub-banda da esquerda, uma nova divisão foi aceita, enquanto na sub-banda da direita o processo foi encerrado sem nova sub-divisão. Após nova iteração, nova sub-divisão ocorre na

primeira faixa de ângulos, resultando em uma escolha final para  $\Theta$  composta por quatro valores de ângulos.

A sub-estrutura dos ângulos da Figura 6 é então codificada como uma árvore binária em formato vetorial. Cada nó da árvore é representado por um bit, indicando se houve divisão (bit 0) ou não (bit 1). São necessários então um total de sete bits para codificar corretamente a subestrutura de ângulos da Figura 6:  $\mathbf{S} = [0, 0, 1, 0, 1, 1, 1]$ . Para  $N = 8$ , a mesma subestrutura de ângulos necessitaria de  $s \lceil \log_2 P \rceil = 20$  bits se fosse utilizada a mesma estratégia da SDCT-AM. De forma geral, se  $s$  denotar o número final de ângulos distintos, tem-se

$$\mathcal{R}_\theta = s \lceil \log_2(q_\theta) \rceil + 2s - 1 = s [2 + \lceil \log_2(q_\theta) \rceil] - 1. \quad (3.17)$$

Para o cálculo da função objetivo (3.16), a distorção é calculada utilizando (3.1) para cada novo valor de  $\hat{\mathbf{f}}(\Theta)$  encontrado pelo algoritmo. A quantização é uniforme para os ângulos em  $q_\theta$  valores distribuídos entre  $0^\circ$  e  $180^\circ$ . A taxa real de bits (experimentalmente obtida) é utilizada para o custo relativo aos coeficientes  $\mathcal{R}_{\hat{\mathbf{f}}}$ .

### 3.5 ESPARSIFICAÇÃO DE SUBESPAÇOS: $S^3$ DCT

Fracastoro e Magli (2016) propõem ainda uma forma de utilizar a SDCT para compressão de imagens a partir da utilização de tantos ângulos quantos forem necessários para concentrar a energia em cada par de coeficientes rotacionados no coeficiente localizado na matriz triangular inferior de  $\hat{\mathbf{f}}$ . Seria justamente o caso da utilização dos ângulos ótimos apresentados na Subseção 2.2.1 (página 27).

Desta forma, para cada par de coeficientes  $c_{k,l}$  e  $c_{l,k}$ , um ângulo de rotação  $\theta_{k,l}^{opt}$  dado por  $\arctan(c_{k,l}/c_{l,k})$  é utilizado, de modo que os coeficientes correspondentes da  $S^3$ DCT obtidos são  $c'_{k,l} = 0$  e  $c'_{l,k} \neq 0$ . Entretanto, para evitar que os ganhos obtidos sejam anulados pela necessidade de informar ao decodificador os  $N(N-1)/2$  ângulos necessários, estes ângulos são quantizados, utilizando alguns poucos  $q_\theta$  valores uniformemente distribuídos em  $[-\pi/2, \pi/2]$ . Assim, o par de coeficientes rotacionados pela  $S^3$ DCT se torna  $c_{k,l}^{S^3DCT} \approx 0$  e  $c_{l,k}^{S^3DCT} \neq 0$ .

Apesar de  $c_{k,l}$  não ser garantidamente nulo, esta aproximação é assumida pelo codificador, de forma que apenas  $N(N+1)/2$  coeficientes são transmitidos para o decodificador, além de no máximo  $N(N-1)/2$  ângulos quantizados em  $q_\theta$  valores. Esta quantidade de ângulos a ser transmitida pode ser reduzida já que os mesmos têm significado apenas para os casos em que  $Q[c_{l,k}] \neq 0$ . Quando o quantizador mapeia este coeficiente para o valor zero, o ângulo correspondente não precisa ser transmitido.

Os autores propõem ainda a utilização de um bit adicional para o bloco indicando os casos em que a  $S^3$ DCT foi efetivamente utilizada. Para os casos em que a lagrangeana de taxa e distorção ( $J = D + \lambda \mathcal{R}$ ) for melhor utilizando a 2D-DCT usual, este bit é utilizado para informar ao decodificador que a transformada usual deverá ser utilizada.

### 3.6 CODEC DE COMPRESSÃO DE IMAGENS BASEADO NA SDCT

No caso das SDCT-AM e SDCT-BT, é necessário codificar e transmitir três tipos diferentes de informações para que se possa recuperar no decodificador uma versão aproximada da imagem original: os coeficientes da transformada, os ângulos de rotação e a estrutura de distribuição dos ângulos entre os pares de coeficientes.

Em relação aos coeficientes, Fracastoro, Fosson e Magli (2017) utilizam uma quantização uniforme através de um passo de quantização único por imagem. Quanto maior o passo, menor a taxa de bits na imagem codificada. O valor deste passo é utilizado ainda para obter o valor adequado do multiplicador de Lagrange  $\lambda$  utilizado na otimização RD. Os coeficientes passam em seguida por uma codificação aritmética binária adaptativa por planos de bit.

Os valores dos ângulos são codificados utilizando  $q_\theta = 8$  valores uniformemente quantizados no intervalo  $[0, \pi]$ , de forma que são necessários 3 bits por ângulo. Já a distribuição dos ângulos entre as sub-bandas é codificada de acordo com as especificidades de cada um dos dois algoritmos, conforme detalhado nas seções anteriores deste capítulo. Nenhuma compressão adicional é utilizada para os ângulos ou para sua distribuição.

Já no caso das SDCT-1 (FRACASTORO; MAGLI, 2015) e  $S^3$ DCT (FRACASTORO; MAGLI, 2016), não há necessidade de transmissão de informações sobre a estrutura dos ângulos, mas apenas seus valores, além dos valores dos coeficientes. Os coeficientes da SDCT-1 sofrem o mesmo tipo de quantização utilizado para a SDCT-AM e SDCT-BT. Já os coeficientes da  $S^3$ DCT são quantizados utilizando a matriz de quantização padrão JPEG (PENNEBAKER, 1993), em que passos de quantização diferentes são utilizados a depender o índice do coeficiente. Após quantização, os coeficientes são codificados utilizando a mesma codificação aritmética binária adaptativa por planos de bit utilizada no caso das SDCT-AM e SDCT-BT.

Para a  $S^3$ DCT, os ângulos são quantizados utilizando  $q_\theta = 8$  valores uniformemente quantizados no intervalo  $[-\pi/2, \pi/2]$  (3 bits por ângulo), enquanto a SDCT-1 utiliza a mesma distribuição de ângulos das SDCT-AM e SDCT-BT. Os ângulos em ambos os casos são transmitidos sem compressão adicional.

Por fim, em todas as transformadas, um bit adicional é considerado para informar ao decodificador se determinado bloco da imagem foi codificado utilizando a SDCT ou a 2D-DCT usual. Métricas objetivas de distorção (PSNR e SSIM) são calculadas sobre a imagem aproximada obtida utilizando compressão através dos quatro algoritmos propostos para várias taxas de bits por pixel. Estes resultados são comparados com a DCT usual e com outras transformadas direcionais, utilizando, além das curvas RD, as métricas de Bjontegaard (BJONTEGAARD, 2001).

## 4 MELHORIAS NA CODIFICAÇÃO DE IMAGEM BASEADA NA SDCT

Neste capítulo, são apresentadas as melhorias empregadas na codificação de imagem baseada na transformada discreta manobrável do cosseno de forma a torná-la competitiva em termos de RD quando comparada com a codificação realizada pelo JPEG (PENNEBAKER, 1993). Na próxima seção são detalhados os ajustes que foram necessários sobre o que foi proposto em Fracastoro e Magli (2016) e Fracastoro e Magli (2017) no sentido de transformar os algoritmos propostos num CODEC de compressão de imagens completamente funcional, permitindo assim uma comparação com o padrão JPEG tanto em termos de distorção como em termos de taxa e complexidade computacional. Na Seção 4.2, são descritas as contribuições deste trabalho para melhoria do desempenho RD deste CODEC.

### 4.1 CRIAÇÃO DE UM CODEC COMPLETO BASEADO EM SDCT

Os algoritmos utilizados em Fracastoro e Magli (2017) foram escritos pelos autores originais parcialmente em C, incluindo também partes escritas em Shell Script e Matlab. Cada uma das transformadas propostas (SDCT-1, SDCT-BT e SDCT-AM) é executada através de um programa principal independente escrito em C. Entretanto, a entrada de dados (uma imagem sem compressão a ser processada) precisa ser feita através de arquivos codificados como texto puro (ASCII), tarefa feita por um script em linguagem Matlab. Cada bloco  $N \times N$  da imagem a ser processada é fornecido através de um arquivo texto separado, de forma que o processamento de uma única imagem envolve a utilização de uma grande quantidade de arquivos.

O programa principal então converte a informação de texto para valores em aritmética de ponto flutuante, sobre os quais é aplicada a transformação apropriada, bloco a bloco. Os coeficientes gerados neste processo passam por quantização uniforme: divisão por fator de qualidade  $Q$  seguida por arredondamento para o inteiro mais próximo, gerando perda de informações. Uma versão aproximada dos coeficientes originais é então obtida pela multiplicação dos coeficientes quantizados pelo mesmo fator  $Q$ .

A versão aproximada dos coeficientes é utilizada, mediante aplicação de transformação inversa, para obter uma versão aproximada do bloco de imagem original. O erro médio quadrático entre a versão original e a versão aproximada é calculado, para que possa ser utilizado no cálculo do fator RD, conforme (3.5) na página 31.

Para obtenção da taxa, informação também utilizada na composição do fator RD, o bloco  $N \times N$  de coeficientes quantizados é passado como entrada para uma função que o codifica em  $PB$  planos de bit e um plano de sinal. A quantidade de planos  $PB$  é calculada como sendo o menor inteiro maior que o logaritmo base dois da magnitude máxima entre os coeficientes

quantizados para o bloco, isto é,

$$PB = \lceil \log_2(\max(\hat{\mathbf{f}})) \rceil.$$

Cada um dos planos de bit, e também o plano de sinais, é então vetorizado linha a linha e passado como entrada para um codificador aritmético binário adaptativo.

O codificador aritmético utilizado em Fracastoro e Magli (2017) corresponde ao proposto por Witten, Neal e Cleary (1987). Trata-se de um codificador aritmético binário baseado em aritmética exata de inteiros, incluindo multiplicações e divisões. A modelagem adaptativa utiliza apenas um contexto, em que é atualizada a frequência de ocorrência dos símbolos 0 e 1 em cada plano de bits (ou equivalentemente os sinais positivo e negativo) a medida que novos símbolos são codificados. A cada novo plano de bits entregue ao codificador aritmético, o contexto é reiniciado para uma estimativa inicial equiprovável entre os símbolos, de forma que não há o aproveitamento de correlação estatística entre blocos distintos da imagem, ou mesmo entre planos de bits distintos.

A faixa máxima utilizada para o subintervalo de mensagem é codificada utilizando 16 bits, ocorrendo renormalização sempre que a codificação de um novo símbolo resultar num subintervalo pequeno a ponto de ser possível a ocorrência de *underflow* ou *overflow*, de acordo com a faixa de frequências permitidas na modelagem estatística dos símbolos. No caso, foi utilizada uma frequência de símbolo máxima de 16383, de forma injustificada, já que supera em mais de uma ordem de grandeza a quantidade máxima de bits em determinado plano, para valores típicos de  $N$ . O modelo teria suas probabilidades renormalizadas para a metade caso este valor fosse atingido para algum dos símbolos. Devido ao processo de reinicialização do modelo a cada plano de bit, esta condição não ocorre para valores de  $N$  de até 127.

O algoritmo do codificador aritmético é utilizado apenas para simular uma codificação e fornecer o tamanho, em bits, da palavra-código aritmética relativa a todos os símbolos-fonte de determinado bloco, sem escrita da mesma em arquivo no disco. Como a quantidade de planos de bit a serem codificados varia a cada bloco, de acordo com a maior magnitude entre seus coeficientes quantizados, seria necessário informar ainda ao decodificador esta quantidade, ou introduzir um símbolo-fonte indicativo de final de bloco (utilização de codificador aritmético não binário), o que resultaria num maior valor para o tamanho final da palavra-código aritmética. Foi verificado que esta quantidade adicional de bits não estava sendo considerada na estimativa da taxa para composição do fator RD por Fracastoro e Magli (2017).

As seguintes alterações foram realizadas de forma a obter-se um CODEC de imagem completamente funcional e adequado à incorporação das contribuições que serão apresentadas na próxima seção:

1. Criação e integração de funções em linguagem C para interface direta com arquivos de imagem sem compressão, em formato .TIFF (ASSOCIATION, 1992);

2. Integração de todos os algoritmos SDCT em um único programa principal escrito em C, que passa a incluir também a possibilidade de codificar em JPEG, para fins de comparação;
3. Utilização do codificador aritmético de forma completa, com escrita de arquivo codificado para o disco, incluindo informação de fim de bloco necessária para decodificação posterior;
4. Métricas de distorção são calculadas apenas após decodificação dos arquivos produzidos;
5. Comparação dos resultados com o JPEG (*baseline Huffman*).

O algoritmo modificado passa então a ser capaz de ler do disco um arquivo de imagem sem compressão e, após codificação utilizando um dos algoritmos SDCT, produzir um arquivo codificado com perdas. De forma inversa, é também capaz de ler do disco o arquivo produzido e, após decodificação, fornecer uma versão aproximada da imagem original, que pode ser exibida na tela para comparação subjetiva ou utilizada para cálculo de métricas objetivas de distorção. Por desempenho, o CODEC foi completamente desenvolvido em linguagem C e compilado utilizando GCC para Linux. Entretanto, por conveniência, a versão aproximada da imagem obtida após decodificação é gravada em disco utilizando formato binário *raster* conveniente para pós-processamento por *scripts* escritos em linguagem de programação Python, de mais alto nível, que é utilizada para cálculo das métricas de distorção, exibição da imagem na tela e geração dos gráficos utilizados nesta dissertação.

O CODEC baseado nos algoritmos originais de Fracastoro (2017) foi então utilizado para codificar e decodificar a imagem Lena<sup>1</sup> (WEBER, 1997), com levantamento de dezoito pontos experimentais para cada transformada. Cada um desses pontos experimentais correspondendo à codificação da imagem utilizando um passo de quantização diferente. Os valores das curvas para valores não disponíveis de taxa foram obtidos mediante interpolação linear, de forma a permitir uma melhor comparação entre as diversas transformadas. O resultado é mostrado nas Figuras 7 e 8. Como é possível ver, as SDCTs apresentaram um desempenho próximo ao desempenho da DCT, porém todas as opções apresentaram um desempenho consideravelmente inferior ao padrão JPEG (*baseline Huffman*).

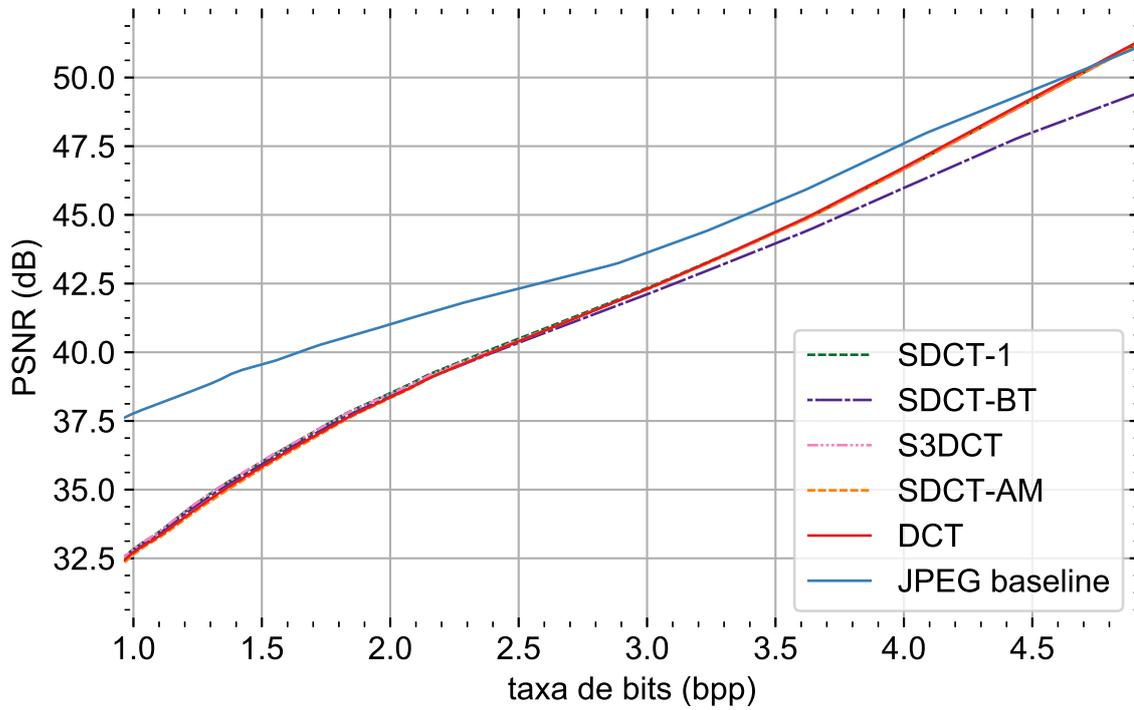
#### 4.1.1 Meta informações para o decodificador

Para permitir a decodificação independente do arquivo final produzido pelo processo de compressão, foi necessário adicionar um pequeno cabeçalho ao arquivo final, contendo as seguintes informações a serem transmitidas ao decodificador: quantidade de camadas de cor na imagem, versão da transformada utilizada para codificação, quantidade de bits utilizada para representação de um ângulo sem compressão, largura e altura da imagem (em pixels) e fator de qualidade utilizado. Estas informações foram inseridas nos primeiros 6 bytes do arquivo final codificado da forma mostrada na Figura 9.

---

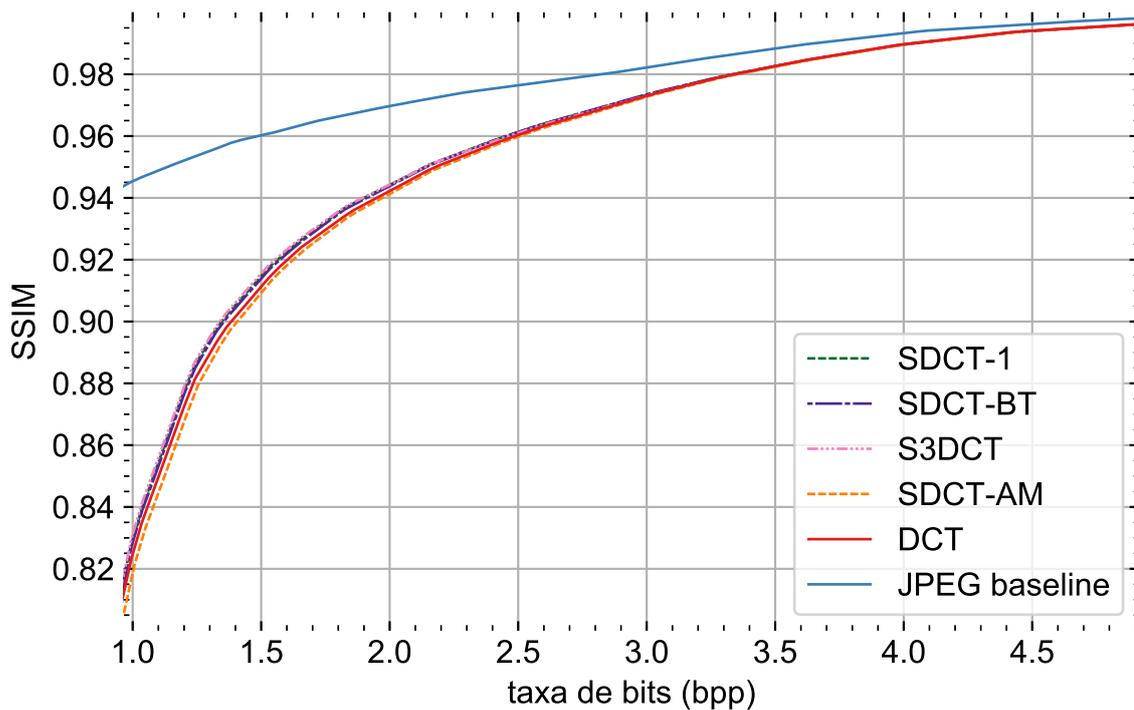
<sup>1</sup> Ver imagens utilizadas no Apêndice A.

Figura 7 – Curvas RD para o CODEC baseado nos algoritmos originais (Peak Signal-to-Noise Ratio - PSNR).



Fonte: O autor (2019).

Figura 8 – Curvas RD para o CODEC baseado nos algoritmos originais (Structural Similarity Index - SSIM).



Fonte: O autor (2019).

Figura 9 – Cabeçalho do arquivo codificado contendo metadados necessários para decodificação.

	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
<b>Byte-0</b>	Bits de ângulo (1-7)			Transformada (0-6)			Cores (1-3)	
<b>Byte-1</b>	Fator de qualidade (0-255)							
<b>Byte-2</b>	Largura da imagem (1-65535)							
<b>Byte-3</b>								
<b>Byte-4</b>	Altura da imagem (1-65535)							
<b>Byte-5</b>								

Fonte: O autor (2019).

O fator de qualidade corresponde a um valor entre 0 (menor taxa, com utilização de um passo de quantização máximo) e 255 (menor distorção, com utilização de um passo de quantização mínimo). Na versão do CODEC utilizada para produzir os resultados desta dissertação, os valores possíveis para o passo uniforme de quantização correspondem à faixa entre 1 e 35, sendo utilizada uma variação linear em relação ao fator de qualidade informado no cabeçalho.

#### 4.1.2 Otimizações para tempo de execução

Ao trabalhar com processamento digital de imagens, um grande número de operações matriciais, especialmente multiplicações matriciais, precisam ser efetuadas, de forma que a otimização do código em termos de tempo de execução para realizar este tipo de operação torna-se de grande relevância para garantir o desempenho geral do *software* desenvolvido. Em processadores da Intel, várias operações relacionadas ao processamento digital de sinais são fornecidas diretamente pelo *hardware* através de conjuntos especiais de instruções, dentre os quais: MMX (Multimedia Extensions), MMX2, SSE (Streaming Single instruction multiple data Extension), SSE2, SSE3 e SSE4 (LANDRÉ; TRUCHETET, 2007). Estas instruções podem ser acessadas através de APIs específicas fornecidas para linguagem C pela própria Intel, dentre as quais a Intel Math Kernel Library (WANG et al., 2014), que foi utilizada para realizar as operações matriciais na versão otimizada do CODEC desenvolvido neste trabalho.

Além das multiplicações matriciais, as transformadas manobráveis do cosseno introduzem ainda um elemento adicional de complexidade: funções trigonométricas do seno e cosseno. Por outro lado, com a quantização realizada sobre os ângulos, na versão final do CODEC desenvolvido neste trabalho, os ângulos de rotação tipicamente podem assumir apenas quatro ou oito valores possíveis, dependendo da configuração utilizada para codificar a imagem, conforme é visto mais adiante. Torna-se então viável a manutenção em memória de tabelas do tipo dicionário contendo o valor previamente calculado para os senos e cossenos de cada um dos oito valores.

Após as otimizações, a aplicação final foi compilada utilizando o GCC (GNU C Compiler) em sua versão 8.3.0, utilizando nível de otimização O3, em sistema operacional Linux Ubuntu 18.10 de 64 bits, para o computador pessoal Lenovo Ideapad 330S-15IKB (processador Intel Core i7-8550U @ 1.80GHz × 8). O tempo de execução do processo completo de codificação e

Tabela 1 – Tempo para completar a seguinte tarefa: codificar e decodificar a imagem F16 utilizando cada um dos algoritmos disponíveis (SDCT-1, SDCT-AM, SDCT-BT, S<sup>3</sup>DCT, DCT e JPEG) dezoito vezes (dezoito níveis de qualidade), com 3 bits para quantização dos ângulos, e utilizando até 8 processos em paralelo.

<b>Tipo de otimização</b>	<b>Mínimo</b>	<b>Máximo</b>	<b>Desvio Padrão</b>	<b>Média</b>
MKL e tabelas trigonométricas	23,94 s	28,10 s	0,57 s	27,59 s
Tabelas trigonométricas	28,21 s	35,04 s	0,88 s	33,83 s
Sem otimizações	28,05 s	36,50 s	1,05 s	34,81 s

Fonte: O autor (2019).

decodificação para todos os algoritmos utilizados (JPEG+Huffman, JPEG+CABAC, S<sup>3</sup>DCT, SDCT-1, SDCT-BT e SDCT-AM) foi medido para processar a imagem F16 em escala de cinza com tamanho  $512 \times 512$  e 8 bits por pixel e utilizando precisão de 3 bits para quantização dos ângulos.

O processo foi então repetido para dezoito valores possíveis de fator de qualidade, com execução paralela dos mesmos, mantendo no máximo oito processos em execução simultânea, para que o tempo de execução total para concluir o processamento de todos os processos fosse medido. Para minimizar a influência de outros processos executados em paralelo pelo sistema operacional, os processos foram enviados ao Kernel Linux com fator de prioridade máxima (fator *niceness* de -20). Essas medições foram repetidas 50 vezes para cada uma de três versões do código: uma utilizando rotinas de multiplicação matricial MKL e tabelas de funções aritméticas, outra utilizando apenas as tabelas de funções trigonométricas e por fim uma última sem utilizar nenhum dos dois recursos. Os valores são mostrados na Tabela 1. Vê-se que uma redução média de cerca de 21% no tempo de execução foi obtida para a versão otimizada com as sub-rotinas Intel MKL e tabelas de funções trigonométricas.

#### 4.1.3 Context Based Arithmetic Coding (CABAC)

O codificador aritmético utilizado no JPEG é chamado QM-coder: um codificador aritmético adaptativo projetado com foco em simplicidade e velocidade. Ele lida apenas com entradas binárias e utiliza uma aproximação no lugar de multiplicações, além de utilizar aritmética de inteiros de precisão fixa. Há a necessidade de renormalizações periódicas do intervalo de probabilidade, mantendo-o próximo da unidade, de forma que a aproximação realizada mantenha-se sempre próxima da multiplicação verdadeira.

O método utilizado para estimativas de probabilidade pelo QM-coder é baseado numa tabela de valores pré-definidos de probabilidade para o símbolo menos provável ( $Q_e$ ).  $Q_e$  é inicializado próximo a 50% e é atualizado sempre que uma renormalização ocorre, de forma que o método para estimativa de probabilidade não precisa ser aplicado de forma muito frequente e envolve apenas consultas a uma tabela e incremento ou decremento de um índice inteiro. A tabela utilizada pelo QM-coder do JPEG foi preparada utilizando conceitos de estimativa de

probabilidade baseada em estatística Bayesiana (SALOMON, 2004).

Para melhor desempenho, o codificador aritmético adaptativo QM-coder necessita ainda de um modelo estatístico de Markov, que permite que estimativas de probabilidade condicional sejam realizadas sobre as decisões de codificação binárias (GONZALES et al., 1990). Este condicionamento precisa ser idêntico tanto no codificador como no decodificador, de forma que se possa utilizar apenas informações conhecidas para ambos em dado momento. Cada estado condicional, ou contexto, no modelo estatístico utiliza um índice independente para estimativa de probabilidade com base na tabela subjacente de valores possíveis para  $Q_e$ .

De forma a permitir codificação de símbolos não binários (como por exemplo os coeficientes quantizados dos blocos de imagem) um estágio adicional de binarização é necessário. Nesta etapa, os símbolos não binários são mapeados em uma sequência binária única, constituída por um bit inicial que indica se o símbolo é nulo ou não, que caso não seja nulo, é seguido por um bit de sinal e mais alguns bits indicando a faixa de magnitude do valor absoluto do símbolo. Por fim, bits indicando o valor específico do símbolo naquela faixa de magnitude são adicionados. Caso o símbolo não seja nulo, ele é seguido ainda por um bit indicativo de final de bloco, de forma que os símbolos nulos seguintes não precisem ser codificados.

Resumidamente, temos as três etapas que precisam ser seguidas para realizar a codificação aritmética binária baseada em contextos (MARPE; SCHWARZ; WIEGAND, 2003):

- Binarização;
- Contextualização;
- Codificação Aritmética Binária.

Neste trabalho, foi utilizado o que é descrito no padrão JPEG em seu anexo D (PENNEBAKER, 1993) para as etapas de binarização e codificação aritmética binária. A etapa de contextualização também foi baseada neste mesmo padrão, porém foi necessário realizar modificações no modelo com a criação de novos contextos para codificação dos ângulos de rotação. Na versão do QM-coder utilizada neste trabalho, são utilizados 49 contextos para codificação dos coeficientes DC, 245 contextos para codificação dos coeficientes AC, 113 contextos para codificação dos valores dos ângulos e mais 113 contextos para codificação da distribuição dos ângulos (no caso da transformada SDCT-AM). A codificação da árvore binária de distribuição dos ângulos para a transformada SDCT-BT utiliza um único contexto com estimativa de probabilidade fixa de 50%.

Cabe ressaltar que os contextos a que nos referimos no parágrafo anterior correspondem a contextualização entre os diferentes tipos de símbolos no processo de binarização. Ou seja, o condicionamento estatístico se dá com base no avanço ao longo da cadeia decisiva da binarização, bem como na natureza da informação sendo codificada, conforme metodologia

Tabela 2 – Métrica de Bjontegaard considerando PSNR (BD-Rate): variações percentuais médias nas taxas de bits graças ao CABAC.

	<b>Lena</b>	<b>House</b>	<b>Aerial</b>	<b>Boat</b>	<b>Bridge</b>	<b>Couple</b>	<b>F16</b>	<b>Média</b>
<b>DCT</b>	-44,14	-47,22	-34,21	-40,64	-32,33	-41,89	-47,84	-41,18
<b>SDCT-1</b>	-46,01	-49,00	-36,21	-41,65	-34,37	-43,73	-49,66	-42,95
<b>SDCT-BT</b>	-48,26	-51,48	-37,63	-43,12	-35,75	-45,55	-52,08	-44,84
<b>S<sup>3</sup>DCT</b>	-43,76	-47,28	-34,38	-40,68	-32,48	-41,92	-47,55	-41,15
<b>SDCT-AM</b>	-44,81	-47,88	-34,53	-41,15	-32,73	-42,28	-48,38	-41,68

Fonte: O autor (2019).

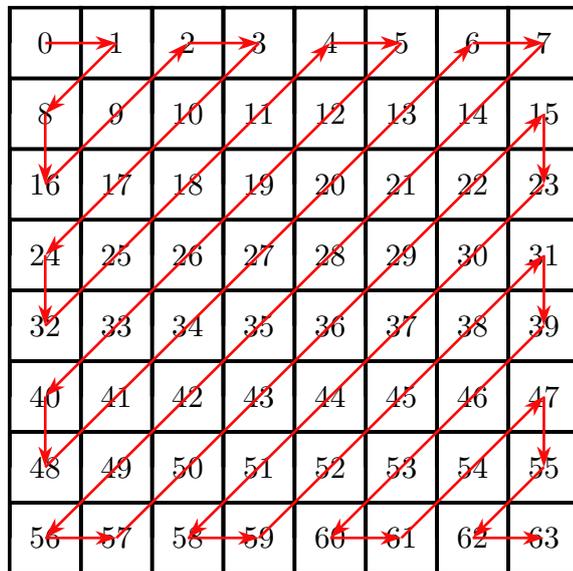
descrita detalhadamente no anexo D do padrão JPEG. Não há uma contextualização em relação aos símbolos originais não binários codificados anteriormente, ao contrário do que é utilizado em construções mais recentes voltadas à codificação de vídeo, como por exemplo em Sze e Budagavi (2012). No caso de codificações de imagens estáticas, as corridas de codificações são mais curtas, de forma que os ganhos obtidos com a utilização da modelagem por cadeias de Markov de ordens mais altas ficam limitados, justificando a escolha da modelagem mais simples, proposta pelo JPEG.

Um outro ajuste que precisou ser realizado em relação ao que é proposto no padrão JPEG consiste na ordenação dos coeficientes em zigue-zague. Esta ordenação foi mantida no caso dos coeficientes das transformadas SDCT-1, SDCT-BT e SDCT-AM, porém precisou ser adaptada para a ordenação dos coeficientes da transformada S<sup>3</sup>DCT, que passou a seguir uma ordenação zigue-zague na matriz triangular inferior incluindo a diagonal principal. Além disso, precisou-se pensar também numa forma de ordenar os ângulos das transformadas S<sup>3</sup>DCT, SDCT-AM e SDCT-BT, que passam a seguir uma ordenação zigue-zague ao longo da matriz triangular inferior excluindo a diagonal principal, em que o elemento percorrido nesta sub-matriz indica um dos coeficientes que compõem o par rotacionado por aquele ângulo. As ordenações utilizadas podem ser vistas na Figura 10.

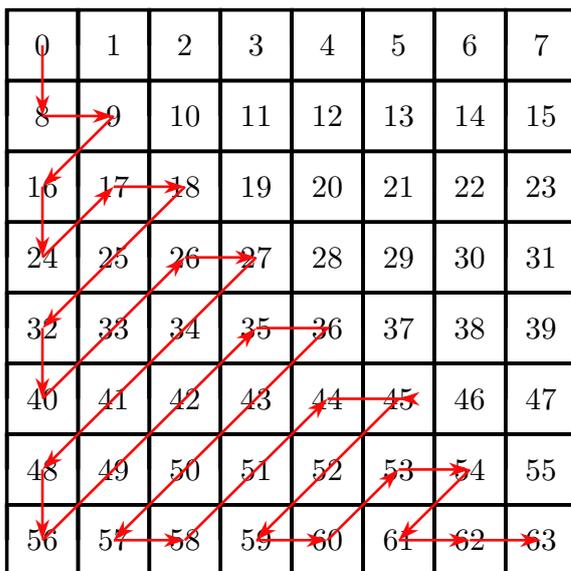
Com a substituição do codificador aritmético utilizado em Fracastoro e Magli (2017) pelo codificador CABAC descrito nesta seção, foi obtido um ganho de codificação que se traduziu nas reduções percentuais médias de taxa mostradas na Tabela 2 calculadas para sete imagens diferentes<sup>2</sup> utilizando a métrica de Bjontegaard (BJONTEGAARD, 2001), com ganhos médios variando entre 41,15% e 44,84%. A melhoria no desempenho também é mostrada nos gráficos RD das Figuras 11 e 12, em que se percebe agora uma clara vantagem em relação às curvas do padrão JPEG baseado em Huffman. Além disso, foram observados ainda os ganhos em tempo de execução mostrados na Tabela 3, medidos usando a metodologia descrita na subseção 4.1.2, mostrando uma redução média adicional de 21%.

<sup>2</sup> Ver reprodução das imagens utilizadas na Figura A.1 no Apêndice A.

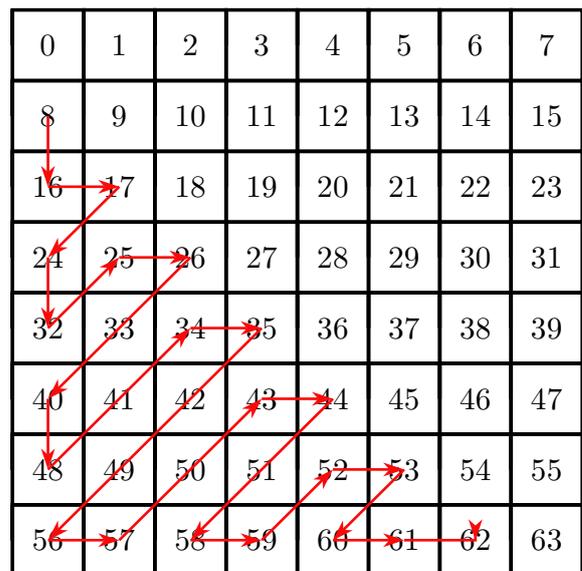
Figura 10 – Os três tipos de ordenação zigue-zague utilizadas para: (a) coeficientes quantizados, (b) coeficientes quantizados da  $S^3DCT$  e (c) valores dos ângulos.



a



b



c

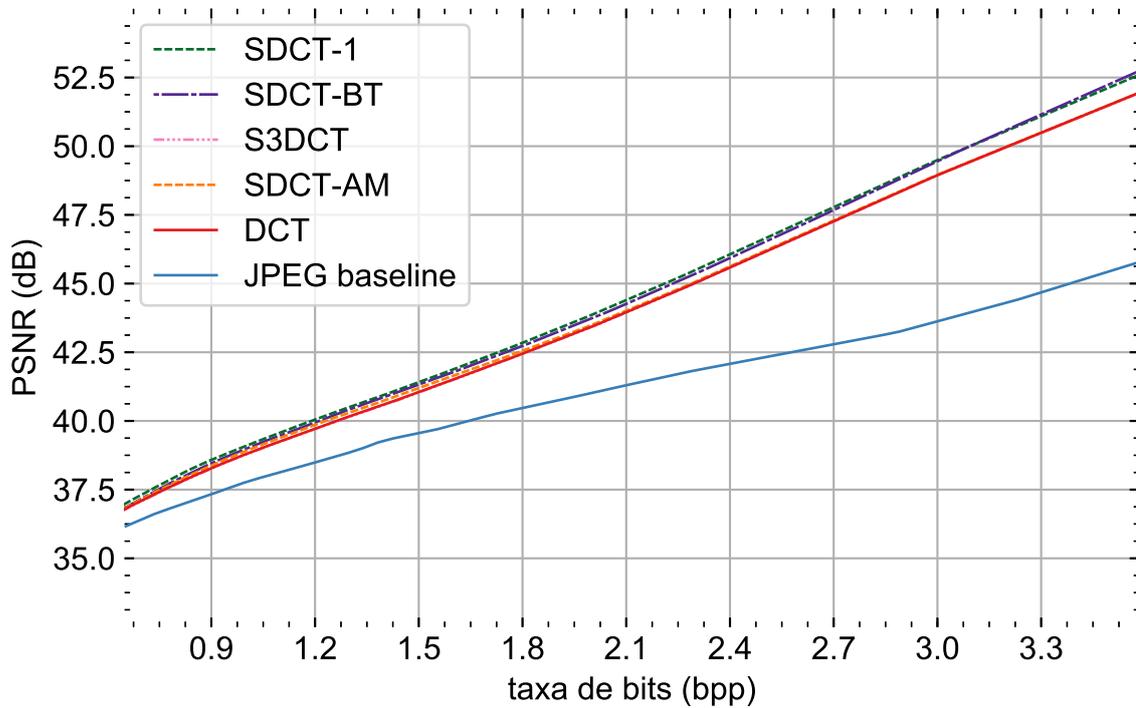
Fonte: O autor (2019).

Tabela 3 – Tempo para completar a seguinte tarefa: codificar e decodificar a imagem F16 utilizando cada um dos algoritmos disponíveis (SDCT-1, SDCT-AM, SDCT-BT,  $S^3DCT$ , DCT e JPEG) dezoito vezes (dezoito níveis de qualidade), com 3 bits para quantização dos ângulos, e utilizando até 8 processos em paralelo.

	Mínima	Máxima	Desvio Padrão	Média
Codificador aritmético original	23,94 s	28,10 s	0,57 s	27,59 s
Codificador aritmético proposto	18,00 s	22,76 s	0,65 s	21,78 s

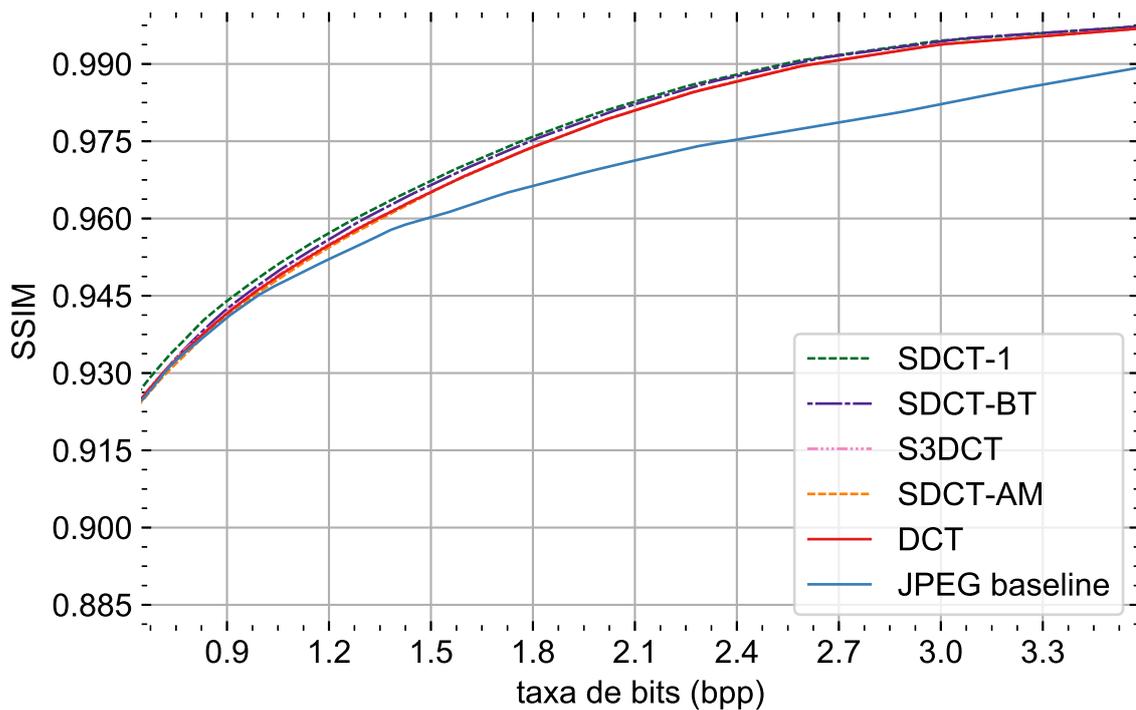
Fonte: O autor (2019).

Figura 11 – Curvas RD para o CODEC após melhorias no codificador aritmético (Peak Signal-to-Noise Ratio - PSNR).



Fonte: O autor (2019).

Figura 12 – Curvas RD para o CODEC melhorias no codificador aritmético (Structural Similarity Index - SSIM).



Fonte: O autor (2019).

## 4.2 MELHORIAS INCREMENTAIS EM DESEMPENHO RD

Nesta seção são apresentadas as principais modificações realizadas no CODEC SDCT buscando melhorias incrementais no desempenho RD.

### 4.2.1 Tabela de Quantização Baseada no Sistema Visual Humano (HVS)

Enquanto o CODEC proposto em Fracastoro e Magli (2017) utiliza um único nível de quantização uniforme para todos os coeficientes gerados após aplicação da transformada, o padrão JPEG utiliza na etapa de quantização dos coeficientes uma tabela de quantização composta por sessenta e quatro valores inteiros que podem variar de 0 a 255 (8 bits) e podem ser escolhidos independentemente pelo codificador. Apesar de não mandatória, o padrão sugere ainda uma tabela de quantização padrão para obter um nível de qualidade de compressão definido como sendo de 50 %. Esta tabela tem suas entradas escolhidas empiricamente de acordo com o peso perceptual que cada coeficiente DCT produz sobre a formação de artefatos visualmente perceptíveis na imagem final. Existem ainda propostas de otimização de tabelas de quantização baseadas no sistema visual humano, como o proposto para imagens estáticas em Watson (1994) e Wang, Lee e Chang (2001), e, mais recentemente, propostas para aplicação no padrão de vídeo HEVC por Shang et al. (2019).

Neste trabalho, propomos a utilização da estratégia de quantização matricial baseada no sistema visual humano com a aplicação da matriz de ponderação por coeficientes proposta por Wang, Lee e Chang (2001) e dada por

$$\mathbf{H} = \begin{bmatrix} 1,0000 & 1,0000 & 1,0000 & 1,0000 & 0,9599 & 0,8746 & 0,7684 & 0,6571 \\ 1,0000 & 1,0000 & 1,0000 & 1,0000 & 0,9283 & 0,8404 & 0,7371 & 0,6306 \\ 1,0000 & 1,0000 & 0,9571 & 0,8898 & 0,8192 & 0,7371 & 0,6471 & 0,5558 \\ 1,0000 & 1,0000 & 0,8898 & 0,7617 & 0,6669 & 0,5912 & 0,5196 & 0,4495 \\ 0,9599 & 0,9283 & 0,8192 & 0,6669 & 0,5419 & 0,4564 & 0,3930 & 0,3393 \\ 0,8746 & 0,8404 & 0,7371 & 0,5912 & 0,4564 & 0,3598 & 0,2948 & 0,2480 \\ 0,7684 & 0,7371 & 0,6471 & 0,5196 & 0,3930 & 0,2948 & 0,2278 & 0,1828 \\ 0,6571 & 0,6306 & 0,5558 & 0,4495 & 0,3393 & 0,2480 & 0,1828 & 0,1391 \end{bmatrix}. \quad (4.1)$$

Cada um dos elementos desta matriz indica a importância perceptual relativa do coeficiente DCT respectivo quando são levadas em conta características do sistema visual humano. Esta matriz deve ser então multiplicada ponto a ponto pelos coeficientes DCT, fazendo com que os coeficientes resultantes desta operação de ponderação possam ser considerados como tendo a mesma importância perceptual sobre a imagem final para observadores humanos. Desta forma, uma quantização uniforme utilizando o mesmo nível pode ser utilizada para todos os coeficientes.

Sejam  $\hat{\mathbf{f}}^{2D-DCT}$  os coeficientes obtidos após aplicação da DCT sobre um bloco de imagem  $N \times N$ ;  $Q^{HVS}(\hat{\mathbf{f}}^{2D-DCT})$  estes mesmos coeficientes após quantização; e  $Q$  o passo

único de quantização uniforme utilizado. Tem-se

$$Q^{HVS}(\hat{\mathbf{f}}^{2D-DCT}) = \text{arred} \left[ \frac{\hat{\mathbf{f}}^{2D-DCT} \circ \mathbf{H}}{Q} \right], \quad (4.2)$$

em que  $\circ$  denota a multiplicação matricial de Hadamard (multiplicação ponto a ponto).

Entretanto, a matriz  $\mathbf{H}$  foi proposta por Wang, Lee e Chang (2001) para aplicação sobre os coeficientes da DCT, se tornando não ótima após aplicação das rotações envolvidas nos vários algoritmos das SDCTs, que resultam em transferência de energia dos coeficientes localizados na submatriz triangular superior de  $\hat{\mathbf{f}}^{2D-DCT}$  para o coeficiente correspondente do par na submatriz triangular inferior, ou vice-versa. Para contornar este problema, este trabalho propõe uma modificação nos algoritmos de Fracastoro e Magli (2017), com a ponderação introduzida pela matriz  $\mathbf{H}$  sobre os coeficientes da transformada DCT antes da multiplicação pela matriz de rotação  $\mathbf{R}(\Theta)$ :

$$Q^{HVS}(\hat{\mathbf{f}}) = \text{arred} \left[ \frac{\mathbf{R}(\Theta) (\hat{\mathbf{f}}^{2D-DCT} \circ \mathbf{H})}{Q} \right]. \quad (4.3)$$

Esta modificação foi realizada para todos os algoritmos SDCT do CODEC e, para que a comparação com a JPEG permanecesse justa, a mesma matriz de quantização resultante foi utilizada na compressão JPEG. Os resultados obtidos após esta modificação são mostrados na Tabela 4, utilizando mais uma vez a redução média percentual na taxa de bits para o mesmo PSNR (métrica de Bjontegaard). Obtivemos uma leve redução de taxa para as transformadas SDCT-1 e SDCT-BT e um leve acréscimo para as transformadas  $S^3$ DCT e SDCT-AM.

Tabela 4 – Métrica de Bjontegaard considerando PSNR (BD-Rate): variações percentuais médias na taxa de bits devido à utilização de quantização com ponderação baseada no Sistema Visual Humano (HVS).

	Lena	House	Aerial	Boat	Bridge	Couple	F16	Média
<b>DCT</b>	0,01	-0,08	-0,02	-0,01	-0,01	-0,02	0,01	-0,02
<b>SDCT-1</b>	-0,39	-0,40	-0,44	-0,14	-0,43	-0,18	-0,37	-0,34
<b>SDCT-BT</b>	-0,12	-0,42	0,06	0,04	0,10	-0,23	-0,19	-0,11
<b><math>S^3</math>DCT</b>	0,84	0,28	0,15	0,78	0,47	0,42	0,40	0,48
<b>SDCT-AM</b>	0,00	0,03	0,01	0,04	0,00	-0,01	0,05	0,02

Fonte: O autor (2019).

#### 4.2.2 Deslocamento de níveis de intensidade e codificação diferencial (DPCM)

Sejam blocos de imagem  $N \times N$  com níveis de intensidade podendo assumir valores entre 0 e  $2^k - 1$ . Neste caso, o coeficiente DC após aplicação da DCT assumirá valores também entre 0 e 255, correspondendo à média de intensidade entre os pixels do bloco de imagem. Para

reduzir a variação dinâmica da magnitude do coeficiente DC, é realizado um deslocamento de nível (subtração) de  $2^{k-2}$ , de forma que os valores são mapeados na faixa de  $-128$  a  $127$ .

Além disso, buscando explorar possíveis correlações entre o nível médio de intensidade em blocos vizinhos de imagem naturais, os coeficientes DC são codificados de forma diferencial. O primeiro bloco a ser processado para determinada camada de cor da imagem tem seu valor DC codificado diretamente, enquanto os blocos seguintes têm codificado apenas o valor da variação deste coeficiente em relação ao bloco anterior.

Essas duas estratégias são utilizadas pelo padrão JPEG baseline e foram replicadas neste trabalho para o CODEC utilizando as SDCTs, resultando nas reduções médias adicionais de taxa de bits mostradas na Tabela 5.

Tabela 5 – Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits com a utilização de DPCM e deslocamento de faixa.

	Lena	House	Aerial	Boat	Bridge	Couple	F16	Média
<b>DCT</b>	-1,59	-3,07	-0,13	-1,44	-0,67	-1,56	-3,10	-1,66
<b>SDCT-1</b>	-1,57	-3,10	-0,19	-1,40	-0,68	-1,53	-3,06	-1,66
<b>SDCT-BT</b>	-1,53	-2,99	-0,17	-1,37	-0,66	-1,52	-2,99	-1,60
<b>S<sup>3</sup>DCT</b>	-1,56	-3,06	-0,18	-1,47	-0,68	-1,55	-3,07	-1,65
<b>SDCT-AM</b>	-1,54	-3,05	-0,12	-1,32	-0,66	-1,41	-2,99	-1,59

Fonte: O autor (2019).

#### 4.2.3 Nova faixa para escolha dos ângulos de rotação

Conforme visto no Capítulo 2, cada ângulo  $\theta_k \in \Theta$  é aplicado para rotacionar um par de autovetores  $\mathbf{v}_{u,v}$  e  $\mathbf{v}_{v,u}$ , para todo  $u \neq v$ , em que  $\theta_k$  é um ângulo em  $[0, 2\pi]$ . Pode-se verificar que todos os pares de vetores de base da 2D-DCT exibem a propriedade de simetria definida pela equação 4.4:

$$\begin{aligned} \mathbf{v}_{u,v}^{\theta+\pi/2} &= -\mathbf{v}_{v,u}^{\theta} \\ \mathbf{v}_{v,u}^{\theta+\pi/2} &= \mathbf{v}_{u,v}^{\theta} \end{aligned} \quad (\forall u, v = 0, 1, \dots, N-1 \mid u > v). \quad (4.4)$$

Para o caso  $N = 8$ , esta propriedade pode ser intuída através da visualização da Figura 13 e foi numericamente verificada para todos os valores de  $\theta$  utilizados neste trabalho. Na Figura 13, os  $N^2$  vetores de base da 2D-SDCT são representados de forma matricial, em que o valor de cada elemento escalar define o nível exibido em cada pixel das imagens  $8 \times 8$  em escala de cinza.

A importância disso para o problema RD envolvido na escolha do ângulo de rotação é a de que não haverá alteração na esparsidade da matriz de coeficientes resultante de uma rotação adicional de  $\pi/2$ , mas apenas uma possível troca de sinal de coeficientes ou transferência de energia da submatriz triangular superior para a submatriz triangular inferior, ou vice-versa. Esta propriedade motivou então uma alteração na faixa utilizada para quantizar os ângulos no CODEC

Tabela 6 – Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits com a nova faixa para escolha dos ângulos quantizados.

	<b>Lena</b>	<b>House</b>	<b>Aerial</b>	<b>Boat</b>	<b>Bridge</b>	<b>Couple</b>	<b>F16</b>	<b>Média</b>
<b>SDCT-1</b>	-9,64	-5,24	-0,16	-5,91	-0,20	-4,55	-9,65	-5,05
<b>SDCT-BT</b>	0,33	0,23	0,11	0,35	0,21	0,40	0,03	0,24

Fonte: O autor (2019).

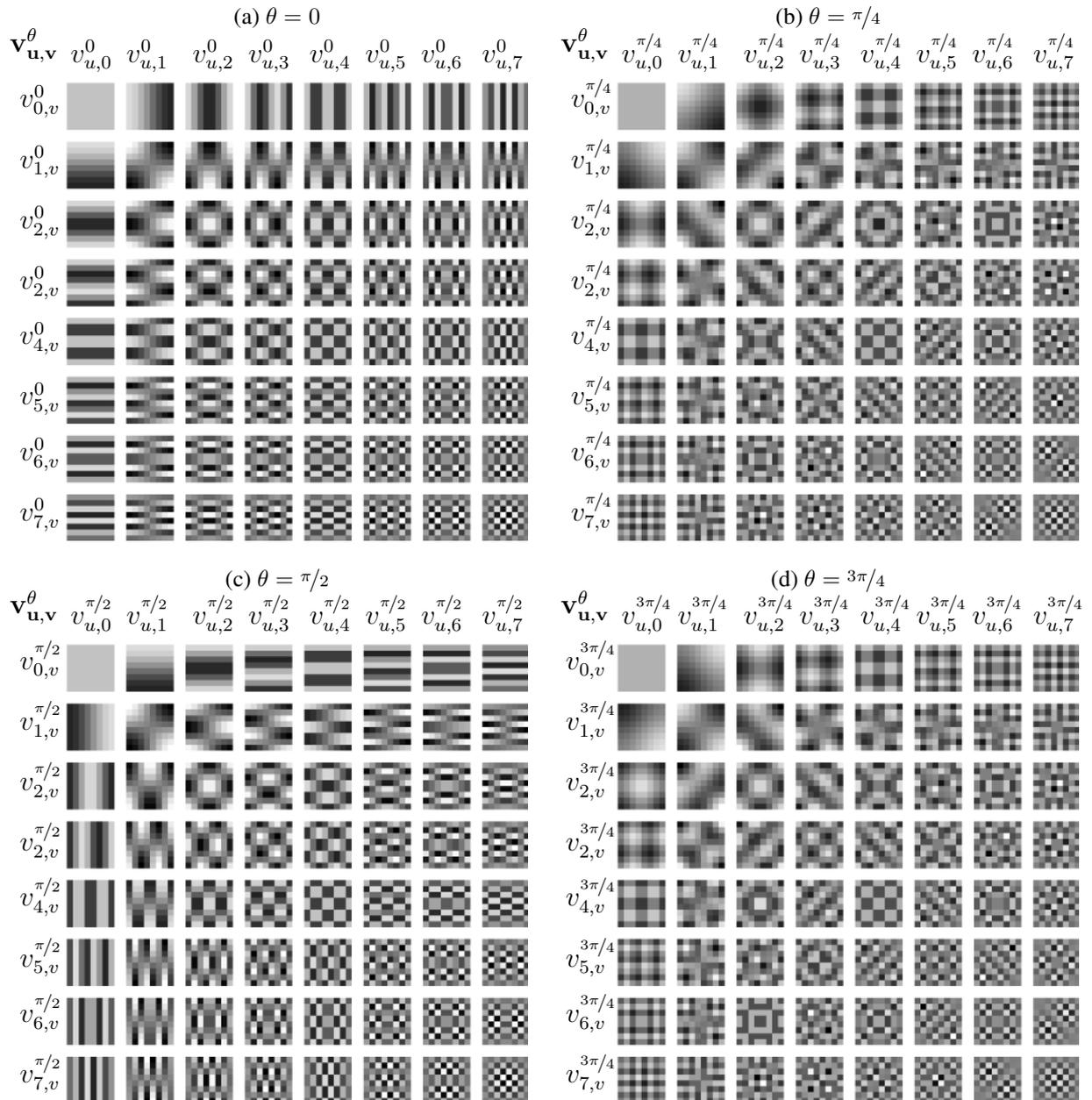
proposto neste trabalho, que passam a ser uniformemente quantizados no intervalo  $[0, \pi/2)$ . Como o ângulo nulo passa a ser incluído entre os possíveis valores assumidos por  $\theta$ , foi possível ainda eliminar o bit adicional de controle utilizado a cada bloco para indicar se é aplicada a SDCT ou a DCT, que passa a ser tratada como um caso particular da SDCT. Esta alteração foi efetuada para o caso das transformadas SDCT-1 e SDCT-BT, enquanto o bit de controle permanece para as SDCT-AM e S<sup>3</sup>DCT. Esta modificação resultou nas alterações médias de taxa de bits mostradas na Tabela 6.

As S<sup>3</sup>DCT e a SDCT-AM não foram incluídas nesta alteração, já que seus algoritmos impõem a necessidade de escolha de um dos coeficientes para concentrar a energia do par. Esta necessidade é explícita no caso da S<sup>3</sup>DCT, cujo objetivo é tornar nula toda a matriz triangular superior. No caso da SDCT-AM, a necessidade dá-se pela escolha dos ângulos ótimos através da equação 3.15, que possui solução em  $[-\pi/2, +\pi/2)$ . Foi mantida então a faixa de  $[-\pi/2, +\pi/2)$  para estas duas transformadas.

Cabe ressaltar neste ponto que a utilização de ângulos uniformemente quantizados na faixa de 0 a 90° foi originalmente proposta por Fracastoro e Magli (2015) para a SDCT-1, porém não no contexto de um CODEC de compressão completo, mas sim no experimento mais simples revisado na Seção 3.3 (página 34) desta dissertação. No capítulo de resultados experimentais de sua tese (FRACASTORO, 2017, p. 70), em que os resultados da SDCT-1 são comparados às demais transformadas em termos RD, são utilizados  $q_\theta = 8$  níveis de quantização uniformemente distribuídos entre 0 e 180°.

#### 4.2.4 Redução na quantidade de níveis de quantização

O estreitamento da faixa de variação do ângulo quantizado de  $[0, \pi]$  para  $[0, \pi/2)$  resultou num ganho em termos de resolução de escolha do ângulo dentro desta faixa. Sob a motivação de redução do tempo de execução para os algoritmos da SDCT, procedeu-se com a redução da quantidade de ângulos de 8 para 4 valores dentro da faixa. Após esta modificação, foram observadas as alterações de taxa média indicadas na Tabela 7 e nos tempos médios de processamento mostrados na Tabela 8. A redução foi realizada apenas para as transformadas que utilizam mais de um ângulo por bloco, que são justamente as que implicam numa maior complexidade computacional e também numa maior taxa devido aos ângulos.

Figura 13 – Vetores de base da 2D-DCT rotacionados pela SDCT-1 para diferentes valores de  $\theta$ .

Fonte: O autor (2019).

Tabela 7 – Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits com a utilização de apenas 4 níveis de quantização para os ângulos.

	Lena	House	Aerial	Boat	Bridge	Couple	F16	Média
<b>SDCT-BT</b>	0,10	0,40	-0,06	0,00	-0,10	0,22	0,18	0,11
<b>SDCT-AM</b>	-0,03	-0,09	-0,01	-0,06	0,00	-0,01	-0,05	-0,04
<b>S<sup>3</sup>DCT</b>	-0,85	-0,30	-0,18	-0,78	-0,46	-0,41	-0,40	-0,48

Fonte: O autor (2019).

Tabela 8 – Variações percentuais médias nos tempos de codificação e decodificação devido à redução de 8 para 4 níveis de quantização.

	<b>Lena</b>	<b>House</b>	<b>Aerial</b>	<b>Boat</b>	<b>Bridge</b>	<b>Couple</b>	<b>F16</b>	<b>Média</b>
<b>SDCT-BT</b>	-73	-61	-70	-72	-72	-72	-72	-70
<b>SDCT-AM</b>	-46	-56	-53	-55	-45	-45	-49	-50
<b>S<sup>3</sup>DCT</b>	-5	-1	-4	-1	0	0	-1	-2

Fonte: O autor (2019).

## 5 RESULTADOS E DISCUSSÃO

Neste capítulo, são apresentados os resultados obtidos após introdução de combinações das alterações descritas no capítulo anterior no CODEC de imagem baseado nas propostas originais das transformadas SDCT-1 (FRACASTORO; MAGLI, 2015),  $S^3$ DCT (FRACASTORO; MAGLI, 2016), SDCT-BT e SDCT-AM (FRACASTORO; FOSSON; MAGLI, 2017). Estes resultados foram comparados com aqueles obtidos utilizando um CODEC compatível com JPEG (*baseline Huffman*) e aqueles obtidos mediante utilização da 2D-DCT usual aliada ao uso da mesma estratégia de quantização e codificação de entropia utilizada para as SDCTs.

Para os resultados utilizando a SDCT-1, foram mantidos  $q_\theta = 8$  ângulos uniformemente quantizados entre  $0^\circ$  e  $90^\circ$ . Já para a transformada SDCT-BT, foram utilizados apenas  $q_\theta = 4$  ângulos uniformemente quantizados entre  $0^\circ$  e  $90^\circ$ . Por fim, para as  $S^3$ DCT e SDCT-AM, foram utilizados  $q_\theta = 4$  ângulos uniformemente quantizados entre  $-90^\circ$  e  $90^\circ$ . Apenas nos casos das  $S^3$ DCT e SDCT-AM é codificado um bit adicional por bloco para indicar se a transformada 2D-DCT usual foi utilizada ao invés da SDCT. Para as SDCT-1 e SDCT-BT, a aplicação da 2D-DCT é tratada como caso particular da SDCT para  $V(\Theta = \vec{0})$ .

Em todos os casos, um arquivo de imagem matricial e sem compressão (8 bits por pixel e em formato .TIFF) é lida do disco e uma versão da mesma imagem, após codificação com perdas, é gravada em disco pelo CODEC. Posteriormente, o arquivo codificado é acessado para o processo de decodificação, que tem como resultado uma nova versão matricial aproximada da imagem original (também 8 bits por pixel). Esta versão aproximada é acessada por *scripts* em linguagem de programação *Python* para cálculo das métricas objetivas de distorção em relação à imagem original. Os *scripts* acessam a taxa média com compressão, em bits por pixel (bpp), através da quantidade de bytes final presente nos arquivos da imagem na sua versão ainda codificada.

Os resultados foram avaliados em detalhes para sete imagens do USC-SIPI (WEBER, 1997): Lena, House, Aerial, Boat, Bridge, Couple e F16. Uma avaliação geral também foi realizada para 886 imagens<sup>1</sup> do banco UCID (SCHAEFER; STICH, 2003). As imagens coloridas foram convertidas de RGB para escala de cinza ( $Y = 0,299R + 0,587G + 0,114B$ ) antes de serem utilizadas. São apresentados gráficos com curvas RD, que apresentam alguma métrica objetiva de distorção em seu eixo vertical (PSNR ou SSIM), enquanto a taxa em bits por pixel é exibida seu eixo horizontal. Adicionalmente, os resultados são comparados empregando a métrica de Bjontegaard BD-Rate (BJONTEGAARD, 2001), que indica o aumento médio percentual da taxa necessária para se obter a mesma distorção em PSNR. Quanto mais negativo, melhor. Apesar de não permitir visualizar que vantagens e desvantagens ocorrem em diferentes graus de acordo com faixas específicas de fator de compressão, as métricas de Bjontegaard permitem uma comparação

<sup>1</sup> Ver colagem com as imagens utilizadas nas figuras de A.2 a A.6 no Apêndice A.

Tabela 9 – Métrica de Bjontegaard (BD-Rate): variações percentuais médias nas taxas de bits em relação à DCT usual.

	<b>Lena</b>	<b>House</b>	<b>Aerial</b>	<b>Boat</b>	<b>Bridge</b>	<b>Couple</b>	<b>F16</b>	<b>Média</b>
<b>SDCT-1</b>	-3,48	-2,98	-2,82	-0,78	-2,84	-1,99	-3,84	-2,68
<b>SDCT-BT</b>	-0,99	-0,15	-1,15	1,72	-1,06	0,34	-1,41	-0,38
<b>SDCT-AM</b>	-0,13	-0,06	-0,03	0,00	-0,03	-0,02	-0,07	-0,06
<b>S<sup>3</sup>DCT</b>	0,24	0,13	0,04	0,25	0,06	0,10	-0,01	0,12

Fonte: O autor (2019).

numérica e direta entre os desempenho das diversas opções.

### 5.1 MÉTRICA DE BJONTEGAARD - PSNR

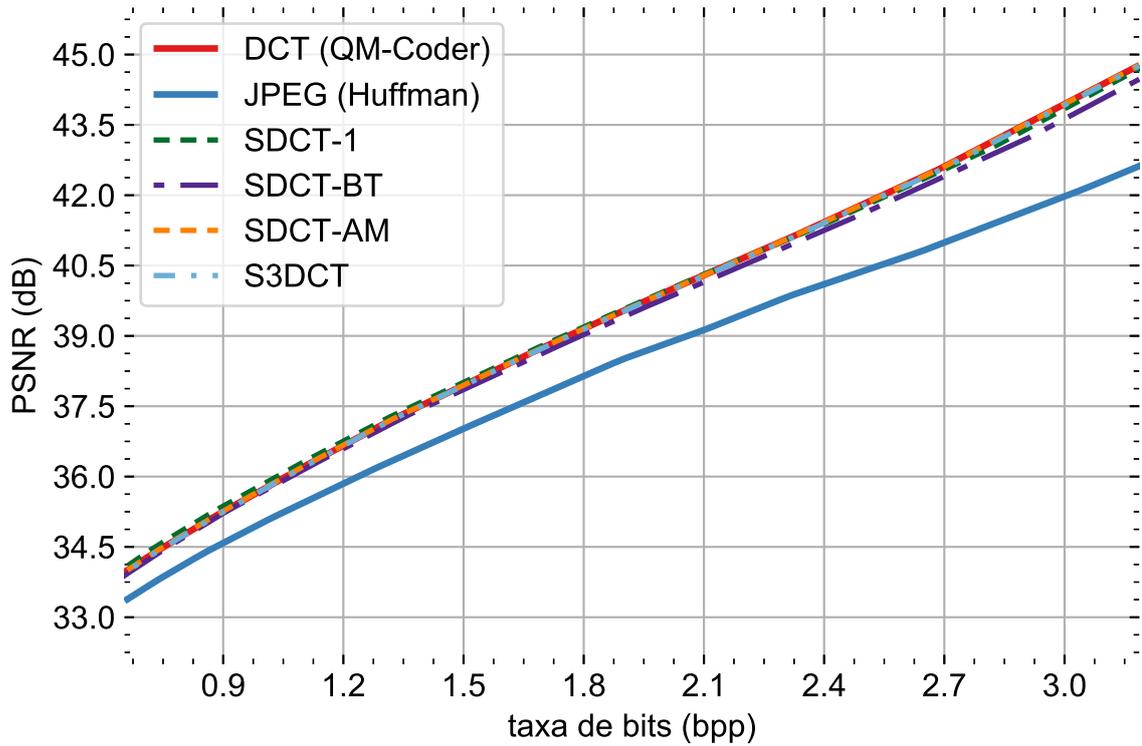
Uma comparação foi realizada entre as versões melhoradas das diversas transformadas SDCT e a utilização da 2D-DCT usual, e os resultados em termos de BD-Rate são mostrados na Tabela 9. Vê-se que, com exceção da S<sup>3</sup>DCT, todas apresentam alguma vantagem percentual, porém tais vantagens se traduzem numa redução percentual de taxa que varia bastante de acordo com a imagem analisada e com a transformada utilizada.

No pior caso, o da S<sup>3</sup>DCT, há uma piora geral na taxa em relação à utilização da 2D-DCT para todas as imagens, com exceção da imagem F16, onde uma pequena vantagem média de 0,01 % é obtida. Na outra ponta, esta transformada entrega uma piora de 0,25 % para a imagem Boat. A S<sup>3</sup>DCT é seguida pela SDCT-AM, que apesar da maior complexidade computacional de seu algoritmo, entrega uma vantagem média de apenas 0,06 % entre todas as sete imagens analisadas.

Os algoritmos de busca exaustiva foram os que se saíram melhor no comparativo, com ganhos médios de 0,38 % para a SDCT-BT e 2,68 % para a SDCT-1. Os ganhos para a SDCT-BT são fortemente dependentes da imagem utilizada e o algoritmo chega a aumentar a taxa de bits em 1,72 % no caso da imagem Boat, enquanto a SDCT-1 entregou um resultado positivo para todas as sete imagens. Além disso, é importante destacar que a SDCT-BT não superou a SDCT-1 para nenhuma das imagens analisadas.

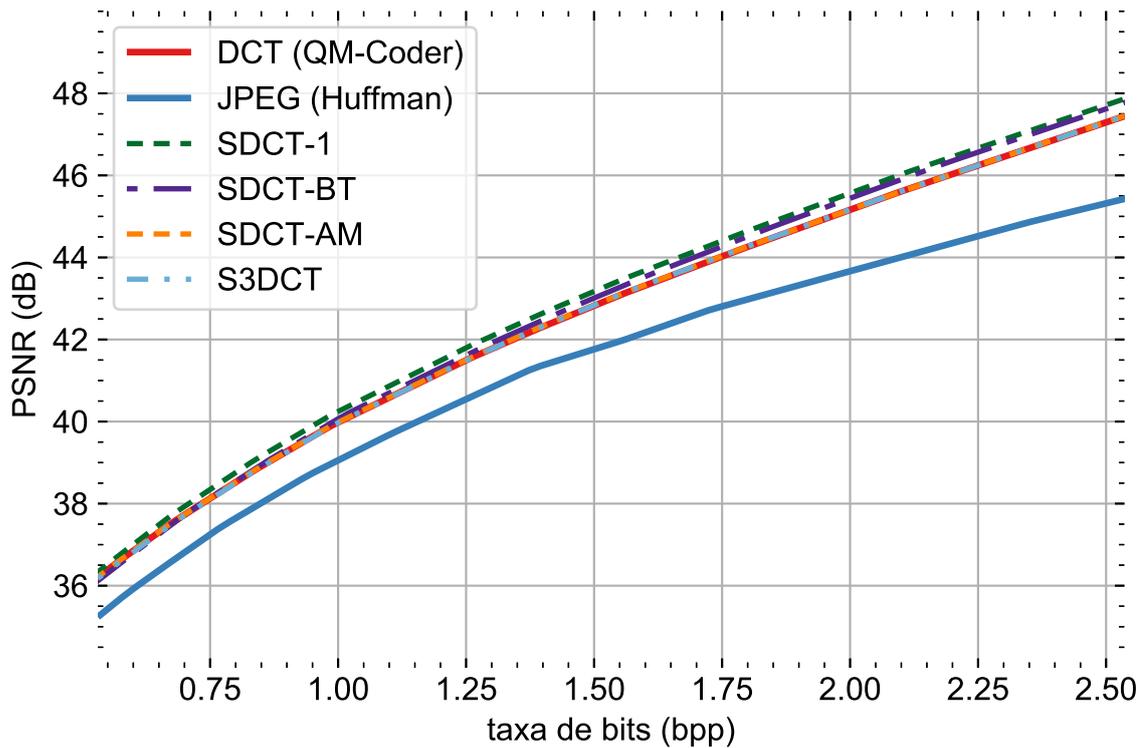
Para permitir uma análise mais detalhada do desempenho do CODEC proposto nos diversos casos, são mostradas as curvas RD para toda a faixa experimental de taxa de compressão para duas imagens. No gráfico da Figura 14, tem-se o resultado para a imagem para a qual o desempenho médio dos algoritmos foi pior (Boat), enquanto no gráfico da Figura 15 é exibido o resultado para a melhor imagem (F16). Vê-se que, além da variação em relação às diferentes imagens, há também um aumento do ganho para taxas de bits mais altas, no caso da imagem F16.

Figura 14 – Curvas RD para a imagem Boat (Peak Signal-to-Noise Ratio - PSNR).



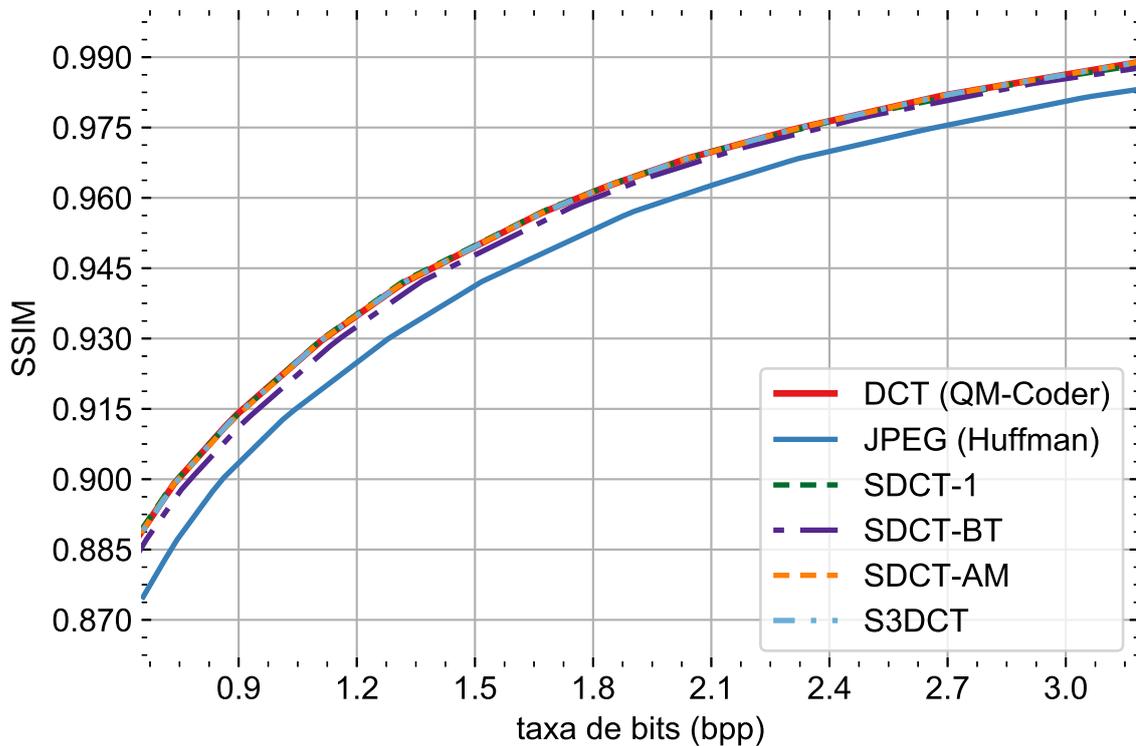
Fonte: O autor (2019).

Figura 15 – Curvas RD para a imagem F16 (Peak Signal-to-Noise Ratio - PSNR).



Fonte: O autor (2019).

Figura 16 – Curvas RD para a imagem Boat (Structural Similarity Index - SSIM).



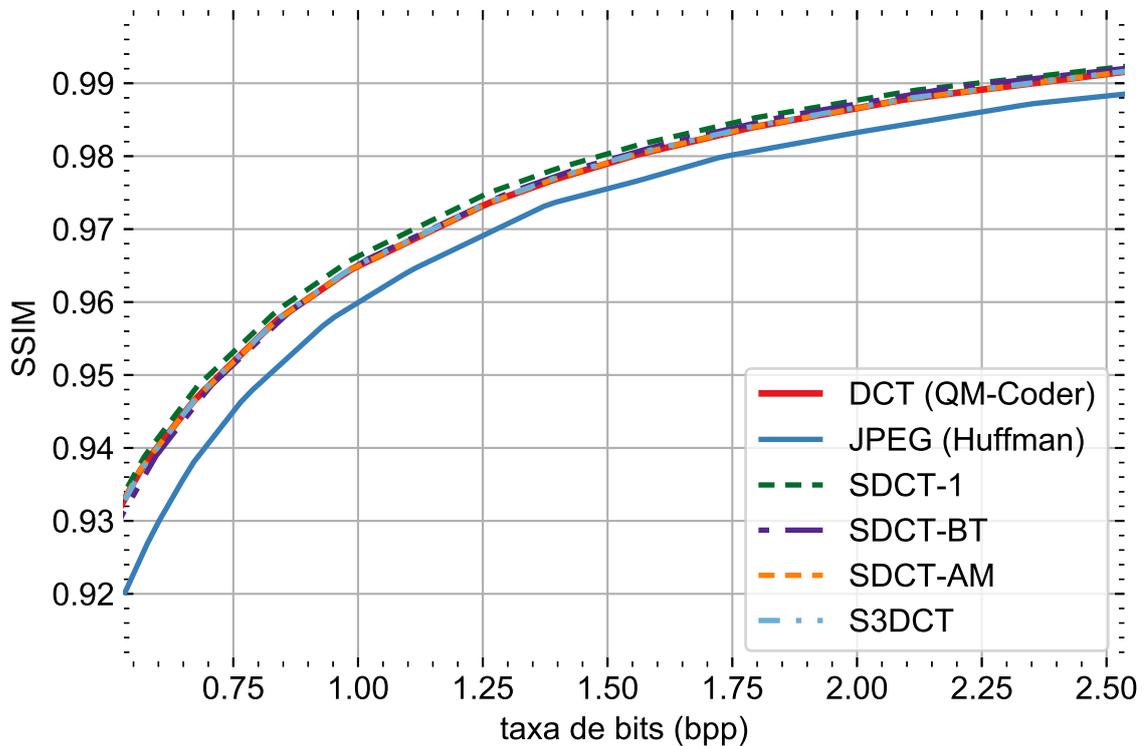
Fonte: O autor (2019).

## 5.2 COMPARATIVO SSIM (STRUCTURAL SIMILARITY INDEX)

Buscando realizar uma comparação de desempenho que se aproxime mais da percepção subjetiva final de qualidade das imagens aproximadas, são mostradas nos gráficos das Figuras 16 e 17 as curvas RD em que a métrica de distorção SSIM (Zhou Wang et al., 2004) é utilizada. O SSIM é uma métrica que leva em consideração características do sistema visual humano (HVS), em especial a hipótese de que o HVS é adaptado a perceber melhor os elementos estruturais de uma imagem, de forma que é esperado que esta métrica reflita melhor a percepção de qualidade subjetiva por um usuário final.

Nas curvas RD para a imagem Boat, vê-se que a SDCT-BT é a transformada de pior desempenho para toda a faixa de taxa analisada. Já no caso da Figura 17, vê-se que a SDCT-BT chega a apresentar perdas para taxas de bits de até 0,9 bpp e apresenta pequenos ganhos apenas para taxas de bits acima de 1,5 bpp. Já a transformada SDCT-1 apresenta desempenho semelhante ao da 2D-DCT para praticamente toda a faixa de taxa analisada para a imagem Boat, enquanto apresenta maior vantagem para faixas intermediárias de taxa de bits no caso da imagem F16.

Figura 17 – Curvas RD para a imagem F16 (Structural Similarity Index - SSIM).



Fonte: O autor (2019).

### 5.3 COMPARAÇÃO VISUAL SUBJETIVA

Por fim, para permitir uma comparação verdadeiramente subjetiva do significado dos resultados obtidos para a imagem F16, suas aproximações utilizando os 4 algoritmos (SDCT-1, SDCT-BT, SDCT-AM e S<sup>3</sup>DCT) com as melhorias propostas, além das aproximações entregues pelo uso da DCT usual e pelo padrão JPEG (*baseline Huffman*), são mostradas nas Figuras de 18 a 21. Nas Figuras 18 e 19, são mostrados os resultados obtidos para uma taxa de aproximadamente 1,05 bpp, enquanto nas Figuras 20 e 21 os resultados mostrados são para uma taxa de apenas 0,5 bpp.

Em cada caso, é mostrado também o melhor resultado obtido utilizando um dos algoritmos propostos por Fracastoro (2017) sem as melhorias introduzidas neste trabalho, mas apenas aquelas necessárias para que os mesmos funcionem como um CODEC completo de imagem (ver detalhes na Seção 4.1). Entretanto, a versão sem melhorias é mostrada apenas para fins de apresentação visual dos efeitos provocados pelas melhorias propostas por este trabalho em relação ao ponto de partida. Não representam uma comparação justa dos algoritmos propostos por Fracastoro *et al.* e a 2D-DCT usual, já que nas imagens mostradas aqui, o CODEC utilizado para a 2D-DCT faz uso da melhoria no codificador aritmético proposta na Subseção 4.1.3, que sozinha representou mais de 40% de ganhos médios em BD-Rate.

Figura 18 – Parte da imagem F16 após compressão a taxa de 1,05 bpp.

(a) Original sem compressão (8 bpp).



(b) JPEG - SSIM de 0,963.



(c) DCT - SSIM de 0,967.



(d) SDCT-BT de Fracastoro, Fosson e Magli (2017) - SSIM de 0,836.



Fonte: Imagem original (sem compressão) obtida de Weber (1997).

Figura 19 – Parte da imagem F16 após compressão a taxa de 1,05 bpp.

(a) SDCT-1 (deste trabalho) - SSIM de 0,968.

(b) SDCT-BT (deste trabalho) - SSIM de 0,968.



(c) SDCT-AM (deste trabalho) - SSIM de 0,967.

(d) S<sup>3</sup>DCT (deste trabalho) - SSIM de 0,967.



Fonte: Imagem original (sem compressão) obtida de Weber (1997).

Figura 20 – Parte da imagem F16 após compressão a taxa de 0,5 bpp.

(a) Original sem compressão (8 bpp).



(b) JPEG - SSIM de 0,926.



(c) DCT - SSIM de 0,929.



(d) SDCT-AM de Fracastoro, Fosson e Magli (2017) - SSIM de 0,550.



Fonte: Imagem original (sem compressão) obtida de Weber (1997).

Figura 21 – Parte da imagem F16 após compressão a taxa de 0,5 bpp.

(a) SDCT-1 (deste trabalho) - SSIM de 0,930.



(b) SDCT-BT (deste trabalho) - SSIM de 0,930.



(c) SDCT-AM (deste trabalho) - SSIM de 0,929.



(d) S<sup>3</sup>DCT (deste trabalho) - SSIM de 0,929.



Fonte: Imagem original (sem compressão) obtida de Weber (1997).

Tabela 10 – Tempo de processamento necessário para codificar a imagem F16.

<b>Transformada</b>	<b>Mínimo</b>	<b>Máximo</b>	<b>Desvio Padrão</b>	<b>Média</b>	<b>Benchmark</b>
<b>DCT</b>	40 ms	57 ms	4 ms	49 ms	100 %
<b>S<sup>3</sup>DCT</b>	60 ms	82 ms	5 ms	68 ms	139 %
<b>SDCT-1</b>	164 ms	177 ms	4 ms	170 ms	346 %
<b>SDCT-BT</b>	367 ms	436 ms	27 ms	413 ms	843 %
<b>SDCT-AM</b>	1104 ms	1333 ms	70 ms	1291 ms	2635 %

Fonte: O autor (2019).

#### 5.4 COMPARATIVO DE TEMPO DE CODIFICAÇÃO E DECODIFICAÇÃO

A comparação entre o desempenho dos CODECs baseados nas SDCTs com o CODEC baseado na DCT não estaria completa caso não fizessemos uma análise do acréscimo de complexidade advindo tanto dos algoritmos de otimização RD como das multiplicações adicionais inseridas pela matriz de rotação  $R(\Theta)$ . A mesma metodologia e equipamento detalhados na Subseção 4.1.2, porém utilizando apenas um processo em paralelo, foi utilizada para obter o tempo de processamento médio demandado por cada uma das alternativas para codificar e decodificar a imagem F16, que, como vimos, foi a imagem que apresentou os maiores ganhos médios com a utilização das SDCTs.

Os tempos necessários para codificar a imagem são mostrados na Tabela 10, enquanto os tempos necessários para decodificá-la são mostrados na Tabela 11. Vê-se que, na implementação proposta neste trabalho, o tempo necessário para efetuar uma codificação utilizando a SDCT-AM (o pior caso) é em média 26,35 vezes maior que o tempo necessário para codificar a mesma imagem utilizando a 2D-DCT usual. Para a SDCT-BT, esse multiplicador é de 8,43 vezes. Sendo assim, esses dois algoritmos impõem um aumento excessivo de complexidade na codificação em relação à 2D-DCT usual. A S<sup>3</sup>DCT excede o tempo de execução obtido para a transformada usual em apenas 39 %, enquanto a SDCT-1 excede em 3,46 vezes.

Por outro lado, na outra ponta do CODEC, o decodificador das SDCTs supera em apenas 19 % o tempo de decodificação da 2D-DCT, no pior caso médio (SDCT-BT) e em 14 % no caso da SDCT-1. Neste ponto, deve-se analisar com cuidado os resultados obtidos para as transformadas S<sup>3</sup>DCT e SDCT-AM, pois as mesmas utilizam um bit adicional inserido pelo codificador para sinalizar que a transformada usual (2D-DCT) deve ser utilizada para determinado bloco. Como o desempenho final destas duas transformadas ficou muito próximo ao da transformada usual, esta última é utilizada para decodificar a grande maioria dos blocos, conforme é visto na próxima seção.

Tabela 11 – Tempo de processamento necessário para decodificar a imagem F16.

<b>Transformada</b>	<b>Mínimo</b>	<b>Máximo</b>	<b>Desvio Padrão</b>	<b>Média</b>	<b>Benchmark</b>
<b>DCT</b>	39 ms	50 ms	3 ms	43 ms	100 %
<b>S<sup>3</sup>DCT</b>	35 ms	51 ms	4 ms	43 ms	100 %
<b>SDCT-1</b>	42 ms	55 ms	3 ms	49 ms	114 %
<b>SDCT-BT</b>	45 ms	60 ms	3 ms	51 ms	119 %
<b>SDCT-AM</b>	38 ms	54 ms	4 ms	45 ms	105 %

Fonte: O autor (2019).

## 5.5 DESEMPENHO MÉDIO SOBRE IMAGENS DO BANCO UCID

Por fim, 886 imagens do banco UCID (SCHAEFER; STICH, 2003) foram submetidas ao mesmo processo de codificação e decodificação aplicado às sete imagens do banco USP-SIPI. Cada uma das 886 imagens foi codificada para 12 valores diferentes de fator de qualidade (diretamente relacionado ao degrau de quantização dos coeficientes e, portanto, à taxa de bits), resultando em um total de aproximadamente  $32 \times 10^6$  blocos de imagem  $8 \times 8$  que foram processados por cada um dos algoritmos SDCT. Para cada um desses blocos foram acumuladas informações sobre: tipo de transformada utilizada (se SDCT ou 2D-DCT usual), quantidade de sub-bandas utilizadas para  $\Theta$  e os valores escolhidos para os ângulos de rotações.

Verificou-se que a SDCT-1 utilizou um ângulo nulo em 40,7 % dos blocos processados, ou seja, a 2D-DCT foi efetivamente utilizada nestes blocos (ver detalhes no gráfico da Figura 22). No caso da SDCT-BT, uma única sub-banda foi utilizada em 45,31 % dos blocos processados, ou seja, algo equivalente à SDCT-1 (com quantização dos ângulos em 4 valores e um bit adicional para representar a árvore) foi efetivamente utilizada em quase metade dos casos. Além disso, o ângulo nulo foi utilizado pela SDCT-BT em 41 % das sub-bandas criadas pelo algoritmo de otimização RD (ver detalhes nos gráficos das Figuras 23 e 26).

No caso da S<sup>3</sup>DCT e da SDCT-AM, observou-se que essas transformadas foram preteridas em relação à 2D-DCT pela otimização RD para a grande maioria dos blocos: a SDCT-AM foi efetivamente utilizada apenas 0,98 % das vezes, enquanto a S<sup>3</sup>DCT o foi para apenas 4,7 % dos blocos processados (ver gráficos detalhados nas Figuras 27 e 28). Além disso, no caso da SDCT-AM, conforme observado no gráfico da Figura 24, a escolha de ângulos ótimos concentrou-se fortemente em um único ângulo de rotação, entre os quatro disponíveis. Nesta análise, considera-se apenas os blocos em que a SDCT-AM foi efetivamente utilizada. Para a S<sup>3</sup>DCT, a escolha dos ângulos ótimos concentrou-se entre as duas opções disponíveis mais próximas de zero, conforme visto na Figura 25.

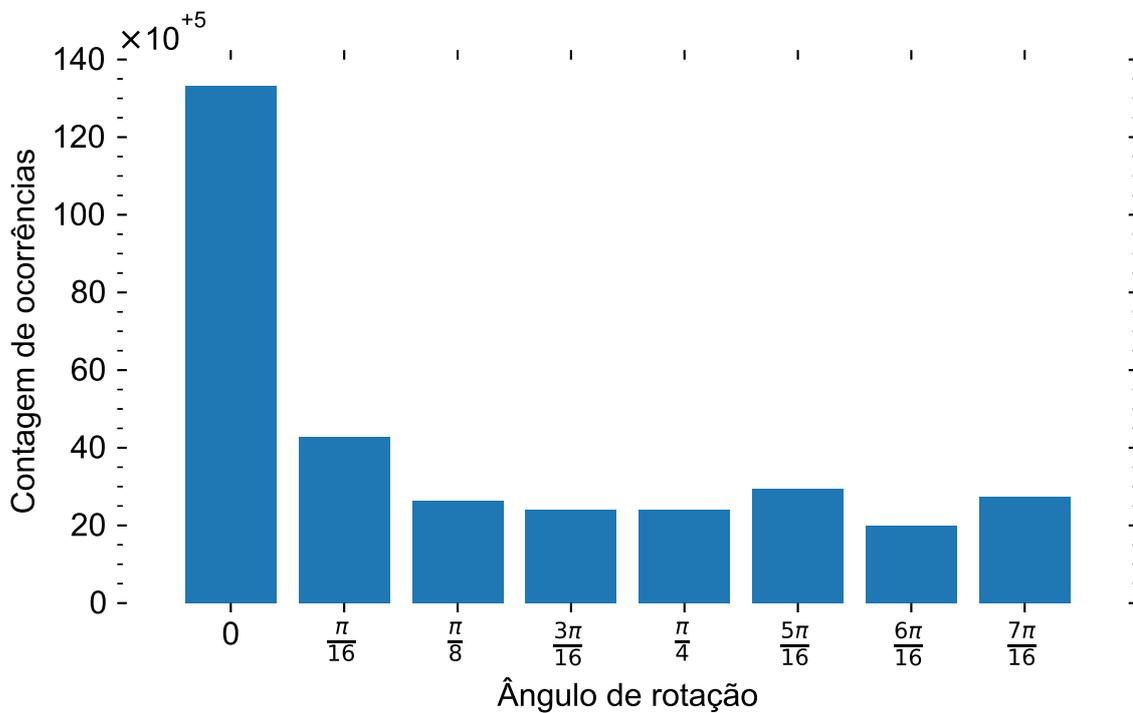
Além disso, métricas de desempenho em comparação com a DCT usual foram calculadas para cada uma das imagens utilizando as métricas de Bjontegaard BD-Rate, BD-PSNR e BD-SSIM (BJONTEGAARD, 2001). Na Tabela 12 são mostrados os valores médios dessas três métricas calculadas sobre as 886 imagens UCID. Vê-se que os resultados médios em termos de

Tabela 12 – Métricas de Bjontegaard (BD-Rate, BD-PSNR, BD-SSIM): resultados médios para todas as 886 imagens do banco UCID.

	<b>BD-Rate</b>	<b>BD-PSNR</b>	<b>BD-SSIM</b>
<b>SDCT-1</b>	-2,73 %	+0,229	+0,00087
<b>SDCT-BT</b>	-0,38 %	+0,041	-0,00027
<b>SDCT-AM</b>	-0,08 %	+0,007	+0,00001
<b>S<sup>3</sup>DCT</b>	+0,09 %	-0,006	+0,00003

Fonte: O autor (2019).

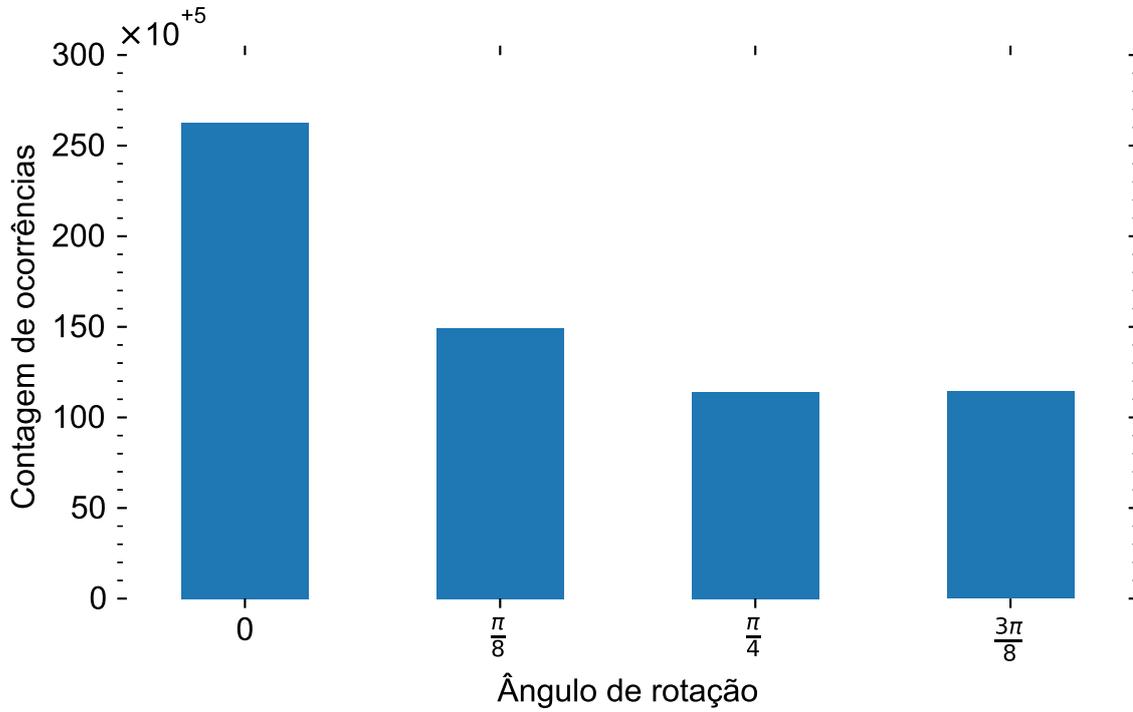
Figura 22 – Histograma de escolha de ângulos para a SDCT-1: o ângulo nulo foi escolhido em 40,7 % dos blocos processados.



Fonte: O autor (2019).

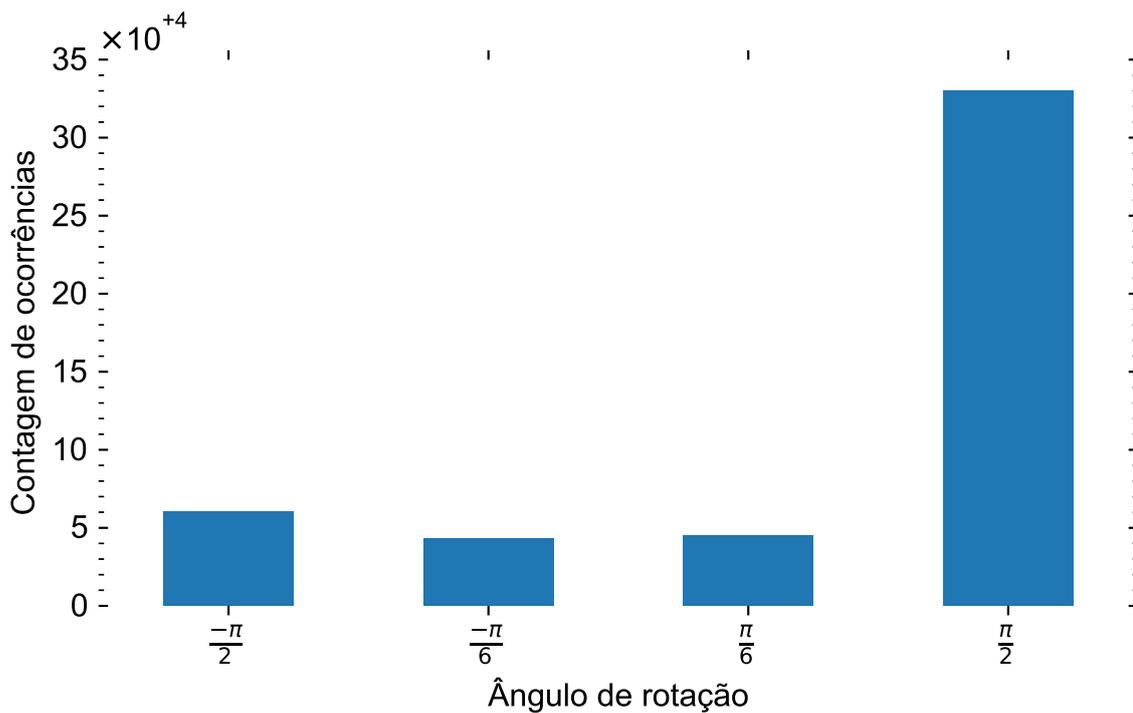
BD-Rate variam pouco em relação ao que foi observado para as 7 imagens USP-SIPI (Tabela 9). Entretanto, constata-se que apesar de entregar um resultado benéfico em termos de PSNR, o resultado em termos de ganhos médios de SSIM foi negativo para a SDCT-BT. Situação oposta ao que foi observado para a S<sup>3</sup>DCT, que entregou um resultado negativo em termos de PSNR e positivo em termos de SSIM.

Figura 23 – Histograma de escolha de ângulos para a SDCT-BT: o ângulo nulo foi escolhido em 41 % das sub-bandas criadas pelo algoritmo.



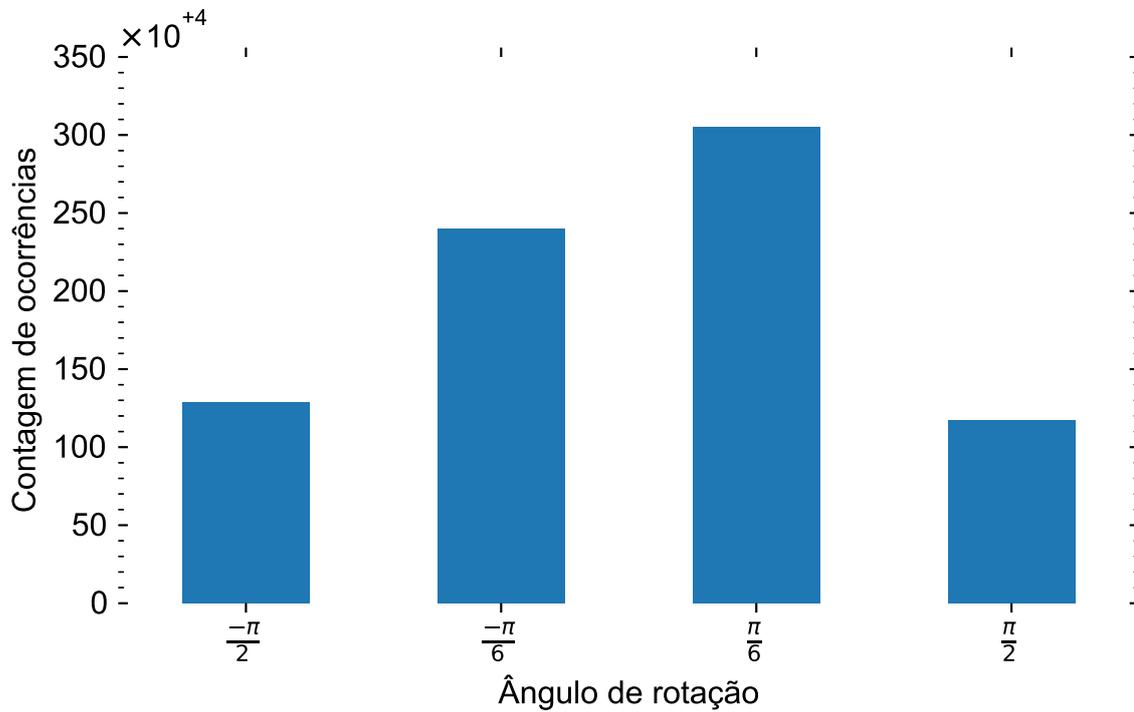
Fonte: O autor (2019).

Figura 24 – Histograma de escolha de ângulos para a SDCT-AM.



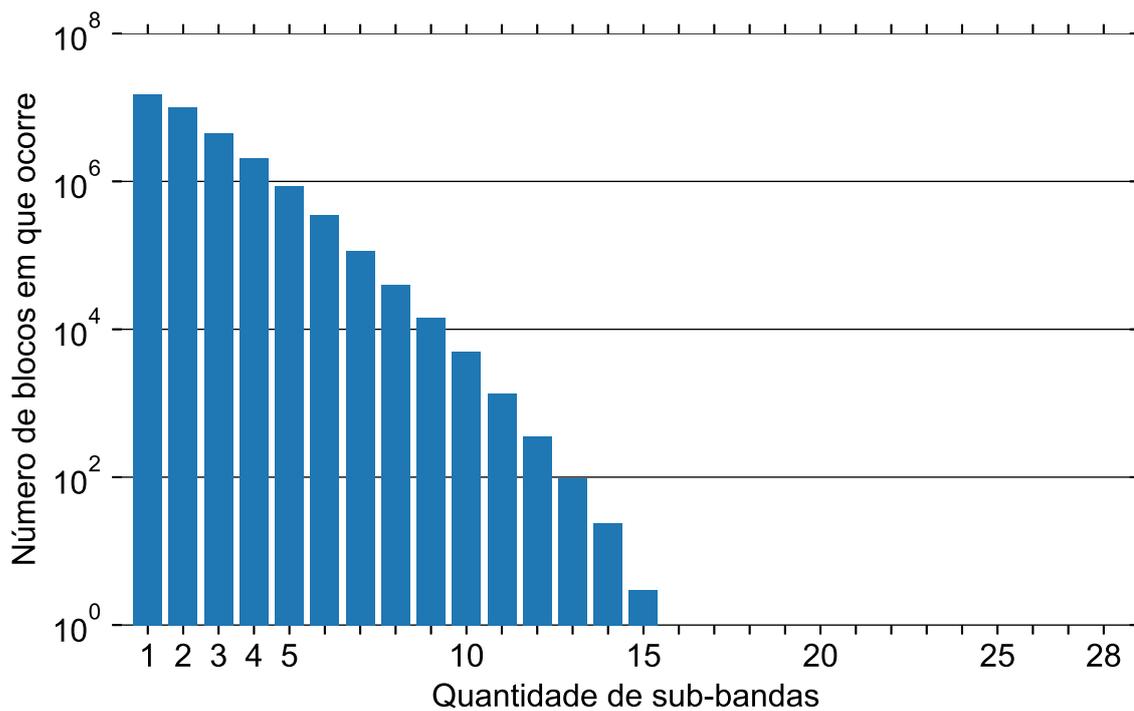
Fonte: O autor (2019).

Figura 25 – Histograma de escolha de ângulos para a S<sup>3</sup>DCT .



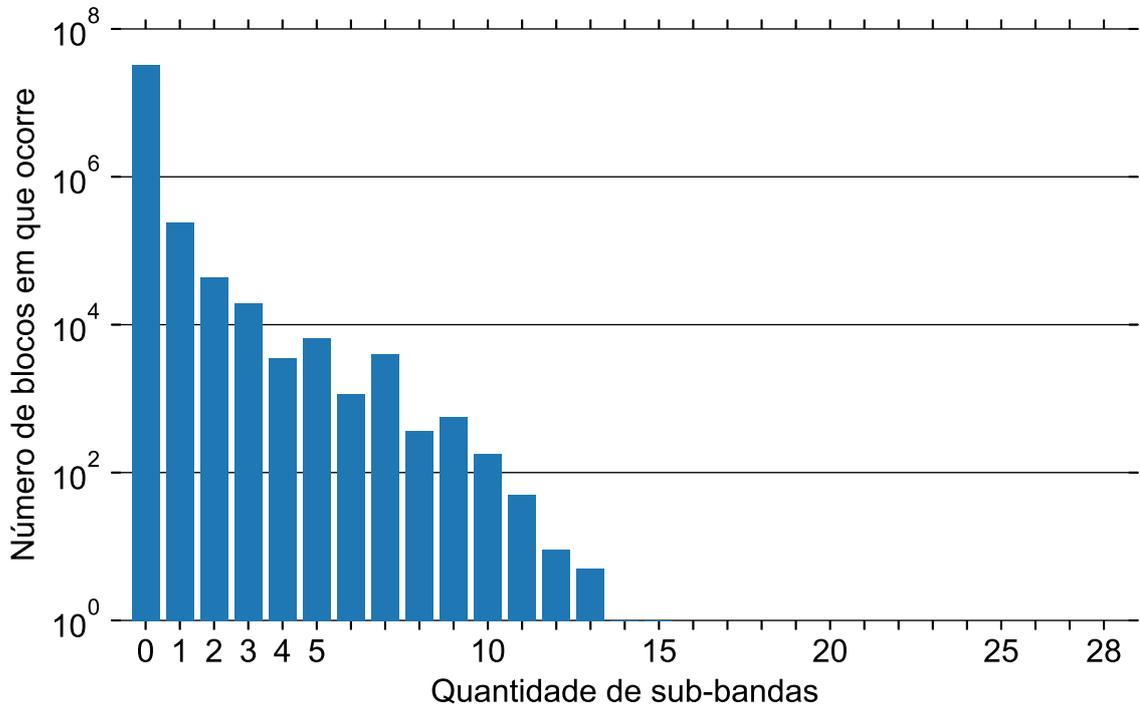
Fonte: O autor (2019).

Figura 26 – Histograma de quantidade de sub-bandas por bloco para a SDCT-BT: uma única sub-banda é utilizada em 45,31 % dos blocos processados.



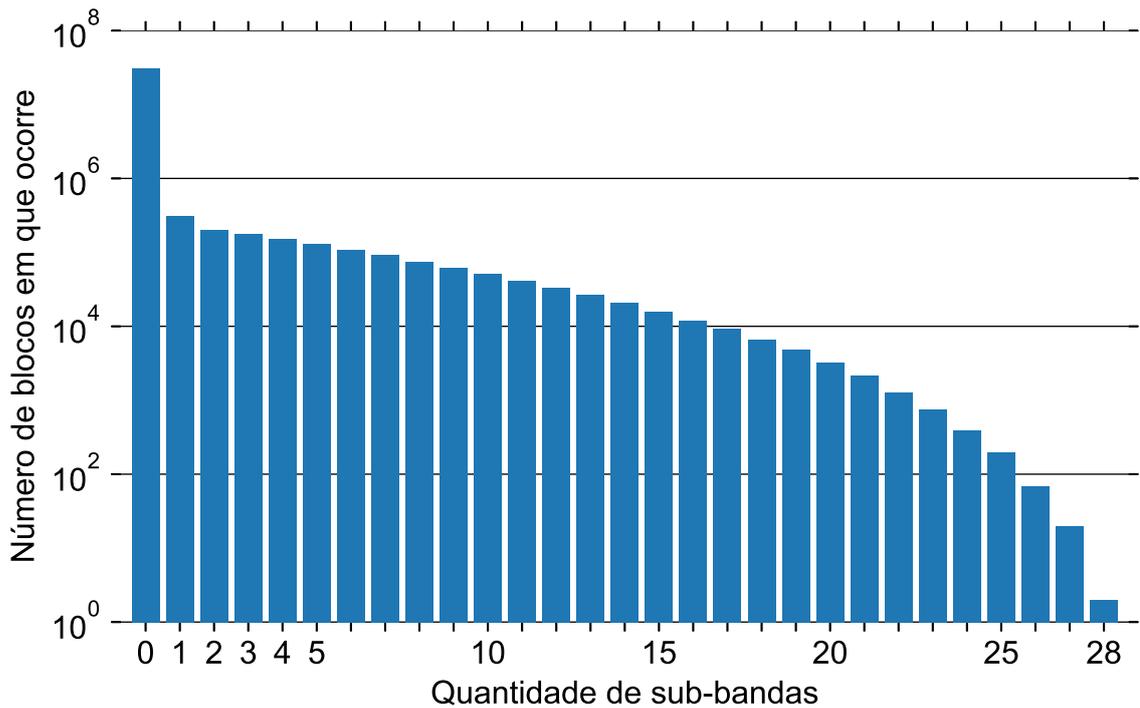
Fonte: O autor (2019).

Figura 27 – Histograma de quantidade de sub-bandas por bloco para a SDCT-AM: a 2D-DCT usual (0 sub-bandas) é escolhida em 99,02 % dos blocos processados.



Fonte: O autor (2019).

Figura 28 – Histograma de quantidade de sub-bandas por bloco para a S<sup>3</sup>DCT: a 2D-DCT usual (0 sub-bandas) é escolhida em 95,29 % dos blocos processados.



Fonte: O autor (2019).

Tabela 13 – Quadro comparativo geral com as particularidades de cada algoritmo estudado.

<b>Transformada</b>	<b>Otimização</b>	$Q_\theta$	<b>Fator</b>	<b>BD-Rate</b>	<b>BD-SSIM</b>
<b>S<sup>3</sup>DCT</b>	Analítica	$\{-\pi/2, -\pi/6, \pi/6, \pi/2\}$	1,39	0,09 %	+0,00003
<b>SDCT-1</b>	Busca exaustiva	$\{0, \pi/16, 2\pi/16, \dots, 7\pi/16\}$	3,46	-2,73 %	+0,00087
<b>SDCT-BT</b>	Busca exaustiva	$\{0, \pi/8, \pi/4, 3\pi/8\}$	8,43	-0,38 %	-0,00027
<b>SDCT-AM</b>	Analítica iterativa	$\{-\pi/2, -\pi/6, \pi/6, \pi/2\}$	26,35	-0,08 %	+0,00001

Fonte: O autor (2019).

## 5.6 QUADRO COMPARATIVO GERAL

Na Tabela 13 é mostrado um resumo dos principais resultados apresentados nesta seção para cada uma das transformadas: fator de tempo para codificação<sup>2</sup> e os ganhos médios nas figuras do BD-Rate e BD-SSIM para as 886 imagens do banco UCID, sempre em relação ao que seria obtido caso fosse empregada a 2D-DCT usual e mantidas as demais características gerais do CODEC.

<sup>2</sup> Fator de multiplicação do tempo de codificação utilizando o algoritmo em relação ao tempo de codificação utilizando a 2D-DCT usual.

## 6 CONCLUSÕES

Neste capítulo são apresentadas as considerações conclusivas desta dissertação, com a indicação de oportunidades para a realização de pesquisas futuras.

### 6.1 CONSIDERAÇÕES FINAIS

Neste trabalho, foi apresentado um conjunto de melhorias voltadas a estabelecer uma estratégia eficiente para a codificação e decodificação de imagens estáticas utilizando a transformada discreta manobrável do cosseno (SDCT). Como ponto de partida, foram utilizados os esquemas inicialmente propostos por Fracastoro *et al.*:  $S^3$ DCT, SDCT-1, SDCT-AM e SDCT-BT. Estes foram os algoritmos utilizados no processo de otimização RD para escolha dos ângulos de rotação.

Em um primeiro momento, nossos experimentos mostraram que, apesar dos ganhos descritos na literatura para a codificação de imagens utilizando a SDCT, havia ainda uma grande perda de desempenho a ser superada em relação à alternativa já consolidada e amplamente utilizada: a codificação padrão JPEG, especialmente para maiores taxas de compressão, em que as taxas de bits por pixel são mais baixas. A ausência de uma forma eficiente para codificar, sem perdas adicionais, os coeficientes esparsificados pela transformada, foi identificada como sendo o principal gargalo a ser superado.

Neste sentido, foi proposta a utilização de um codificador aritmético baseado na solução empregada pelo padrão JPEG: o QM-coder. Este codificador aritmético binário adaptativo inclui, em sua fase de binarização, sinalizadores para codificar de forma eficiente longas cadeias de coeficientes nulos e final antecipado de blocos, o que contribuiu para uma significativa melhora do desempenho para taxas de compressão mais altas. Além disso, alterações foram propostas no QM-coder de modo que fosse possível realizar uma compressão adicional sem perdas também para os ângulos de rotação. Estas melhorias na fase de codificação de entropia já foram suficientes para observar ganhos em relação ao JPEG baseline (com codificação Huffman) para todas as taxas de compressão. Entretanto, os ganhos ainda continuavam tímidos e dependentes da taxa de compressão e imagem processada quando o codificador SDCT era comparado com um codificador JPEG com codificação aritmética.

Uma série de melhorias adicionais foram propostas em seguida para obter ganhos incrementais para o codificador SDCT. A quantização dos coeficientes baseada em passo único foi substituída por uma tabela de passos baseada em características do sistema visual humano, o que foi possível graças a uma inversão de ordem entre as etapas de rotação de autovetores e quantização de coeficientes. Foi proposta também uma melhor codificação dos coeficientes DC de cada bloco a partir de sua codificação diferencial em relação ao bloco anterior, o que resultou

em ganhos adicionais médios da ordem de 1,6 % em termos de redução de taxa de bits.

Explorando a simetria existente entre os vetores de base pertencentes a um determinado par a ser rotacionado pela SDCT, foi proposta a utilização de uma nova faixa de valores para quantização uniforme dos ângulos a serem utilizados nos algoritmos de otimização RD (para as SDCT-1 e SDCT-BT), com o estreitamento da faixa original de  $[0, \pi)$  para  $[0, \pi/2)$ , em radiano. Foi realizada também uma redução na quantidade de níveis de quantização de 8 para 4 para os ângulos do algoritmo  $S^3$ DCT, SDCT-BT e SDCT-AM, obtendo conseqüentemente uma redução do tempo necessário para o processo de escolha dos ângulos e uma menor taxa de bits referente aos ângulos nestas transformadas.

Após a utilização das melhorias propostas, os resultados obtidos pelo CODEC proposto foram comparados com aqueles que seriam obtidos utilizando um esquema semelhante, porém com a transformada discreta do cosseno usual ao invés de sua versão manobrável. Ficou demonstrado, para um universo de 893 imagens de teste utilizadas, que o algoritmo da SDCT-1, apesar de conceitualmente mais simples, entregou os maiores ganhos. Por outro lado, as alternativas que exploram a utilização de mais de um ângulo por bloco entregaram uma menor vantagem, ou mesmo pequenas perdas de desempenho, em troca de uma maior complexidade computacional.

Além disso, mostrou-se que o algoritmo da SDCT-BT resultou na utilização de um ângulo único para mais da metade dos blocos processados, enquanto os algoritmos da SDCT-AM e  $S^3$ DCT resultaram na escolha pelo uso da 2D-DCT usual para a ampla maioria dos casos. Desta forma, os algoritmos baseados na SDCT que fazem uso de vários ângulos de rotação por bloco não entregaram ganhos de desempenho que justificassem seu uso ao invés da SDCT-1. Além disso, os algoritmos da SDCT-AM e  $S^3$ DCT não justificaram seu uso nem mesmo quando comparados com o codificador baseado na transformada usual.

Como vimos, se por um lado o uso de mais de um ângulo por bloco entrega uma maior capacidade de esparsificação dos coeficientes no domínio da transformada, por outro lado implica na necessidade de transmissão dos valores destes ângulos. Para reduzir o impacto destes dados adicionais na taxa de bits resultante, os ângulos são quantizados em alguns poucos valores (oito ou quatro), o que implica como contrapartida em um pior resultado em termos de distorção, já que os ângulos efetivamente utilizados não são ótimos. A utilização, neste trabalho, de uma estratégia de codificação de entropia mais próxima ao estado da arte para os coeficientes, resultou em uma maior importância relativa dada ao acréscimo de taxa devido aos ângulos, conseqüentemente dificultando a obtenção de bons resultados com a utilização dos algoritmos SDCT-BT, SDCT-AM e  $S^3$ DCT.

## 6.2 TRABALHOS FUTUROS

Esta dissertação apresentou uma série de estratégias empregadas com fins de melhorar o desempenho RD entregue por um CODEC de imagem baseado na SDCT. Entretanto, ainda

existem alguns caminhos em aberto a serem explorados que podem levar a ganhos adicionais interessantes. Destacamos a seguir algumas oportunidades de pesquisa futura:

- Buscar por formas de tornar mais eficiente a codificação dos ângulos de rotação: investigar e explorar possíveis correlações existentes entre os ângulos de rotação escolhidos para blocos vizinhos na mesma imagem. Isto pode ser feito com a utilização de contextos adicionais para o codificador QM-coder que permitam fazer uma modelagem para os ângulos com base em cadeias de Markov de ordem 2 ou maior. Entretanto, para minimizar os efeitos do estágio de aprendizagem do codificador, é importante definir um método que permita inicializar as estimativas de probabilidade de cada contexto com valores iniciais adequados, por exemplo, investigando probabilidades condicionais observadas nos ângulos escolhidos quando o codificador proposto neste trabalho é utilizado para um banco de imagens extenso;
- Estender a utilização da SDCT, em especial da SDCT-1, aliada às melhorias propostas neste trabalho, para outros cenários, como, por exemplo, a compressão de imagens multi-dimensionais, imagens do tipo *lenslet light field* ou a compressão de vídeos (utilização em CODEC HEVC).
- Investigar os efeitos do uso de uma aproximação para a transformada discreta do cosseno, buscando reduções nos tempos necessários para realizar a codificação utilizando os algoritmos baseados na SDCT.

## REFERÊNCIAS

- AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transform. **IEEE Transactions on Computers**, [S.l.], C-23, n. 1, p. 90–93, Jan 1974. ISSN 0018-9340.
- AHMED, N.; RAO, K. R. **Orthogonal transforms for digital signal processing**. [S.l.]: Springer Science & Business Media, 2012.
- ANNABY, M. H.; RUSHDI, M. A.; NEHARY, E. A. Image encryption via discrete fractional Fourier-type transforms generated by random matrices. **Signal Processing: Image Communication**, [S.l.], v. 49, p. 25–46, Nov 2016.
- ASSOCIATION, A. D. TIFF Revision 6.0. **Internet publication: <http://www.adobe.com/Support/TechNotes.html>**, [S.l.], 1992.
- AZIM, A. W.; GUENNEC, Y. L.; MAURY, G. Spectrally augmented hartley transform precoded asymmetrically clipped optical ofdm for vlc. **IEEE Photonics Technology Letters**, [S.l.], v. 30, n. 23, p. 2029–2032, Dec 2018.
- BAIG, S. et al. Closed-form ber expression for fourier and wavelet transform-based pulse-shaped data in downlink noma. **IEEE Communications Letters**, [S.l.], v. 23, n. 4, p. 592–595, April 2019.
- BISWAL, S.; BISWAL, M.; MALIK, O. P. Hilbert huang transform based online differential relay algorithm for a shunt-compensated transmission line. **IEEE Transactions on Power Delivery**, [S.l.], v. 33, n. 6, p. 2803–2811, Dec 2018.
- BJONTEGAARD, G. Calculation of average PSNR differences between RD-curves. **VCEG-M33**, [S.l.], 2001.
- BLAHUT, R. E. **Algebraic codes for data transmission**. [S.l.]: Cambridge University Press, 2003. ISBN 0521553741.
- BLAHUT, R. E. **Fast algorithms for signal processing**. S.l.: Cambridge University Press, 2010.
- BLUMENSATH, T.; DAVIES, M. E. Iterative thresholding for sparse approximations. **Journal of Fourier analysis and Applications**, Springer, [S.l.], v. 14, n. 5-6, p. 629–654, 2008.
- BONDY, J. A.; MURTY, U. S. R. **Graph Theory with Applications**. [S.l.]: Elsevier, 1976.
- BRACEWELL, R. N. Discrete hartley transform. **JOSA**, Optical Society of America, [S.l.], v. 73, n. 12, p. 1832–1835, 1983.
- CAMPELLO DE SOUZA, R. M. et al. Trigonometry in finite fields and a new Hartley transform. In: **1998 IEEE International Symposium on Information Theory, Proceedings**. [S.l.]: [s.n.], 1998. p. 293.
- CANDAN, C.; KUTAY, M. A.; OZAKTAS, H. M. The discrete fractional Fourier transform. **IEEE Transactions on Signal Processing**, [S.l.], v. 48, n. 5, p. 1329–1337, May 2000. ISSN 1053-587X.

- CHANG, C.-L.; GIROD, B. Direction-adaptive partitioned block transform for image coding. In: **2008 15th IEEE International Conference on Image Processing**. [S.l.]: [s.n.], 2008. p. 145–148.
- CHUNG, F. R. K. **Spectral Graph Theory**. [S.l.]: American Mathematical Society, 1997.
- CIONINI, A. et al. Engineering graph-based models for dynamic timetable information systems. **Journal of Discrete Algorithms**, [S.l.], v. 46-47, p. 40–58, 2017. ISSN 1570-8667.
- CIZEK, V. Discrete hilbert transform. **IEEE Transactions on Audio and Electroacoustics**, IEEE, [S.l.], v. 18, n. 4, p. 340–343, 1970.
- COHEN, R. A. et al. Direction-adaptive transforms for coding prediction residuals. In: **2010 IEEE International Conference on Image Processing**. [S.l.]: [s.n.], 2010. p. 185–188.
- COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex Fourier series. **Mathematics of Computation**, American Mathematical Society, [S.l.], v. 19, n. 90, p. 297–301, 1965. ISSN 00255718, 10886842.
- DIESTEL, R. **Graph Theory**. [S.l.]: Springer, 1997. (Graduate Texts in Mathematics, 173).
- DRÉMEAU, A. et al. Sparse optimization with directional dct bases for image compression. In: **2010 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.]: [s.n.], 2010. p. 1290–1293.
- FEKRI, F. et al. Block error correcting codes using finite-field wavelet transforms. **IEEE Transactions on Signal Processing**, [S.l.], v. 54, n. 3, p. 991–1004, Mar 2006. ISSN 1053-587X.
- FRACASTORO, G. **Design and Optimization of Graph Transform for Image and Video Compression, 2017**. Tese (Doutorado em Engenharia Eletrônica) — Politecnico di Torino, Itália, 2017.
- FRACASTORO, G.; FOSSON, S. M.; MAGLI, E. Steerable Discrete Cosine Transform. **IEEE Transactions on Image Processing**, [S.l.], v. 26, n. 1, p. 303–314, jan. 2017. ISSN 1057-7149.
- FRACASTORO, G.; MAGLI, E. Steerable Discrete Cosine Transform. In: **2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)**. [S.l.]: [s.n.], 2015. p. 1–6.
- FRACASTORO, G.; MAGLI, E. Subspace-sparsifying steerable discrete cosine transform from graph fourier transform. In: IEEE. **2016 IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2016. p. 1534–1538.
- FRACASTORO, G.; MAGLI, E. Steerable Discrete Fourier Transform. **IEEE Signal Processing Letters**, [S.l.], v. 24, n. 3, p. 319–323, mar. 2017. ISSN 1070-9908.
- FRANCIS, A. et al. Extensions and applications of equitable decompositions for graphs with symmetries. **Linear Algebra and its Applications**, [S.l.], v. 532, p. 432 – 462, 2017. ISSN 0024-3795.
- GONZALES, C. A. et al. DCT coding for motion video storage using adaptive arithmetic coding. **Signal Processing: Image Communication**, [S.l.], v. 2, n. 2, p. 145–154, 1990.

GONZALEZ, R. C.; WOODS, R. E. **Data compression: the complete reference**. 4. ed. [S.l.]: Pearson, 2017.

HAJOMER, A. A. E.; YANG, X.; HU, W. Secure ofdm transmission precoded by chaotic discrete hartley transform. **IEEE Photonics Journal**, [S.l.], v. 10, n. 2, p. 1–9, April 2018.

HARRISON, W. K. The role of graph theory in system of systems engineering. **IEEE Access**, [S.l.], v. 4, p. 1716–1742, 2016. ISSN 2169-3536.

HEIL, C. E.; WALNUT, D. F. Continuous and discrete wavelet transforms. **SIAM review**, SIAM, [S.l.], v. 31, n. 4, p. 628–666, 1989.

HSUE, W. L.; CHANG, W. C. Real discrete fractional Fourier, Hartley, generalized Fourier and generalized Hartley transforms with many parameters. **IEEE Transactions on Circuits and Systems I: Regular Papers**, [S.l.], v. 62, n. 10, p. 2594–2605, Oct 2015.

KAMISLI, F.; LIM, J. S. Transforms for the motion compensation residual. In: **2009 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.]: [s.n.], 2009. p. 789–792.

KIM, W.; NARANG, S. K.; ORTEGA, A. Graph based transforms for depth video coding. In: **2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.]: [s.n.], 2012. p. 813–816.

KIM, Y. K.; HE, Z.; MITRA, S. K. A novel linear source model and a unified rate control algorithm for h. 263/mpeg-2/mpeg-4. In: IEEE. **2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)**. [S.l.], 2001. v. 3, p. 1777–1780.

KOTENKO, I.; DOYNIKOVA, E.; CHECHULIN, A. Security metrics based on attack graphs for the olympic games scenario. In: **2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing**. [S.l.]: [s.n.], 2014. p. 561–568. ISSN 1066-6192.

KUZIAK, D.; YERO, I. G.; RODRIGUEZ-VELAZQUEZ, J. A. On the strong metric dimension of corona product graphs and join graphs. **Discrete Applied Mathematics**, [S.l.], v. 161, n. 7, p. 1022 – 1027, 2013. ISSN 0166-218X.

LANDRÉ, J.; TRUCHETET, F. Optimizing signal and image processing applications using intel libraries. In: **Eighth International Conference on Quality Control by Artificial Vision**. [S.l.]: International Society for Optics and Photonics, 2007. v. 6356, p. 63560O.

LIMA, J. B.; CAMPELLO DE SOUZA, R. M. Finite field trigonometric transforms. **Applicable Algebra in Engineering, Communication and Computing**, [S.l.], v. 22, n. 5-6, p. 393–411, Dec 2011.

LIMA, J. B.; CAMPELLO DE SOUZA, R. M. Closed-form Hermite–Gaussian-like number-theoretic transform eigenvectors. **Signal Processing**, [S.l.], v. 128, n. Complete, p. 409–416, Nov 2016.

LIMA, J. B.; MADEIRO, F.; SALES, F. J. R. Encryption of medical images based on the cosine number transform. **Signal Processing: Image Communication**, Elsevier BV, [S.l.], v. 35, p. 1–8, Jul 2015.

- LIU, Y. et al. Single-channel color image encryption algorithm based on fractional Hartley transform and vector operation. **Multimedia Tools and Applications**, [S.l.], v. 74, n. 9, p. 3171–3182, May 2015.
- LU, G.; LIU, J.; YAN, P. Graph-based structural change detection for rotating machinery monitoring. **Mechanical Systems and Signal Processing**, [S.l.], v. 99, p. 73 – 82, 2018. ISSN 0888-3270.
- LU, G.; XIAO, M.; WEI, P. Adaptive short time fractional Fourier transform for time-frequency segmentation. **Electronics Letters**, [S.l.], v. 52, n. 8, p. 615–617, Apr 2016.
- MARPE, D.; SCHWARZ, H.; WIEGAND, T. Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard. **IEEE Transactions on circuits and systems for video technology**, [S.l.], v. 13, n. 7, p. 620–636, 2003.
- MARTUCCI, S. A. Symmetric convolution and the discrete sine and cosine transforms. **IEEE Transactions on Signal Processing**, [S.l.], v. 42, n. 5, p. 1038–1051, May 1994.
- MERRIS, R. Laplacian matrices of graphs: a survey. **Linear Algebra and its Applications**, [S.l.], v. 197, p. 143–176, 1994.
- MERRIS, R. Laplacian graph eigenvectors. **Linear Algebra and its Applications**, [S.l.], v. 278, n. 1, p. 221–236, 1998.
- ORTEGA, A. et al. Graph signal processing: overview, challenges, and applications. **Proceedings of the IEEE**, [S.l.], v. 106, n. 5, p. 808–828, May 2018.
- PEI, S.-C.; YEH, M.-H. The discrete fractional cosine and sine transforms. **IEEE Transactions on Signal Processing**, [S.l.], v. 49, n. 6, p. 1198–1207, Jun 2001.
- PENNEBAKER, W. B. **JPEG: Still Image Data Compression Standard**. [S.l.]: ISO/IEC, 1993. Disponível em: <<http://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/01/89/18902.html>>.
- POULARIKAS, A. D. **Transforms and Applications Handbook**. 3. ed. [S.l.]: CRC Press, 2018.
- RIBEIRO, G. B. Mestrado em Engenharia Eletrônica, **Um Estudo sobre Operadores para Deslocamento de Sinais sobre Grafos, 2018**. Brasil: [s.n.], 2018.
- SALOMON, D. **Data compression: the complete reference**. [S.l.]: Springer Science & Business Media, 2004.
- SCHAEFER, G.; STICH, M. Ucid: An uncompressed color image database. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Storage and Retrieval Methods and Applications for Multimedia 2004**. [S.l.], 2003. v. 5307, p. 472–481.
- SCHAFER, R. W.; OPPENHEIM, A. V. **Discrete-Time Signal Processing**. 3. ed. [S.l.]: Prentice Hall, 2010.
- SERBES, A. Compact fractional Fourier domains. **IEEE Signal Processing Letters**, [S.l.], v. 24, n. 4, p. 427–431, April 2017.
- SHANG, X. et al. Weighting Quantization Matrices for HEVC/H. 265-coded RGB Videos. **IEEE Access**, [S.l.], 2019.

- SHI, Y. et al. Statistical graph classification in intelligent mathematics problem solving system for high school student. In: **2017 12th International Conference on Computer Science and Education (ICCSE)**. [S.l.]: [s.n.], 2017. p. 645–650.
- SHUMAN, D. I. et al. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. **IEEE Signal Processing Magazine**, [S.l.], v. 30, n. 3, p. 83–98, May 2013.
- SLUSARCZYK, G. et al. An extended hierarchical graph-based building model for design and engineering problems. **Automation in Construction**, [S.l.], v. 74, p. 95–102, 2017. ISSN 0926-5805.
- STRANG, G. The discrete cosine transform. **SIAM Review**, [S.l.], v. 41, n. 1, p. 135–147, March 1999.
- SULLIVAN, G. J.; WIEGAND, T. et al. Rate-distortion optimization for video compression. **IEEE signal processing magazine**, [S.l.], v. 15, n. 6, p. 74–90, 1998.
- SZE, V.; BUDAGAVI, M. High throughput CABAC entropy coding in HEVC. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 22, n. 12, p. 1778–1791, 2012.
- WANG, C.-Y.; LEE, S.-M.; CHANG, L.-W. Designing JPEG quantization tables based on human visual system. **Signal Processing: Image Communication**, [S.l.], v. 16, n. 5, p. 501–506, 2001.
- WANG, E. et al. Intel math kernel library. In: **High-Performance Computing on the Intel® Xeon Phi™**. [S.l.]: Springer, 2014. p. 167–188.
- WATSON, A. B. Perceptual Optimization of DCT Color Quantization Matrices. In: **ICIP (1)**. [S.l.]: [s.n.], 1994. p. 100–104.
- WEBER, A. G. The USC-SIPI image database version 5. **USC-SIPI Report**, [S.l.], v. 315, p. 1–24, 1997.
- WITTEN, I. H.; NEAL, R. M.; CLEARY, J. G. Arithmetic coding for data compression. **Communications of the ACM**, [S.l.], v. 30, n. 6, p. 520–540, 1987.
- XU, H.; XU, J.; WU, F. Lifting-based directional dct-like transform for image coding. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 17, n. 10, p. 1325–1335, Oct 2007.
- YEO, C. et al. Mode-dependent transforms for coding directional intra prediction residuals. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 22, n. 4, p. 545–554, April 2012.
- ZENG, B.; FU, J. Directional discrete cosine transforms—a new framework for image coding. **IEEE Transactions on Circuits and Systems for Video Technology**, [S.l.], v. 18, n. 3, p. 305–313, March 2008.
- ZHANG, C.; FLORENCIO, D. Analyzing the optimality of predictive transform coding using graph-based models. **IEEE Signal Processing Letters**, [S.l.], v. 20, n. 1, p. 106–109, January 2013.

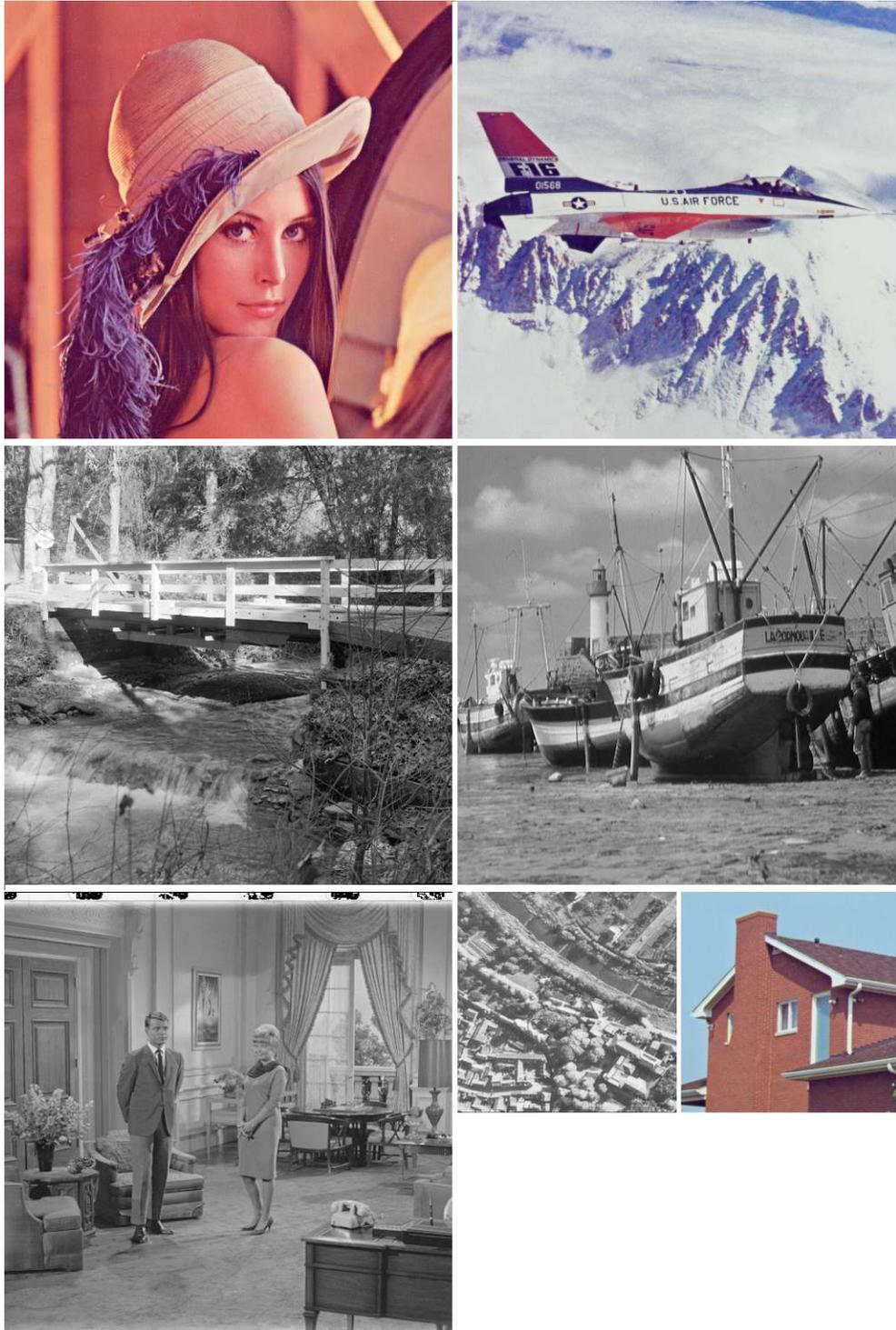
ZHENG, A. X.; DUNAGAN, J.; KAPOOR, A. Active graph reachability reduction for network security and software engineering. In: **Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)**. [S.l.: s.n.], 2011. p. 1750–1756. ISBN 978-1-57735-514-4.

ZHENG, P.; HUANG, J. Efficient encrypted images filtering and transform coding with walsh-hadamard transform and parallelization. **IEEE Transactions on Image Processing**, [S.l.], v. 27, n. 5, p. 2541–2556, May 2018.

Zhou Wang et al. Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, [S.l.], v. 13, n. 4, p. 600–612, abr. 2004. ISSN 1057-7149.

## APÊNDICE A – IMAGENS UTILIZADAS NESTE TRABALHO

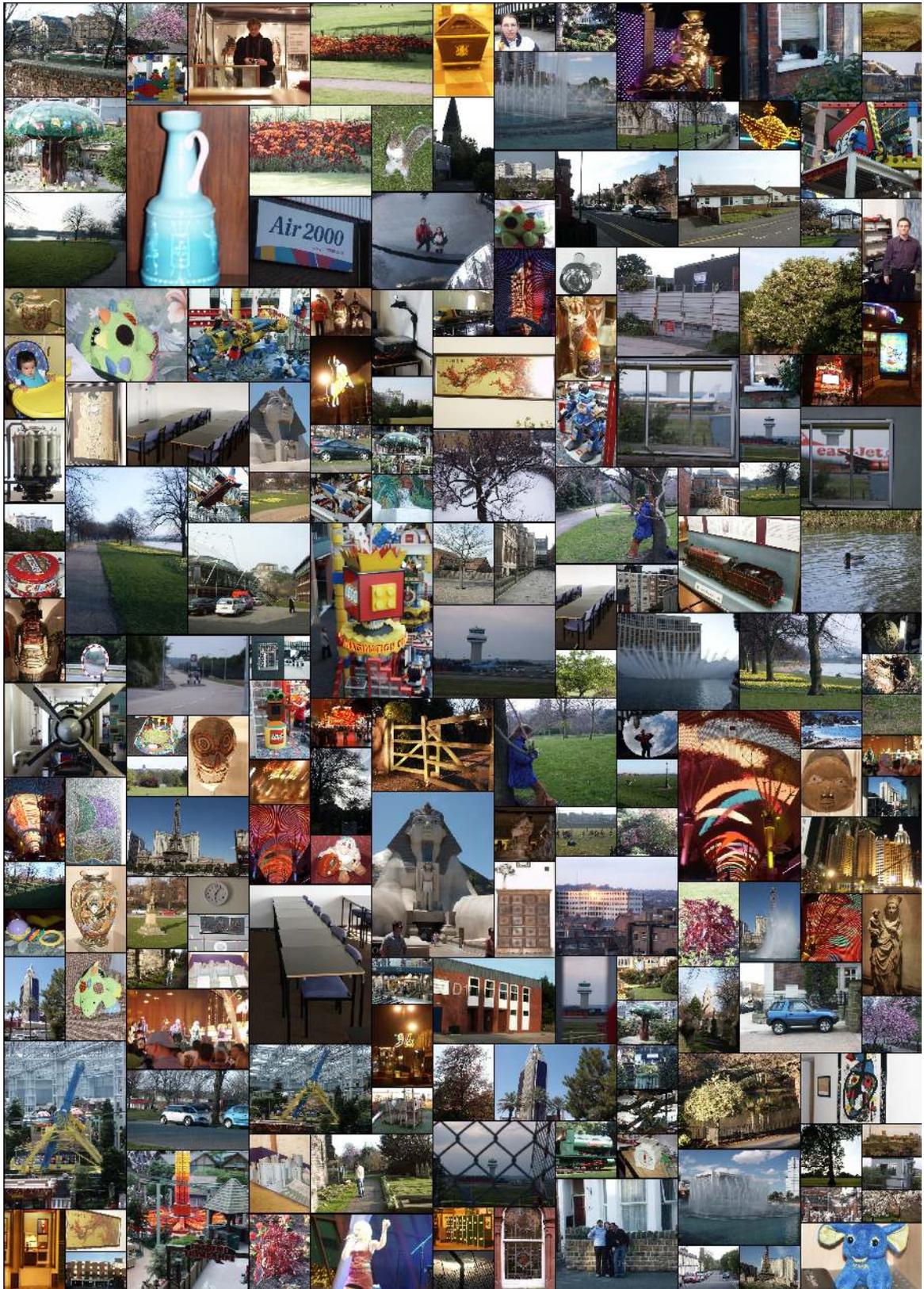
Figura A.1 – Imagens retiradas do banco USC-SIPI que foram utilizadas neste trabalho. Da esquerda para a direita e de cima para baixo: Lena, F16, Bridge, Boat, Couple, Aerial e House.



Fonte: Weber (1997).

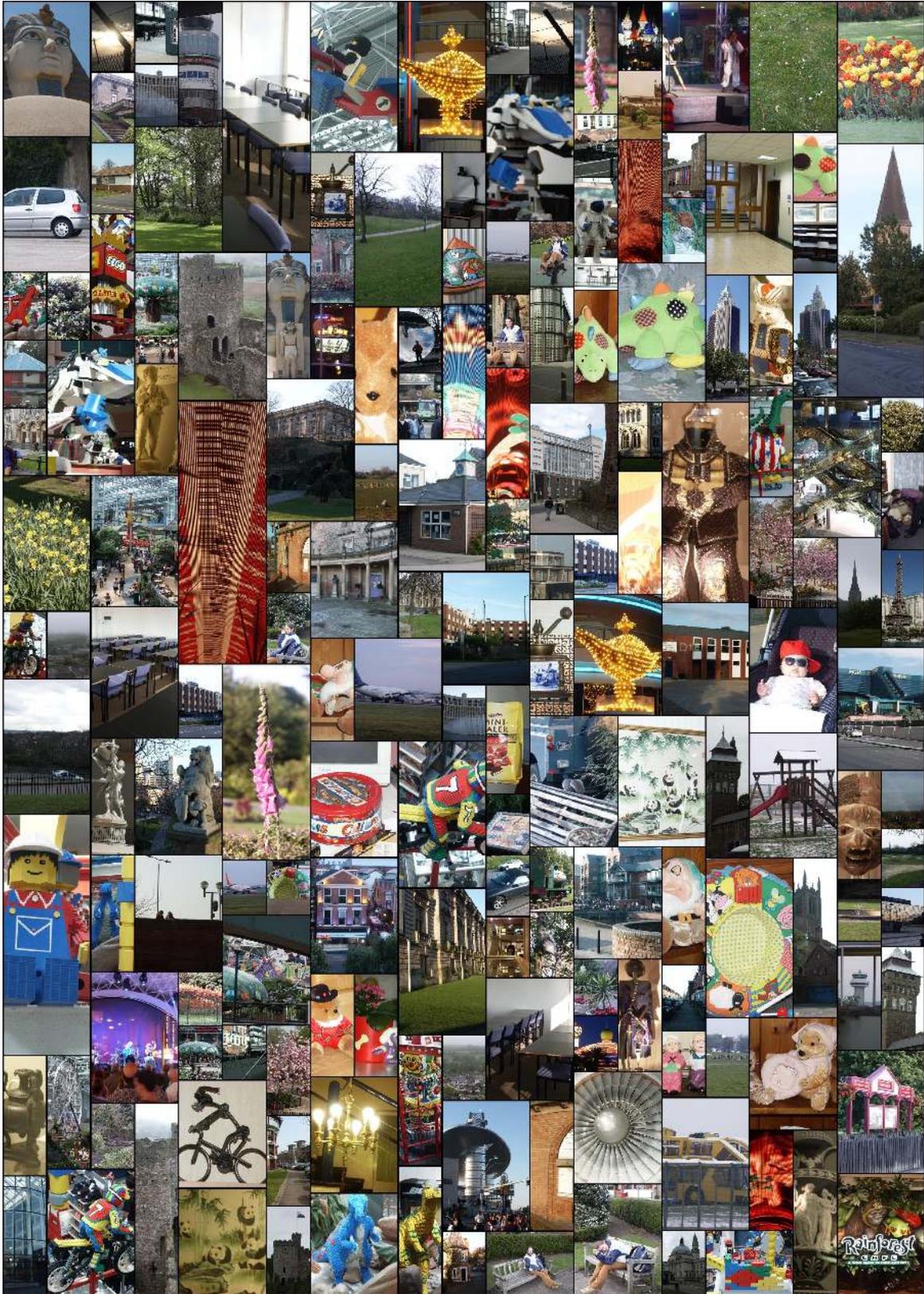


Figura A.3 – Montagem com miniaturas das imagens retiradas do banco UCID que foram utilizadas neste trabalho (parte 2 de 5).



Fonte: Compilação do próprio autor a partir de imagens retiradas de Schaefer e Stich (2003).

Figura A.4 – Montagem com miniaturas das imagens retiradas do banco UCID que foram utilizadas neste trabalho (parte 3 de 5).



Fonte: Compilação do próprio autor a partir de imagens retiradas de Schaefer e Stich (2003).

Figura A.5 – Montagem com miniaturas das imagens retiradas do banco UCID que foram utilizadas neste trabalho (parte 4 de 5).



Fonte: Compilação do próprio autor a partir de imagens retiradas de Schaefer e Stich (2003).

