



Universidade Federal de Pernambuco
Centro de Ciências Exatas e da Natureza
Programa de Pós-Graduação em Estatística

ANABETH PETRY RADÜNZ

**APROXIMAÇÕES DE BAIXA COMPLEXIDADE PARA A
TRANSFORMADA DISCRETA DE KARHUNEN-LOËVE: Projeto,
Algoritmos Rápidos e Compressão de Imagens**

Recife

2019

Anabeth Petry Radünz

**APROXIMAÇÕES DE BAIXA COMPLEXIDADE PARA A
TRANSFORMADA DISCRETA DE KARHUNEN-LOÈVE:
Projeto, Algoritmos Rápidos e Compressão de Imagens**

Dissertação apresentada ao Programa de Pós-Graduação em Estatística do Centro de Ciências Exatas e da Natureza da Universidade Federal de Pernambuco, como requisito parcial à obtenção do título de mestre em Estatística.

Área de Concentração: Estatística Aplicada

Orientador: Dr. Renato J. Cintra

Coorientador: Dr. Fábio Mariano Bayer

Recife

2019

Catálogo na fonte
Bibliotecária Leonice Cavalcante CRB 4-1996

R124a Radünz, Anabeth Petry
Aproximações de baixa complexidade para a transformada discreta Karhunen-Loève: projeto, algoritmos rápidos e compressão de imagens / Anabeth Petry Radünz. – 2019.
95 f.: fig., tab.

Orientador: Renato Cintra
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CCEN. Estatística. Recife, 2019.
Inclui referências e apêndice.

1. Análise de componentes principais. 2. Compressão de imagens. 3. Transformada de Karhunen-Loève. I. Cintra, Renato (orientador) II. Título.

519.5 CDD (22. ed.) UFPE-MEI 2019-36

ANABETH PETRY RADÜNZ

**APROXIMAÇÕES DE BAIXA COMPLEXIDADE PARA A TRANSFORMADA
DISCRETA DE KARHUNEN-LOÈVE: PROJETO, ALGORITMOS RÁPIDOS E
COMPRESSÃO DE IMAGENS**

Dissertação apresentada ao Programa de Pós-Graduação em Estatística da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Estatística.

Aprovada em: 21 de fevereiro de 2019.

BANCA EXAMINADORA

Prof.(º) Raydonal Ospina Martínez
UFPE

Prof.(º) Helio Magalhães de Oliveira
UFPE

Dr. Vitor de Andrade Coutinho
UFPE

À minha família:

Roberto, Ana, Amanda e Eduardo.

AGRADECIMENTOS

A Eduardo, pela cumplicidade, carinho e amor. Por ter “topado” encarar este desafio comigo e por estar ao meu lado em todos os momentos, me apoiando e encorajando sempre, tornando esta jornada um pouco mais fácil.

Aos meus pais e minha irmã, por tudo o que sou e pelo apoio incondicional. Se posso voar é porque sei que tenho para onde voltar, são minha base.

Aos amigos e colegas do PPGE, pelas inúmeras tardes, que pareciam não ter fim, de estudos. O caminho foi mais fácil com vocês ao meu lado. Também aos amigos da graduação e do tempo do colégio, por estarem sempre presentes, mesmo de longe.

Agradeço aos meus mentores. Ao meu orientador Prof. Renato Cintra, pelos ensinamentos, paciência e por me mostrar que posso ir cada vez mais longe. Ao meu coorientador, Prof. Fábio Bayer, por me incentivar a seguir na vida acadêmica e pela parceria de sempre. Agradeço também a todos os professores que passaram em minha vida, com certeza não estaria aqui se não fosse por alguns deles.

A CAPES, pelo apoio financeiro.

RESUMO

A análise de componentes principais (PCA) é frequentemente utilizada na descorrelação de dados e na redução de dimensionalidade. Por sua característica de compactação de energia em poucas componentes principais, encontram-se aplicações importantes da PCA no contexto de compressão de imagens. Na área de processamentos de sinais, a PCA é conhecida como transformada de Karhunen-Loève (KLT). Pelo fato de seu cômputo depender da matriz de variâncias e covariâncias do sinal de entrada, o uso da KLT em aplicações em tempo real é severamente restringido pela dificuldade do desenvolvimento de algoritmos rápidos que a implemente. Nesse contexto, esta dissertação propõe uma nova classe de transformadas de baixa complexidade que são obtidas por meio da aplicação de funções inteiras nos elementos da matriz da KLT. É dada ênfase às transformadas de comprimento 8 devido sua à ampla utilização em codificação de imagem e vídeo, como nos padrões JPEG e HEVC. Quando o sinal de entrada é um processo Markoviano de ordem I, o cômputo da KLT depende apenas do coeficiente de correlação ρ dos dados de entrada. Nesse sentido, obtemos transformadas aproximadas considerando os valores de ρ variando de zero até um. As transformadas ótimas, considerando algumas figuras de mérito que medem o poder de codificação e distância da aproximação proposta para a KLT exata, são exploradas em experimentos de compressão de imagens. Essa avaliação considera métricas tradicionais, tais como a relação sinal-ruído de pico e o índice de similaridade estrutural. De acordo com os resultados obtidos, mostrou-se que as transformadas aproximadas propostas possuem bom desempenho em aplicações de compressão de imagem e requerem baixo custo de implementação. Também são introduzidos algoritmos rápidos para as transformadas aproximadas propostas.

Palavras-chave: Análise de componentes principais. Compressão de imagens. Transformada de Karhunen-Loève. KLT aproximada. Transformada de baixo custo

ABSTRACT

Principal Component Analysis (PCA) is often used for data decorrelation and dimensionality reduction. Because of its energy compression characteristics in a few main components, there are important PCA applications in the framework of image compression. In the area of signal processing, PCA is known as the Karhunen-Loève transform (KLT). Due to the fact that its computation depends on the matrix of variances and covariances of the input signal, the use of the KLT in real time applications is severely constrained by the difficulty in developing fast algorithms to implement it. In this context, this dissertation proposes a new class of low-complexity transforms that are obtained through the application of integer functions to the elements of the KLT matrix. Emphasis is given to transforms of length 8 due to its wide use in image and video coding as in the JPEG and HEVC standards. When the input signal is a Markovian process of order I, the KLT computation depends only on the correlation coefficient ρ of the input data. In this sense, we obtain approximate transforms considering the values of ρ ranging from zero to one. The optimal transforms, considering some figures of merit that measure the coding power and distance of the proposed approximations to the exact KLT, are explored in image compression experiments. This evaluation considers traditional metrics, such as the peak signal-to-noise ratio and the structural similarity index. According to the results obtained, it was shown that the proposed approximate transforms perform well in image compression and require a low implementation cost. Fast algorithms are also introduced for the proposed approximate transforms.

Keywords: Principal component analysis. Image compression. Karhunen-Loève transform. Approximate KLT. Low-complexity transforms

LISTA DE FIGURAS

Figura 1 – Contextos para aplicação de transformadas aproximadas. . . .	18
Figura 2 – Valores de γ considerando a KLT para diferentes valores de ρ	43
Figura 3 – Separação do intervalos de ρ em grupos.	50
Figura 4 – Dendrograma do o ganho de codificação unificado.	52
Figura 5 – Diagrama de fluxo de sinal geral para T_6 , T_8 e T_{18}	62
Figura 6 – Diagrama de fluxo de sinal geral para T_{21} e T_{22}	62
Figura 7 – Diagrama de fluxo de sinal geral para T_{23}	63
Figura 8 – Blocos A e B dos diagramas de fluxo de sinal.	64
Figura 9 – Padrão zig-zag.	68
Figura 10 – Diagrama de blocos da operação de compressão de imagem. . .	68
Figura 11 – Imagens originais.	71
Figura 12 – Imagem <i>Lena</i> comprimida	72
Figura 13 – Diferença entre a Imagem <i>Lena</i> comprimida e a original . . .	73
Figura 14 – Imagem <i>Boat</i> comprimida	74
Figura 15 – Diferença entre a Imagem <i>Boat</i> comprimida e a original . . .	75
Figura 16 – Imagem <i>Grass</i> comprimida	76
Figura 17 – Diferença entre a Imagem <i>Grass</i> comprimida e a original . . .	77
Figura 18 – PSNR médio das imagens comprimidas	79
Figura 19 – MSSIM médio das imagens comprimidas	80

LISTA DE TABELAS

Tabela 1 – Aproximações para a DCT baseadas em funções inteiras . . .	34
Tabela 2 – Medidas de avaliação das transformadas aproximadas	37
Tabela 3 – Aproximações SKLT	39
Tabela 4 – Medidas de codificação e similaridade das SKLT	40
Tabela 5 – Aproximações RKLT	41
Tabela 6 – Medidas de codificação e similaridade das RKLT	42
Tabela 7 – Intervalos de α	43
Tabela 8 – Novas transformadas (função piso)	47
Tabela 9 – Novas transformadas (função teto)	48
Tabela 10 – Novas transformadas (função de truncamento)	48
Tabela 11 – Novas transformadas (função de arredondamento)	49
Tabela 12 – Medidas de codificação e similaridade	50
Tabela 13 – Medidas de codificação e similaridade após agrupamento . . .	53
Tabela 14 – Aproximadas propostas e suas respectivas matrizes diagonais	55
Tabela 15 – Comparação das medidas de codificação e similaridade	56
Tabela 16 – Complexidade aritmética dos algoritmos rápidos da DCT-8 .	58
Tabela 17 – Constantes necessárias para os algoritmos rápidos	61
Tabela 18 – Complexidade aritmética das transformadas	66
Tabela 19 – Medidas de qualidade das imagens	71
Tabela 20 – Valores do PSNR e MSSIM médios, considerando $r = 10$. . .	78

LISTA DE ALGORITMOS

Algoritmo 1 – Pseudocódigo para a busca de transformadas ótimas	46
Algoritmo 2 – Agrupamento <i>k-means</i> utilizando distância Euclidiana . . .	51

LISTA DE SÍMBOLOS

ρ	Coefficiente de correlação
\mathbf{x}	Vetor de entrada N -dimensional
\mathbf{y}	Vetor transformado N -dimensional
\mathbf{R}_x	Matriz de covariância de \mathbf{x}
$E(\cdot)$	Retorna o valor esperado ou esperança
$\mathbf{K}^{(\rho)}$	Representação da matriz da KLT para um valor de ρ definido
\mathbf{C}_N	Representação da matriz da DCT de comprimento N
\mathbb{N}	Conjunto dos números naturais
\mathbb{Z}	Conjunto dos números inteiros
\mathbf{A}	Matriz arbitrária
\mathbf{T}	Transformada de baixa complexidade
\top	Transposição matricial
\mathbf{S}	Matriz diagonal utilizada no processo de ortonormalização caso \mathbf{T} seja ortogonal
$\text{diag}(\cdot)$	Retorna a matriz diagonal
$\text{tr}(\cdot)$	Retorna traço de uma matriz quadrada
$\hat{\mathbf{S}}$	Aproximação para a matriz diagonal \mathbf{S} caso caso \mathbf{T} não seja ortogonal
$\text{int}(\cdot)$	Função inteira de arredondamento
$\text{Cg}(\cdot)$	Ganho de codificação unificado
$\eta(\cdot)$	Eficiência da transformada
$\text{MSE}(\cdot)$	Erro quadrático médio
$\epsilon(\cdot)$	Erro de energia total
$\delta(\cdot)$	Desvio de diagonalidade
\mathbf{K}	Transformada proposta com entradas apenas de elementos inteiros
$\hat{\mathbf{K}}$	Aproximação para a KLT baseada nas transformadas com entradas apenas de elementos inteiros
\mathcal{C}	Conjunto de elementos de baixa complexidade

$\mathcal{A}(\cdot)$ Complexidade aditiva de uma transformada
 $\mathcal{S}(\cdot)$ Quantidade de operações de deslocamento de *bits*

SUMÁRIO

1	INTRODUÇÃO	15
1.1	MOTIVAÇÃO	15
1.2	ESTADO DA ARTE	16
1.3	OBJETIVOS	18
1.4	ESTRUTURA DO DOCUMENTO	19
2	KLT E DCT	21
2.1	A TRANSFORMADA DE KARHUNEN-LOÈVE	21
2.1.1	KLT para sinais Markovianos de primeira ordem	22
2.2	DCT COMO APROXIMAÇÃO ASSINTÓTICA PARA A KLT	23
2.3	APROXIMAÇÕES PARA A DCT	26
2.3.1	SDCT	29
2.3.2	DCT Arredondada (RDCT)	30
2.3.3	DCT arredondada modificada (MRDCT)	31
2.3.4	Aproximações da DCT baseadas em funções inteiras	31
2.4	MEDIDAS DE AVALIAÇÃO DE DESEMPENHO	35
2.4.1	Ganho de codificação	35
2.4.2	Eficiência da transformada	35
2.4.3	Erro quadrático médio	36
2.4.4	Erro de energia total	36
2.4.5	Medidas de avaliação das transformadas aproximadas para DCT	36
3	APROXIMAÇÕES PARA A KLT	38
3.1	SKLT	38
3.2	RKLT	40
3.3	APROXIMAÇÕES BASEADAS EM FUNÇÕES INTEIRAS	42
3.4	OTIMIZAÇÃO	44
3.4.1	Função objetiva	44
3.4.2	Espaço de busca	45
3.4.3	Restrições	45
3.5	RESULTADOS	46
3.5.1	Refinamento dos resultados	49
4	ALGORITMOS RÁPIDOS	57

4.1	ALGORITMOS RÁPIDOS PROPOSTOS	58
4.2	COMPLEXIDADE COMPUTACIONAL	63
4.2.1	Complexidade aritmética dos algoritmos rápidos propostos . .	65
5	RESULTADOS NUMÉRICOS	67
5.1	COMPRESSÃO DE IMAGENS	67
5.2	MEDIDAS DE QUALIDADE DE IMAGEM	68
5.2.1	Erro quadrático médio	69
5.2.2	Relação sinal-ruído de pico	69
5.2.3	Índice de similaridade estrutural médio	69
5.2.4	Resultados	70
6	CONCLUSÕES	81
6.1	PRODUÇÃO ACADÊMICA	82
	REFERÊNCIAS	83
	APÊNDICE A – TRANSFORMADAS KLT INTEIRAS . . .	90
	APÊNDICE B – BANCO DE IMAGENS	92

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Estamos vivendo na era em que bilhões de dados são gerados todos os dias, a era *Big Data* [1,2]. Com o desenvolvimento cada vez maior da tecnologia de informação, a capacidade de gerarmos quantidades ainda maiores de dados cresce a cada dia em todos os domínios da ciência e da engenharia, incluindo ciências físicas, biológicas e biomédicas [3]. Aproximadamente 1.2 *zettabytes* (10^{21}) (neste trabalho, é utilizado ‘.’ como separador decimal) de dados são gerados todos os anos nestas áreas, desde experimentos científicos até postagens no Twitter [2]. Muitas ferramentas estatísticas, como aprendizado de máquina e mineração de dados, foram desenvolvidas para se analisar estes dados [3,4]. Porém, não há muitos avanços entre os estatísticos na elaboração de procedimentos inferenciais de baixo custo computacional. Em seu artigo “*On Some Useful ‘Inefficient’ Statistics*”, o Professor Frederick Mosteller (Universidade de Princeton) [5] já chamava a atenção dos avanços em estatísticas eficientes e testes mais poderosos sem considerar a economia computacional associada (“eficiência” computacional). Hoje em dia, com demandas computacionais cada vez maiores para análise de grandes massas de dados em tempo real, o desenvolvimento de métodos estatísticos de baixa complexidade computacional se faz de grande interesse.

A compressão de dados pode ser uma boa alternativa para a redução da dimensionalidade e útil para transmissão e armazenamento dos dados. O processo de compressão transforma dados de entrada em dados de saída com menor dimensão [6]. Há vários métodos para compressão de dados, mas todos eles se baseiam no mesmo princípio da redução ou remoção de redundância dos dados de entrada. Na terminologia de processamento de sinais, estes dados podem ser entendidos como sinais, e alguns exemplos são: áudio, vídeo, imagem, linguagem, multimídia, sonar, radar, registros de atividade biomédica/biológica, sequências de atributos, quantidades numéricas, entre outros dados [7].

A análise de componentes principais (PCA) [8] é comumente utilizada neste contexto de descorrelação de dados e redução de dimensionalidade [9,10]. A PCA envolve um procedimento matemático que transforma as variáveis correlacionadas em variáveis não-correlacionadas, que são denominadas de componentes principais. Usualmente, ape-

nas as primeiras componentes principais são utilizadas para representar grande parte da variabilidade (energia) dos dados (sinal). Essa característica de retenção de energia tem grande importância para a compressão de imagens, pois a PCA é capaz de preservar a imagem mesmo reduzindo a dimensionalidade dos dados [11].

1.2 ESTADO DA ARTE

As transformadas discretas têm um papel muito importante no contexto de processamento digital de sinais [12]. Elas são adequadas em aplicações a dados reais e complexos quando utilizamos computadores digitais, que são capazes apenas de realizar cálculos numéricos e finitos [13]. Quando aplicamos as transformadas discretas, podemos analisar os dados sob a perspectiva do domínio da transformada e gerar novas interpretações para estes dados. Como exemplo de transformadas discretas temos a transformada discreta de Fourier (DFT) que decompõe um sinal na frequência de seus componentes e a transformada de Karhunen-Loève (KLT) que descorrelaciona completamente um dado sinal de entrada [14].

No contexto de processamento de sinais e imagens, a PCA é conhecida como KLT [14]. Uma característica da KLT é que essa transformada descorrelaciona completamente o sinal no domínio da transformada. De fato, a KLT é uma transformada linear ótima capaz de minimizar o erro quadrático médio na compressão dos dados e concentrar maior energia em poucos coeficientes do sinal de saída [14]. No entanto, o uso da KLT em processamento de sinais é limitado, pois sua construção depende da matriz de variância e covariância dos dados de entrada, dificultando o desenvolvimento de algoritmos rápidos que a implemente. Neste contexto, surge a transformada discreta do cosseno (DCT) que é uma aproximação assintótica da KLT quando a correlação do sinal de entrada tende à unidade [15] e quando o sinal de entrada é um processo Markoviano de primeira ordem. A definição da DCT independe do sinal de entrada, o que permite o desenvolvimento de algoritmos rápidos computacionalmente eficientes. Assim, a DCT é amplamente adotada em padrões de compressão de imagem e vídeo como JPEG [16], MPEG [17] e HEVC [18], por exemplo.

A DCT de comprimento oito é muito utilizada no contexto de compressão de imagem e vídeo [14]. Por esse motivo, muitos algoritmos rápidos considerando este comprimento foram propostos [19–26]. Em particular, Loeffler *et al.* [25] propuseram um

algoritmo que requer apenas 29 adições e 11 multiplicações, atingindo a complexidade multiplicativa mínima prevista em teoria [27, p. 116]. Entretanto, mesmo com um custo computacional reduzido, o uso da DCT ainda pode ser proibitivo em contextos que disponham de baixo poder de processamento ou severa restrição de autonomia energética [28,29]. Isto se dá pelo fato de que a DCT, em sua forma exata, requer multiplicações por irracionais que são usualmente aproximados por arredondamento ou truncamento em forma racional e implementados em aritmética de ponto flutuante ou aritmética de ponto fixo com inteiros grandes [30]. Implementações em *hardware* para realizar operações aritméticas em ponto flutuante – especialmente multiplicações – são usualmente demandantes em termos de circuitaria e de consumo [14]. Neste sentido, diversas transformadas que são aproximações para a DCT vêm sendo propostas [28, 29, 31–33], de forma que sejam livres de multiplicação. Algumas aproximações para a DCT são a clássica DCT sinalizada (SDCT) [31], a DCT arredondada (RDCT) [32] e as aproximações para a DCT baseadas em funções inteiras [28].

Proposta por Haweel [31], a SDCT é construída através da aplicação da função sinal aos elementos da matriz da DCT, se tornando livre de multiplicações não triviais. Dessa forma, a matriz da SDCT tem apenas elementos no conjunto $\{\pm 1\}$ e sua transformada direta requer apenas 24 adições. A SDCT mantém a capacidade de codificação semelhante à DCT, mas oferece um custo computacional bastante reduzido. De forma semelhante, a RDCT, introduzida em [32], é obtida após a aplicação de uma função de arredondamento à matriz da DCT exata. Esta transformada aproximada é ortogonal e também possui apenas elementos no conjunto $\{0, \pm 1\}$. Seu cômputo requer apenas 22 adições, e por ser ortogonal, sua transformada inversa também é de baixa complexidade. Baseada na RDCT, a RDCT modificada (MRDCT) foi introduzida em [34] como uma nova aproximação de baixa complexidade para a DCT de comprimento 8. Ela necessita de apenas de 14 adições para seu cômputo. Em [28], é proposta uma coleção de transformadas aproximadas para a DCT baseada em funções inteiras. Em [28], são apresentadas 12 novas aproximadas de baixa complexidade, em que seus elementos variam no conjunto $\{0, \pm 1, \pm 2, \pm 3\}$. Os autores decidiram considerar os elementos ± 3 pelo fato de que multiplicações por ± 3 podem ser implementadas por apenas uma adição e uma operação de deslocamento de *bit* ($3 \cdot x = 2 \cdot x + x$). Outras aproximações alternativas para a DCT serão discutidas e apresentadas na Seção 2.3 de modo a objetivar a metodologia utilizada para encontrarmos novas transformadas aproximadas para a KLT.

Apesar do fato da KLT ser considerada uma transformada discreta ótima sob o critério de decorrelação de dados – ao nosso melhor conhecimento – pouco pode ser encontrado na literatura sobre aproximadas para a KLT. Alguns motivos que justificam esta falta de trabalhos envolvendo a KLT são citados em [35]:

- (i) a DCT é uma aproximação assintótica (quando $\rho \rightarrow 1$) para a KLT (quando o sinal é um processo Markov-1);
- (ii) o cômputo da KLT é custoso do ponto de vista aritmético;
- (iii) o custo computacional da implementação da KLT é alto.

O alto custo computacional da implementação da KLT continua sendo um problema a ser resolvido em alguns contextos. Em particular, aproximações para a KLT podem ser úteis nos contextos em que há uma quantidade massiva de dados a ser processada ou no caso de implementação em *hardware* de baixo consumo. A Figura 1 contextualiza essa assertiva. Por exemplo, a menos que dispomos de um poder computacional alto e baixa quantidade de dados – onde não há a necessidade da utilização de aproximações, em todos os demais casos é conveniente utilizarmos transformadas aproximadas.

		Poder computacional	
		Baixo	Alto
Quantidade de dados	Baixo	Potencialmente aplicável	Não aplicável
	Alto	Aplicável	Potencialmente aplicável

Figura 1 – Contextos para aplicação de transformadas aproximadas.

1.3 OBJETIVOS

Devido à lacuna na literatura no que tange estudos envolvendo alternativas de baixo custo para a KLT, e à condição de que os dados sejam altamente correlacionados, o que nem sempre é verificado em aplicações reais, o objetivo geral deste trabalho é:

Projetar/desenvolver transformadas de baixa complexidade que sejam aproximações para a KLT, considerando diferentes valores para o coeficiente de correlação (ρ).

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

1. Realizar uma revisão de literatura do estado da arte sobre a KLT, apresentando definições e a relação da DCT com a KLT;
2. Apresentar uma revisão sobre a DCT e as diferentes metodologias para encontrar transformadas aproximadas bem como seus algoritmos rápidos;
3. Propor novas transformadas aproximadas para a KLT baseadas na metodologia já aplicada em aproximadas da DCT;
4. Propor algoritmos rápidos para as transformadas propostas obtidas;
5. Avaliar matematicamente as transformadas aproximadas propostas quando comparadas à KLT exata;
6. Realizar simulação em compressão de imagens para avaliar o desempenho das novas transformadas propostas.

1.4 ESTRUTURA DO DOCUMENTO

O presente trabalho está estruturado da seguinte forma. No Capítulo 2, é apresentada a formalização teórica da KLT. Na Seção 2.2, é introduzida a DCT como uma aproximação assintótica para a KLT quando o sinal de entrada é um processo markoviano de primeira ordem. Na Seção 2.3, diversas aproximações para a DCT de comprimento $N = 8$ são apresentadas. Ainda no Capítulo 2, são explicitadas as medidas de avaliação de transformadas que serão utilizadas como figuras de mérito.

No Capítulo 3, são descritas as metodologias e as representações numéricas empregadas para implementar as aproximações para a KLT. É posto o problema de otimização sobre um conjunto de medidas de avaliação com o objetivo de identificar soluções eficientes e novas transformadas ótimas dentro da classe de transformadas propostas. A Seção 3.5 apresenta as transformadas aproximadas ótimas propostas.

O Capítulo 4 introduz os algoritmos rápidos propostos para as transformadas aproximadas. Na Seção 4.2, é discutida a complexidade computacional e é apresentada a complexidade aritmética das novas transformadas.

No Capítulo 5, é explicitada a metodologia utilizada para compressão de imagens bem como métricas utilizadas para avaliação de imagens comprimidas. Também são exibidas as imagens comprimidas pelas transformadas propostas e gráficos e tabelas apresentando as medidas de qualidade de imagem das mesmas.

Por fim, no Capítulo 6, são apresentadas as conclusões finais do presente trabalho. O Apêndice A apresenta as transformadas de comprimento $N = 8$ obtidas na Seção 3.5 e no Apêndice B são exibidas as imagens utilizadas no experimento de compressão de imagens.

2 KLT E DCT

No presente capítulo, é apresentado o enquadramento teórico da KLT, bem como sua formalização matemática quando sinais de entrada markovianos de primeira ordem são considerados. Como o cômputo da KLT pode ser muito custoso em contextos em que se exige baixo consumo, baixa complexidade de *hardware*, elevada autonomia, ou possivelmente, em casos em que se haja uma quantidade absurdamente grande de dados a serem processados, seu uso pode tornar-se proibitivo. Nesse sentido, por ser uma aproximação assintótica para a KLT quando o coeficiente de correlação do sinal tende à unidade [15, 36] e por herdar suas boas propriedades, a DCT também é evidenciada neste capítulo. Aproximações clássicas para a DCT que foram propostas com o intuito de reduzir o custo computacional de sua implementação também são apresentadas. As aproximações para a KLT propostas nesta dissertação serão baseadas na metodologia utilizada para encontrar estas aproximações da DCT.

2.1 A TRANSFORMADA DE KARHUNEN-LOÈVE

A KLT [37–39], desenvolvida de forma independente pelos probabilistas Kari Karhunen (1915 - 1992) [40] e Michel Loève (1907 - 1979) [41], é derivada através da busca de uma transformada que reduza a dimensionalidade e descorrelacione completamente os dados. Na literatura, podemos encontrar abordagens equivalentes da KLT, como análise de componentes principais [42] e transformada de Hotelling [43].

Uma dificuldade que podemos encontrar em problemas físicos é que os dados são extremamente correlacionados e contém informação redundante. Em particular, os *pixels* de imagens naturais são altamente correlacionados entre si [6]. Nesse sentido, vê-se a importância de utilizarmos uma transformada que seja capaz de reduzir a correlação e a dimensão dos dados. A KLT possui as seguintes propriedades [35, 44]: (i) descorrelaciona completamente os componentes do sinal; (ii) minimiza o erro quadrático médio na compressão dos dados; (iii) concentra maior energia em poucos coeficientes do sinal de saída e (iv) minimiza a representação total de entropia da sequência. Ela é a representação de um sinal aleatório, na qual as funções de base ortogonais são obtidas pelos autovetores da matriz de autocorrelação correspondente, ou seja, seu cálculo depende do sinal de entrada a ser transformado.

Considere um sinal aleatório N -dimensional de entrada $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_{N-1}]^T$

com matriz de covariância associada dada por [45]:

$$\mathbf{R}_x = E\{(\mathbf{x} - E\{\mathbf{x}\})(\mathbf{x} - E\{\mathbf{x}\})^\top\}. \quad (2.1)$$

Buscamos transformar este vetor, por uma transformação linear \mathbf{K} , em um vetor não correlacionado $\mathbf{y} = [y_0 \ y_1 \ \dots \ y_{N-1}]^\top$. Em outras palavras, queremos escrever \mathbf{y} :

$$\mathbf{y} = \mathbf{K} \cdot \mathbf{x}, \quad (2.2)$$

de modo que a matriz de correlação de \mathbf{y} seja estritamente diagonal. Ou seja:

$$E\{(\mathbf{y} - E\{\mathbf{y}\})(\mathbf{y} - E\{\mathbf{y}\})^\top\} = \begin{bmatrix} \lambda_0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_{N-1} \end{bmatrix}, \quad (2.3)$$

em que cada λ_i representa um autovalor da matriz \mathbf{R}_x .

2.1.1 KLT para sinais Markovianos de primeira ordem

Não há forma fechada para o cômputo da KLT de um sinal arbitrário qualquer, mas levando em conta que muitos sinais reais têm comportamento estatístico do tipo Markov-1 [14], Ray e Driver [46] derivaram uma solução analítica para a KLT considerando este tipo de sinal.

Assim, assumindo que \mathbf{x} é um sinal Markoviano de primeira ordem, podemos reescrever sua matriz de covariância por [14]:

$$\mathbf{R}_x = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{N-1} \\ \rho & 1 & \rho & \dots & \rho^{N-2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \dots & \rho & 1 \end{bmatrix}, \quad (2.4)$$

em que ρ é o coeficiente de correlação de Pearson, $0 < \rho < 1$. Os (i, j) -ésimos elementos da matriz de transformação $\mathbf{K}^{(\rho)}$, dado um valor arbitrário de ρ , são dados por:

$$u_{ij} = \sqrt{\frac{2}{N + \lambda_i}} \sin \left[\omega_i \left(i - \frac{N-1}{2} \right) + \frac{(j+1)\pi}{2} \right], \quad (2.5)$$

em que os autovalores da matriz de covariância do sinal transformado \mathbf{y} são:

$$\lambda_i = \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos \omega_i}, \quad (2.6)$$

e $\omega_1, \omega_2, \dots, \omega_N$ são as N soluções da equação:

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

Como a matriz da KLT para um sinal Markoviano depende de ρ , identificaremos essa matriz por $\mathbf{K}^{(\rho)}$. Por exemplo, a matriz da KLT de comprimento oito, para $\rho = 0.95$, é dada por:

$$\mathbf{K}^{(0.95)} = \begin{bmatrix} 0.338 & 0.351 & 0.360 & 0.364 & 0.364 & 0.360 & 0.351 & 0.338 \\ 0.481 & 0.420 & 0.286 & 0.101 & -0.101 & -0.286 & -0.420 & -0.481 \\ 0.467 & 0.207 & -0.179 & -0.456 & -0.456 & -0.179 & 0.207 & 0.467 \\ 0.423 & -0.085 & -0.487 & -0.278 & 0.278 & 0.487 & 0.085 & -0.423 \\ 0.360 & -0.347 & -0.356 & 0.351 & 0.351 & -0.356 & -0.347 & 0.360 \\ 0.283 & -0.488 & 0.094 & 0.415 & -0.415 & -0.094 & 0.488 & -0.283 \\ 0.195 & -0.462 & 0.460 & -0.190 & -0.190 & 0.460 & -0.462 & 0.195 \\ 0.100 & -0.279 & 0.416 & -0.490 & 0.490 & -0.416 & 0.279 & -0.100 \end{bmatrix}, \quad (2.8)$$

em que os elementos foram arredondados em três casas decimais.

No contexto de processamento de imagens, os dados dos *pixels* adjacentes são altamente correlacionados [47]. A suposição de que uma imagem digital seja uma realização de um processo Markoviano de primeira ordem altamente correlacionado, usualmente suposto $\rho = 0.95$, é amplamente aceita [14]. Nessa classe de sinais, a KLT é a transformação linear ótima para descorrelação dos dados [14].

2.2 DCT COMO APROXIMAÇÃO ASSINTÓTICA PARA A KLT

Como a obtenção da KLT depende da matriz de variância e covariância do sinal de entrada, dificultando o desenvolvimento de algoritmos rápidos que a implemente [45, 48, 49], seu uso em aplicações em processamento de sinais tais como compressão de imagem, transmissão e processamento de vídeo em tempo real, sistemas de comunicação, entre outros, pode ser proibitivo. Em 1981, Clarke [36] indicou em seu artigo que a DCT se aproxima da KLT quando o coeficiente de correlação ρ entre os elementos dos dados adjacentes se aproximam de 1.

Assumindo que as componentes do sinal de entrada são altamente correlacionados, temos que $\rho \approx 1$. Dessa maneira, podemos reescrever a Equação (2.7) como:

$$\tan N\omega = \lim_{\rho \rightarrow 1} \frac{-(1 - \rho^2) \sin \omega}{(1 + \rho^2) \cos \omega - 2\rho} = 0. \quad (2.9)$$

As N raízes positivas da Equação (2.9) são dadas por:

$$\omega_i = \frac{i\pi}{N}, \quad i = 0, 1, \dots, N-1. \quad (2.10)$$

Substituindo (2.10) em (2.6), temos como resultado $\lambda_i = 0$ para $i \neq 0$. Além disso, temos que o traço da matriz \mathbf{R}_x , definido por:

$$\text{tr}(\mathbf{R}_x) = \sum_{n=0}^{N-1} [\mathbf{R}_x]_{n,n}, \quad (2.11)$$

é igual à soma dos N autovalores [50]. Assim,

$$\lim_{\rho \rightarrow 1} \text{tr}(\mathbf{R}_x) = \lim_{\rho \rightarrow 1} \sum_{n=0}^{N-1} [\mathbf{R}_x]_{n,n} = \lim_{\rho \rightarrow 1} \sum_{i=0}^{N-1} \lambda_i = \lambda_0. \quad (2.12)$$

Dado que $[\mathbf{R}_x]_{n,n} = 1$, para $n = 0, 1, \dots, N-1$, então $\lambda_0 = N$ quando $\rho \rightarrow 1$. Substituindo em (2.5) temos:

$$u_{0,j} = \frac{1}{\sqrt{N}}, \quad (2.13)$$

$$u_{i,j} = \sqrt{\frac{2}{N}} \sin \left\{ \frac{i(2j+1)\pi}{2N} + \frac{\pi}{2} \right\} = \sqrt{\frac{2}{N}} \cos \left\{ \frac{i(2j+1)\pi}{2N} \right\}, \quad i \neq 0. \quad (2.14)$$

Definindo a constante c_i como:

$$c_i = \begin{cases} \frac{1}{\sqrt{2}} & \text{se } i = 0, \\ 1, & \text{se } i \neq 0, \end{cases} \quad (2.15)$$

temos que os (i, j) -ésimos elementos da matriz da DCT de comprimento N são dados por [14]:

$$u_{i,j} = \sqrt{\frac{2}{N}} c_i \cos \left\{ \frac{i(2j+1)\pi}{2N} \right\}, \quad i, j = 0, 1, \dots, N-1. \quad (2.16)$$

Especialmente, no caso em que $N = 8$, a matriz da DCT é dada por:

$$\mathbf{C}_8 = \frac{1}{2} \begin{bmatrix} \gamma_3 & \gamma_3 \\ \gamma_0 & \gamma_2 & \gamma_4 & \gamma_6 & -\gamma_6 & -\gamma_4 & -\gamma_2 & -\gamma_0 \\ \gamma_1 & \gamma_5 & -\gamma_5 & -\gamma_1 & -\gamma_1 & -\gamma_5 & \gamma_5 & \gamma_1 \\ \gamma_2 & -\gamma_6 & -\gamma_0 & -\gamma_4 & \gamma_4 & \gamma_0 & \gamma_6 & -\gamma_2 \\ \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 & \gamma_3 & -\gamma_3 & -\gamma_3 & \gamma_3 \\ \gamma_4 & -\gamma_0 & \gamma_6 & \gamma_2 & -\gamma_2 & -\gamma_6 & \gamma_0 & -\gamma_4 \\ \gamma_5 & -\gamma_1 & \gamma_1 & -\gamma_5 & -\gamma_5 & \gamma_1 & -\gamma_1 & \gamma_5 \\ \gamma_6 & -\gamma_4 & \gamma_2 & -\gamma_0 & \gamma_0 & -\gamma_2 & \gamma_4 & -\gamma_6 \end{bmatrix}, \quad (2.17)$$

em que

$$\gamma_0 = \frac{\sqrt{2 + \sqrt{2 + \sqrt{2}}}}{2}, \quad \gamma_1 = \frac{\sqrt{2}}{2}, \quad (2.18)$$

$$\gamma_2 = \frac{\sqrt{2 + \sqrt{2 - \sqrt{2}}}}{2}, \quad \gamma_3 = \frac{\sqrt{2 + \sqrt{2}}}{2}, \quad (2.19)$$

$$\gamma_4 = \frac{\sqrt{2 - \sqrt{2 - \sqrt{2}}}}{2}, \quad \gamma_5 = \frac{\sqrt{2 - \sqrt{2}}}{2}, \quad (2.20)$$

$$\gamma_6 = \frac{\sqrt{2 - \sqrt{2 + \sqrt{2}}}}{2}. \quad (2.21)$$

Arredondando em três casas decimais, temos:

$$\mathbf{C}_8 \approx \begin{bmatrix} 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 & 0.354 \\ 0.490 & 0.416 & 0.278 & 0.098 & -0.098 & -0.278 & -0.416 & -0.490 \\ 0.462 & 0.191 & -0.191 & -0.462 & -0.462 & -0.191 & 0.191 & 0.462 \\ 0.416 & -0.098 & -0.490 & -0.278 & 0.278 & 0.490 & 0.098 & -0.416 \\ 0.354 & -0.354 & -0.354 & 0.354 & 0.354 & -0.354 & -0.354 & 0.354 \\ 0.278 & -0.490 & 0.098 & 0.416 & -0.416 & -0.098 & 0.490 & -0.278 \\ 0.191 & -0.462 & 0.462 & -0.191 & -0.191 & 0.462 & -0.462 & 0.191 \\ 0.098 & -0.278 & 0.416 & -0.490 & 0.490 & -0.416 & 0.278 & -0.098 \end{bmatrix}. \quad (2.22)$$

A título de comparação, a Equação (2.23) apresenta a matriz da KLT de comprimento 8 para $\rho = 0.99$. Podemos notar a semelhança entre os elementos das duas matrizes, já que a DCT é um caso assintótico da KLT quando $\rho \rightarrow 1$.

$$\mathbf{K}^{(0.99)} = \begin{bmatrix} 0.350 & 0.353 & 0.355 & 0.356 & 0.356 & 0.355 & 0.353 & 0.350 \\ 0.489 & 0.417 & 0.279 & 0.098 & -0.098 & -0.279 & -0.417 & -0.489 \\ 0.463 & 0.194 & -0.189 & -0.461 & -0.461 & -0.189 & 0.194 & 0.463 \\ 0.417 & -0.095 & -0.490 & -0.278 & 0.278 & 0.490 & 0.095 & -0.417 \\ 0.355 & -0.352 & -0.354 & 0.353 & 0.353 & -0.354 & -0.352 & 0.355 \\ 0.279 & -0.490 & 0.097 & 0.416 & -0.416 & -0.097 & 0.490 & -0.279 \\ 0.192 & -0.462 & 0.462 & -0.191 & -0.191 & 0.462 & -0.462 & 0.192 \\ 0.098 & -0.278 & 0.416 & -0.490 & 0.490 & -0.416 & 0.278 & -0.098 \end{bmatrix}. \quad (2.23)$$

É importante ressaltar que a matriz da DCT é sempre a mesma independente do sinal de entrada, o que faz dela uma transformada muito útil na prática [48]. Em particular, a DCT de comprimento oito é bastante utilizada no contexto de processamento de

imagens e vídeo, como nos padrões JPEG [16], MPEG-1 [51], MPEG-2 [52], e HEVC [18]. Por esse motivo, diversos algoritmos rápidos foram propostos considerando este comprimento da DCT. Alguns desses algoritmos rápidos são apresentados no Capítulo 4.

2.3 APROXIMAÇÕES PARA A DCT

Apesar do cômputo da DCT independer do sinal de entrada e da existência de vários algoritmos rápidos que a implemente, sua complexidade computacional ainda pode ser alta pelo fato de que sua implementação requer aritmética de ponto flutuante [14], o que demanda maiores exigências de *hardware*. Nesse sentido, diversas aproximações para a DCT vêm sendo propostas [28, 29, 31, 32, 34, 53–64] obtendo-se significativas reduções de custo aritmético. Uma breve revisão dos métodos para encontrar aproximadas para a DCT será feita a seguir.

Uma maneira para aproximar a DCT é encontrar matrizes onde seus elementos de entrada sejam racionais diádicos [14, p. 221], o que torna as aproximações livres de aritmética de ponto flutuante. O conjunto de números racionais diádicos pode ser definido como:

$$\frac{p}{2^k}, \quad p, k \in \mathbb{N}, \quad p = 1, 2, \dots, 2^k - 1. \quad (2.24)$$

No contexto de implementação de circuitos aritméticos digitais, a multiplicação ou divisão de uma quantidade por potências de dois é equivalente à aplicação de um deslocamento de *bits*, requerendo um menor custo de implementação que adições e multiplicações. Dessa forma, a multiplicação de uma quantidade x por um racional diádico pode ser implementada somente usando adições e deslocamentos de *bits*. Por exemplo, a multiplicação de x pelo racional diádico $5/8$ pode ser explicitada por:

$$\frac{5}{8}x = \frac{1}{2}x + \frac{1}{8}x, \quad (2.25)$$

o que requer uma adição e quatro deslocamentos. Embora a concepção de matrizes aproximadas para a DCT utilizando racionais diádicos ofereça boas aproximações, como as baseadas em *lifting scheme* [65], esta metodologia implica em um incremento na complexidade aditiva e no número de deslocamentos de *bits*.

Uma boa alternativa a utilizar racionais diádicos, que é mais efetiva do ponto de vista computacional, é empregar quantidades que não introduzam adições extras na implementação, como no caso de potências de dois. A multiplicação de uma quantidade

qualquer x por 2^k , com $k \in \mathbb{Z}$, necessita somente k deslocamentos de *bits* para a sua implementação [14, p. 222]. Ainda que esta metodologia não incorpore adições ou multiplicações, pode requerer um grande número de deslocamento de *bits*.

A fim de minimizar o número de somas e deslocamentos, muitas aproximações encontradas na literatura geralmente possuem elementos definidos no conjunto $\mathcal{C}_0 = \{0, \pm 1/2, \pm 1, \pm 2\}$. Neste caso, os elementos neste conjunto não introduzem adições a mais, e apenas os elementos em $\{\pm 1/2, \pm 2\}$ demandam um deslocamento de *bit*. Em [28, 66], os autores consideram utilizar os elementos em \mathcal{C}_0 e também ± 3 , pelo fato de que multiplicações pela constante ± 3 podem ser implementadas por meio de uma adição e um deslocamento de *bits* ($3 \cdot x = 2 \cdot x + x$) [28].

Outra característica importante que é buscada em aproximações para a DCT é que as transformadas sejam ortogonais [14, 61], pois se uma matriz é ortogonal, sua inversa pode ser obtida pela transposta. Assim, se a transformada de baixa complexidade for ortogonal, sua inversa também será de baixa complexidade. A seguir, dois conceitos importantes quando tratamos de aproximações para transformadas são definidos.

Definição 1. *Uma matriz \mathbf{A} é dita ser **ortogonal** se $\mathbf{A} \cdot \mathbf{A}^\top$ resulta em uma matriz diagonal.*

Definição 2. *Se $\mathbf{A} \cdot \mathbf{A}^\top$ resulta na matriz identidade, então \mathbf{A} é dita ser **ortonormal**.*

A matriz da DCT exata é ortonormal. Dessa forma, é interessante buscar aproximações para a DCT que também sejam ortonormais. Dado \mathbf{T} , uma transformada ortogonal de baixa complexidade, é possível derivar uma matriz $\hat{\mathbf{T}}$ ortonormal baseada em \mathbf{T} através de decomposição polar [67, 68]. A partir desta metodologia, a ortonormalização da matriz \mathbf{T} pode ser dada apenas calculando a matriz diagonal \mathbf{S} :

$$\mathbf{S} = \sqrt{(\mathbf{T} \cdot \mathbf{T}^\top)^{-1}}, \quad (2.26)$$

em que $\sqrt{\cdot}$ representa a raiz quadrada da matriz. Assim, uma aproximação ortonormal para a DCT pode ser dada por:

$$\hat{\mathbf{T}} = \mathbf{S} \cdot \mathbf{T}, \quad (2.27)$$

e, neste caso, $\hat{\mathbf{T}}$ é ortonormal e satisfaz $\hat{\mathbf{T}}^{-1} = \hat{\mathbf{T}}^\top$. Seguindo esta formalização, como \mathbf{S} é uma matriz diagonal, a complexidade computacional de $\hat{\mathbf{T}}$ é a mesma de \mathbf{T} , exceto pelos

multiplicandos contidos na matriz diagonal \mathbf{S} . Em algumas aplicações, como no caso de compressão de imagens, a matriz \mathbf{S} pode ser incorporada na etapa de quantização [32, 34, 56, 61, 69] e, dessa forma, não há incrementação na complexidade computacional do processamento.

As Equações (2.26) e (2.27) só são satisfeitas se \mathbf{T} for uma matriz ortogonal. Se essa característica de \mathbf{T} não é satisfeita, \mathbf{S} não é uma matriz diagonal. Nesse caso, os elementos fora da diagonal de $\mathbf{T} \cdot \mathbf{T}^\top$ resultam em um aumento no custo computacional de $\hat{\mathbf{T}}$. Porém, é mostrado em [28] que, sob certas condições, é possível aproximar a matriz diagonal \mathbf{S} substituindo os elementos fora da diagonal por zero. Dessa maneira, a matriz resultante $\hat{\mathbf{S}}$ é dada por:

$$\hat{\mathbf{S}} = \sqrt{[\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)]^{-1}}, \quad (2.28)$$

em que $\text{diag}(\cdot)$ retorna a matriz diagonal. Assim, para o caso em que \mathbf{T} é não-ortogonal, a aproximação $\hat{\mathbf{T}}$ pode ser definida por:

$$\hat{\mathbf{T}} = \hat{\mathbf{S}} \cdot \mathbf{T}. \quad (2.29)$$

Esta aproximação pode ser muito relevante se $\hat{\mathbf{S}}$ é próximo de \mathbf{S} ou se \mathbf{T} é quase-ortogonal. Uma métrica que pode ser utilizada para medir o quão próxima uma matriz está de sua diagonal é o **desvio de diagonalidade**. Essa medida é definida por [70]:

$$\delta(\mathbf{T}) = 1 - \frac{\|\text{diag}(\mathbf{T})\|_{\text{F}}}{\|\mathbf{T}\|_{\text{F}}}. \quad (2.30)$$

em que $\|\cdot\|_{\text{F}}$ é a norma de Frobenius para matrizes [68]. No presente trabalho, utilizaremos como critério, assim como em [28], o desvio de diagonalidade da matriz da SDCT [31] como o valor máximo permitido para matrizes não ortogonais. O desvio de diagonalidade da SDCT é igual a $1 - 2/\sqrt{5}$. Dessa forma, a condição para o desvio de diagonalidade será:

$$\delta(\mathbf{T} \cdot \mathbf{T}^\top) \leq 1 - \frac{2}{\sqrt{5}} \approx 0.1056. \quad (2.31)$$

Definição 3. *Se \mathbf{T} satisfizer a Equação (2.31), então \mathbf{T} é dita ser **quase-ortogonal**.*

No contexto de processamento de sinais e imagens, diversas aproximações para a DCT foram propostas seguindo as metodologias citadas acima. Algumas transformadas podem ser citadas, como a DCT sinalizada (SDCT) [31], a série de aproximações BAS [29,

A SDCT não é uma transformada ortogonal, porém como sua matriz tem entradas somente em $\{\pm 1\}$, requer apenas adições para sua implementação. Haweel [31] avaliou a SDCT em termos de identificação de sistemas e compressão de imagens e mostrou que esta transformada tem um desempenho similar ao da DCT no quesito decorrelação e compactação de energia.

2.3.2 DCT Arredondada (RDCT)

Proposta em [32], a RDCT é uma matriz ortogonal que é obtida quando aplicamos a função de arredondamento (*round-off*) aos elementos da matriz DCT. A RDCT é uma transformada que demanda um baixo custo aritmético e pode ser definida como:

$$\mathbf{T}_{\text{RDCT}} = \text{round}(2 \cdot \mathbf{C}), \quad (2.36)$$

em que \mathbf{C} é a matriz da DCT e a função $\text{round}(\cdot)$ é dada por:

$$\text{round}(x) = \text{sign}(x) \cdot \lfloor |x| + 0.5 \rfloor, \quad (2.37)$$

em que $\lfloor x \rfloor = \max\{m \in \mathbb{Z} | m \leq x\}$. A função de arredondamento é aplicada elemento a elemento na matriz. A transformada RDCT para $N = 8$ é apresentada a seguir:

$$\mathbf{T}_{\text{RDCT}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}, \quad (2.38)$$

com

$$\widehat{\mathbf{S}}_{\text{RDCT}} = \text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}} \right). \quad (2.39)$$

Como suas entradas estão definidas em $\{0, \pm 1\}$, esta transformada requer apenas adições para sua implementação. Em [32], os autores demonstram que o desempenho da RDCT

também é muito próximo ao da DCT em termos de compressão de imagens, superando a SDCT.

2.3.3 DCT arredondada modificada (MRDCT)

Em [34], é proposta uma aproximação baseada na RDCT de comprimento $N = 8$ a qual possui a menor complexidade de todas as aproximações para a DCT consideradas neste trabalho, necessitando apenas de 14 adições para o seu cômputo. A MRDCT é dada por:

$$\mathbf{T}_{\text{MRDCT}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad (2.40)$$

com

$$\hat{\mathbf{S}}_{\text{MRDCT}} = \text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{2}}, \frac{1}{2}, \frac{1}{\sqrt{2}} \right). \quad (2.41)$$

Esta transformada foi obtida trocando alguns elementos da matriz da RDCT por zero. Por ter um grande número de entradas com valor zero, esta transformada tem um baixo custo computacional associado.

2.3.4 Aproximações da DCT baseadas em funções inteiras

Em [28], introduz-se uma série de aproximações de baixa complexidade para a DCT baseadas em funções inteiras de arredondamento [72]. O objetivo foi mapear as entradas reais da DCT em quantidades inteiras. O mapeamento utilizado para tal finalidade foi:

$$\mathbb{R} \longrightarrow \mathcal{M}_8(\mathbb{Z}) \quad (2.42)$$

$$\alpha \longmapsto \text{int}(\alpha \cdot \mathbf{C}_8), \quad (2.43)$$

em que $\mathcal{M}_8(\mathbb{Z})$ é o espaço de matrizes 8×8 sob o conjuntos dos inteiros \mathbb{Z} , $\text{int}(\cdot)$ é uma função inteira a ser definida e α é um parâmetro definido como um fator de expansão da

DCT que nos permite uma ampla gama de mapeamentos de inteiros [14, p. 273] e \mathbf{C}_8 é a matriz exata da DCT de comprimento $N = 8$. Assim, a formalização matemática para derivar as transformadas aproximadas para a DCT é dada por:

$$\mathbf{T} = \text{int}(\alpha \cdot \mathbf{C}_8). \quad (2.44)$$

Os exemplos das funções inteiras que foram utilizadas neste contexto são:

$$\begin{aligned} \text{floor}(x) &= \lfloor x \rfloor = \max\{m \in \mathbb{Z} | m \leq x\}, \\ \text{ceil}(x) &= \lceil x \rceil = \min\{n \in \mathbb{Z} | n \geq x\}, \\ \text{trunc}(x) &= \text{sign}(x) \cdot \lfloor |x| \rfloor, \\ \text{round}_{AFZ}(x) &= \text{sign}(x) \cdot \lceil |x| \rceil, \end{aligned} \quad (2.45)$$

em que $|\cdot|$ retorna o valor absoluto de seu argumento. O termo AFZ na função $\text{round}_{AFZ}(x)$ significa *away-from-zero*.

Os autores também utilizaram as funções inteiras de arredondamento para o inteiro mais próximo [73] para derivar novas aproximações para a DCT. Estas funções são definidas por:

$$\begin{aligned} \text{round}_{HU}(x) &= \left\lfloor x + \frac{1}{2} \right\rfloor, \\ \text{round}_{HD}(x) &= \left\lceil x - \frac{1}{2} \right\rceil, \\ \text{round}_{HAFZ}(x) &= \text{sign}(x) \cdot \left\lfloor |x| + \frac{1}{2} \right\rfloor, \\ \text{round}_{HTZ}(x) &= \text{sign}(x) \cdot \left\lceil |x| - \frac{1}{2} \right\rceil, \\ \text{round}_{EVEN}(x) &= \begin{cases} \lfloor x - \frac{1}{2} \rfloor, & \text{se } \frac{2x-1}{4} \in \mathbb{Z}, \\ \lfloor x + \frac{1}{2} \rfloor, & \text{caso contrário,} \end{cases} \\ \text{round}_{ODD}(x) &= \begin{cases} \lfloor x + \frac{1}{2} \rfloor, & \text{se } \frac{2x-1}{4} \in \mathbb{Z}, \\ \lfloor x - \frac{1}{2} \rfloor, & \text{caso contrário,} \end{cases} \end{aligned} \quad (2.46)$$

em que os termos HU, HD, HAFZ e HTZ significam, respectivamente, *half-up*, *half-down*, *half-away-from-zero* e *half-towards-zero*. A função round_{HU} é a mesma função que foi utilizada para derivar a RDCT.

A busca computacional para encontrar as matrizes ótimas de baixa complexidade para um valor fixo de α foi feita da seguinte maneira:

- i) os elementos da matriz \mathbf{T} devem estar contidos em $\mathcal{C} = \{0, \pm 1, \pm 2, \pm 3\}$;
- ii) $\mathbf{T} \cdot \mathbf{T}^\top$ deve ser uma matriz diagonal ou possuir um pequeno desvio de diagonalidade (2.30);
- iii) se \mathbf{T} é não-ortogonal, mas $\mathbf{T} \cdot \mathbf{T}^\top$ é aproximadamente uma matriz diagonal, então a matriz inversa \mathbf{T}^{-1} deve ser de baixa complexidade com seus elementos definidos em \mathcal{C} .

Para que os elementos de $\mathbf{T} = \text{int}(\alpha \cdot \mathbf{C}_8)$ sejam definidos em \mathcal{C} , convém limitar a faixa de possíveis valores de α . Sendo γ o maior elemento de \mathbf{C}_8 , é suficiente resolver a seguinte inequação para α : $0 \leq \text{int}(\alpha \cdot \gamma) \leq 3$. Para as funções inteiras floor, ceil, trunc e round_{AFZ} e todas as funções de arredondamento para o inteiro mais próximo (2.46) foram obtidos os seguintes intervalos de α , respectivamente: $[0, 3/\gamma]$, $[1/\gamma, 4/\gamma]$, $[1/\gamma, 4/\gamma]$, $[0, 3/\gamma]$ e $[0, 3/\gamma]$.

A partir destas restrições, são apresentadas em [28] as matrizes 8×8 de baixa complexidade, evidenciadas na Tabela 1. As aproximadas propostas foram avaliadas em termos de compressão de imagens e apresentaram desempenho semelhante ao da DCT, com um custo computacional bastante reduzido.

2.4 MEDIDAS DE AVALIAÇÃO DE DESEMPENHO

Para a avaliação de desempenho de transformadas aproximadas, usualmente são consideradas algumas medidas como figuras de mérito. Essas medidas são classificadas basicamente em dois tipos: (i) medidas de codificação, como ganho de codificação [74] e eficiência da transformada [75], que medem o poder de decorrelação e compactação de energia; e (ii) medidas de proximidade com a transformada exata, como erro quadrático médio [14] e erro de energia total [32], que medem semelhanças ou dissemelhanças entre as matrizes aproximadas e a transformada exata em um sentido de distância euclidiana. Essas medidas vêm sendo usadas para avaliação de aproximações à diferentes transformadas. As medidas são apresentadas a seguir. Denotaremos uma matriz de transformação exata por \mathbf{K} e uma matriz de aproximação de \mathbf{K} é denotada por \mathbf{T} .

2.4.1 Ganho de codificação

O ganho de codificação unificado mede a capacidade de compactação de energia da transformada e é dado por [74]:

$$\text{Cg}(\mathbf{T}) = 10 \cdot \log_{10} \left\{ \prod_{k=1}^N \frac{1}{\sqrt{A_k \cdot B_k}} \right\}, \quad (2.47)$$

em que $A_k = \text{su} \{ (\mathbf{h}_k^\top \cdot \mathbf{h}_k) \odot \mathbf{R}_x \}$, \mathbf{h}_k é o k -ésimo vetor linha de \mathbf{T} , a função $\text{su}(\cdot)$ retorna a soma dos elementos de seu argumento matricial, \odot é o produto matricial de Hadamard [68], \mathbf{R}_x é a matriz de covariância especificada em (2.4), $B_k = \|\mathbf{g}_k\|^2$ e \mathbf{g}_k é o k -ésimo vetor linha de \mathbf{T}^{-1} .

2.4.2 Eficiência da transformada

A eficiência da transformada é definida por [75]:

$$\eta(\mathbf{T}) = 100 \frac{\sum_{i=1}^N |r_{i,i}|}{\sum_{i=1}^N \sum_{j=1}^N |r_{i,j}|}, \quad (2.48)$$

em que $r_{i,j}$ é o (i, j) -ésimo elemento de $\mathbf{T} \cdot \mathbf{R}_x \cdot \mathbf{T}^\top$.

Estas duas medidas acima não são relativas à matriz de transformação exata, elas medem o desempenho das transformadas na compressão dos dados. Já as medidas a seguir, de proximidade, medem o quão próxima a transformada aproximada está da matriz exata.

2.4.3 Erro quadrático médio

O erro quadrático médio (MSE) é dado por [14]:

$$\text{MSE}(\mathbf{K}, \mathbf{T}) = \frac{1}{N} \cdot \text{tr} \{ (\mathbf{K} - \mathbf{T}) \cdot \mathbf{R}_x \cdot (\mathbf{K} - \mathbf{T})^\top \}, \quad (2.49)$$

em que N é o comprimento da transformada e $\text{tr}(\cdot)$ é a função traço.

2.4.4 Erro de energia total

O erro de energia total mede a similaridade espectral entre a transformada aproximada e a matriz da transformada exata, e pode ser calculado por [32]:

$$\epsilon(\mathbf{K}, \mathbf{T}) = \pi \cdot \|\mathbf{K} - \mathbf{T}\|_{\mathbb{F}}^2. \quad (2.50)$$

2.4.5 Medidas de avaliação das transformadas aproximadas para DCT

Nesta subseção, vamos quantificar as medidas de avaliação, de codificação e similaridade, para cada uma das transformadas aproximadas da DCT apresentadas na Seção 2.3. Para as medidas de similaridade, é considerada a semelhança das matrizes aproximadas para a matriz da DCT exata.

Na Tabela 2, são apresentadas as medidas de codificação e similaridade das transformadas aproximadas para a DCT de comprimento $N = 8$ que são amplamente citadas na literatura. As medidas para a matriz da KLT considerando $\rho = 0.95$ e para a DCT exatas foram incluídas para referência. Os cinco melhores resultados para cada medida estão destacados em negrito. Para essa comparação, não foram consideradas as questões de complexidade/custo aritmético. Podemos notar que nessa avaliação, as transformadas que apresentaram as melhores medidas são as aproximações para a DCT baseadas em funções inteiras [28].

Tabela 2 – Medidas de avaliação das transformadas aproximadas

Transformada	$C_g(\mathbf{T})$	$\eta(\mathbf{T})$	$\epsilon(\mathbf{C}_8, \mathbf{T})$	$MSE(\mathbf{C}_8, \mathbf{T})$
KLT	8.8462	100	0	0
DCT	8.8259	93.9912	0	0

\mathbf{T}_{SDCT} [31]	7.7870	82.5525	3.3158	0.0207
\mathbf{T}_{RDCT} [32]	8.1826	87.4297	1.7945	0.0098
\mathbf{T}_{MRDCT} [34]	7.3326	80.8969	8.6592	0.0594
$\tilde{\mathbf{T}}_1$ [28]	8.1361	86.8051	8.5953	0.0375
$\tilde{\mathbf{T}}_2$ [28]	8.1361	86.8051	1.7945	0.0099
$\tilde{\mathbf{T}}_3$ [28]	8.4966	91.0133	0.5252	0.0037
$\tilde{\mathbf{T}}_4$ [28]	7.8071	83.1483	3.3158	0.0208
$\tilde{\mathbf{T}}_5$ [28]	8.1834	87.1566	1.7945	0.0098
$\tilde{\mathbf{T}}_6$ [28]	8.4816	89.6915	0.5765	0.0040
$\tilde{\mathbf{T}}_7$ [28]	8.4816	89.6915	0.5765	0.0040
$\tilde{\mathbf{T}}_8$ [28]	8.1369	86.5359	1.7945	0.0099
$\tilde{\mathbf{T}}_9$ [28]	8.3437	88.0594	0.8695	0.0062
$\tilde{\mathbf{T}}_{10}$ [28]	8.5294	89.3724	0.5881	0.0041

3 APROXIMAÇÕES PARA A KLT

Neste capítulo, serão apresentadas as metodologias utilizadas para derivar as aproximações para a KLT. Foram derivadas aproximadas para a KLT baseadas nas metodologias utilizadas em Haweel [31], Cintra e Bayer [32] e Cintra *et al.* [28]. Como o cômputo da KLT depende do coeficiente de correlação do sinal de entrada [14], consideramos ρ variando de zero até um com passos de 10^{-1} .

A busca por novas transformadas foi feita de duas maneiras distintas: (i) considerando as funções sinal e de arredondamento aplicadas à matriz da KLT exata; (ii) e considerando as funções inteiras, aplicadas aos elementos da matriz da KLT multiplicados pelo fator de escala α .

No primeiro caso, para cada matriz da KLT (calculada em cada passo de ρ) foram aplicadas a função sinal aos elementos da matriz, e a função de arredondamento aos elementos da matriz multiplicados por dois. Considerando o segundo caso, foi necessário definir os intervalos de α e então propor uma função de otimização que retornasse apenas as transformadas aproximadas ótimas de acordo com as figuras de mérito apresentadas na Seção 2.4. Depois de encontradas as transformadas ótimas, utilizamos o algoritmo de agrupamento *k*-means [76] para reunir as transformadas em grupos segundo a métrica de ganho de codificação.

Por fim, apresentamos como resultados as transformadas ótimas propostas e uma comparação entre as medidas de codificação e similaridade das transformadas aproximadas novas *versus* transformadas aproximadas clássicas da literatura. Todas as transformadas propostas neste capítulo seguirão a seguinte formulação matemática:

$$\hat{\mathbf{K}} = \hat{\mathbf{S}} \cdot \mathbf{T}, \quad (3.1)$$

em que \mathbf{T} serão as transformadas propostas com entradas apenas de elementos inteiros e $\hat{\mathbf{S}}$ é a matriz diagonal cuja formulação é apresentada na Seção 2.3.

3.1 SKLT

Nesta seção, formalizamos matematicamente a aproximação proposta para a KLT baseada na técnica empregada no desenvolvimento da SDCT. Tal técnica considera a função sinal para gerar uma aproximação matricial da DCT. Desse modo, propomos a

seguinte matriz de transformação:

$$\mathbf{T} \triangleq \text{sign}(\mathbf{K}^{(\rho)}), \quad (3.2)$$

em que $\mathbf{K}^{(\rho)}$ é a matriz da KLT de comprimento $N = 8$ para um coeficiente de correlação ρ definido, dada pelos elementos em (2.5). Quando aplicada a uma matriz, a função sinal é empregada elemento-a-elemento. Denominamos essa aproximação de KLT sinalizada (SKLT).

Para encontrarmos as novas transformadas, realizamos a seguinte busca computacional na família de matrizes para diferentes valores de ρ :

- (i) Calculamos a KLT para valores de ρ variando de 0.1 a 0.9 com passos de 10^{-1} . Os casos de $\rho = 0$ e $\rho = 1$ não foram considerados, pois o primeiro resulta em degeneração da matriz de covariância e o segundo resulta na DCT exata, cuja teoria de aproximação é coberta em [14, 15, 77].
- (ii) Para cada matriz da KLT calculada, aplicamos a função sinal e checamos se ela era diferente ou igual a matriz anterior.

Dessa forma, considerando $N = 8$ encontramos apenas duas matrizes SKLT diferentes. A matriz \mathbf{T}_0 é definida para qualquer valor de autocorrelação no intervalo $(0, 0.6]$ e \mathbf{T}_1 é definida para $\rho \in (0.6, 1)$. As respectivas matrizes \mathbf{T}_0 e \mathbf{T}_1 são apresentadas na Tabela 3 bem como suas matrizes diagonais $\hat{\mathbf{S}}$.

Tabela 3 – Aproximações SKLT

Transformada	ρ	Matriz	$\hat{\mathbf{S}}$
\mathbf{T}_0	$(0, 0.6]$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}} \right)$
\mathbf{T}_1 [31]	$(0.6, 1)$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}} \right)$

Destaca-se que \mathbf{T}_1 é a mesma matriz da SDCT [31], mas a \mathbf{T}_0 , ao nosso melhor conhecimento, é nova na literatura. Estas transformadas têm entradas apenas no

conjunto $\{\pm 1\}$ o que faz delas transformadas de baixa complexidade. Dessa forma, as aproximações para a KLT derivadas segundo esta metodologia são dadas por $\widehat{\mathbf{K}}_0 = \widehat{\mathbf{S}} \cdot \mathbf{T}_0$ e $\widehat{\mathbf{K}}_1 = \widehat{\mathbf{S}} \cdot \mathbf{T}_1$. A Tabela 4 apresenta as medidas de codificação e similaridade com a KLT exata das transformadas aproximadas obtidas. Para comparação, as medidas da KLT para $\rho = 0.6$ ($\mathbf{K}^{(0.6)}$) e 0.95 ($\mathbf{K}^{(0.95)}$) também foram consideradas. Para as métricas ϵ e MSE, consideramos a similaridade de cada transformada proposta com a KLT calculada dado algum valor de ρ definido no intervalo da aproximada. Por exemplo, para $\widehat{\mathbf{K}}_0$ calculamos sua similaridade com a $\mathbf{K}^{(0.6)}$ e para $\widehat{\mathbf{K}}_1$ a similaridade com a $\mathbf{K}^{(0.95)}$. Podemos notar que os valores das medidas das transformadas propostas são semelhantes aos da KLT, principalmente considerando o ganho de codificação unificado.

Tabela 4 – Medidas de codificação e similaridade das SKLT

	Intervalo de ρ	$Cg(\widehat{\mathbf{K}})$	$\eta(\widehat{\mathbf{K}})$	$\epsilon(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$	$MSE(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$
$\mathbf{K}^{(0.6)}$	0.6	1.6959	100	0	0
$\widehat{\mathbf{K}}_0$	(0, 0.6]	1.2884	61.4431	3.5782	0.1144
$\mathbf{K}^{(0.95)}$	0.95	8.8462	100	0	0
$\widehat{\mathbf{K}}_1$	(0.6, 1)	7.7870	82.5525	3.3065	0.0215

3.2 RKLT

Baseada na mesma metodologia aplicada para derivar a RDCT [32], derivamos a KLT arredondada (RKLT). A seguir, apresentaremos a formulação matemática utilizada para encontrar estas novas transformadas. A matriz de transformação da RKLT é definida por:

$$\mathbf{T} \triangleq \text{round}(2 \cdot \mathbf{K}^{(\rho)}), \quad (3.3)$$

em que a função $\text{round}(\cdot)$ é a mesma definida na Equação (2.37) e também coincide com a função round_{HAFZ} apresentada na Equação (2.46), $\mathbf{K}^{(\rho)}$ é a matriz da KLT de comprimento $N = 8$ para um coeficiente de correlação ρ definido. A busca computacional das novas transformadas aproximadas RKLT foi feita da seguinte maneira:

- (i) Calculamos a KLT para valores de ρ no intervalo de 0.1 a 0.9 com passos de 10^{-1} . Como no caso da SKLT, os valores de $\rho = 0$ e $\rho = 1$ não foram considerados.
- (ii) Para cada matriz da KLT calculada, aplicamos a função de arredondamento e checamos se ela era diferente ou igual a matriz anterior.

Obtivemos quatro aproximações diferentes para a KLT. As transformadas obtidas, bem como os intervalos de ρ e a matriz diagonal $\widehat{\mathbf{S}}$, são mostradas na Tabela 5.

Tabela 5 – Aproximações RKLT

Transformada	ρ	Matriz	$\widehat{\mathbf{S}}$
\mathbf{T}_2	$(0, 0.4)$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & 0 & -1 & 1 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}} \right)$
\mathbf{T}_3	$[0.4, 0.7)$	$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}} \right)$
\mathbf{T}_4	$[0.7, 0.8)$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}} \right)$
\mathbf{T}_5 [30]	$[0.8, 1)$	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$	$\text{diag} \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}} \right)$

A transformada \mathbf{T}_5 obtida utilizando esta metodologia é a mesma matriz encontrada em [30]. A Tabela 6 apresenta as medidas de avaliação das transformadas aproximadas $\widehat{\mathbf{K}}$ obtidas. As medidas das matrizes da KLT consideradas nas métricas de similaridade também estão exibidas na tabela. Podemos notar que, considerando os valores de ρ , as transformadas propostas apresentam bons resultados principalmente considerando as medidas $\text{Cg}(\widehat{\mathbf{K}})$ e $\text{MSE}(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$.

Tabela 6 – Medidas de codificação e similaridade das RKLT

	Intervalo de ρ	$Cg(\widehat{\mathbf{K}})$	$\eta(\widehat{\mathbf{K}})$	$\epsilon(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$	$MSE(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$
$\mathbf{K}^{(0.3)}$	0.3	0.3584	100	0	0
$\widehat{\mathbf{K}}_2$	(0, 0.4)	0.2829	80.70876	1.6751	0.0659
$\mathbf{K}^{(0.4)}$	0.4	0.6626	100	0	0
$\widehat{\mathbf{K}}_3$	[0.4, 0.7)	0.5616	70.2996	1.7011	0.0660
$\mathbf{K}^{(0.7)}$	0.7	2.5588	100	0	0
$\widehat{\mathbf{K}}_4$	[0.7, 0.8)	2.1398	65.8777	1.4716	0.0523
$\mathbf{K}^{(0.8)}$	0.8	3.8824	100	0	0
$\widehat{\mathbf{K}}_5$	[0.8, 1)	3.4058	74.4747	1.7715	0.0362

3.3 APROXIMAÇÕES BASEADAS EM FUNÇÕES INTEIRAS

Na presente seção, vamos apresentar a formalização matemática utilizada para derivar aproximações para a KLT baseada na mesma metodologia apresentada em [28] para encontrar aproximações para a DCT. Analogamente a estrutura para derivar a RKLT, esta metodologia aplica funções inteiras de arredondamento na matriz da KLT exata para algum valor de ρ para encontrar novas transformadas aproximadas. O mapeamento adotado para derivar estas novas aproximações da KLT foi o seguinte:

$$\mathbb{R} \longrightarrow \mathcal{M}_8(\mathbb{Z}) \quad (3.4)$$

$$\alpha \longmapsto \text{int}(\alpha \cdot \mathbf{C}), \quad (3.5)$$

em que $\mathcal{M}_8(\mathbb{Z})$ é o espaço de matrizes 8×8 sob o conjunto dos inteiros \mathbb{Z} , $\text{int}(\cdot)$ é uma função inteira definida e α é um parâmetro definido como um fator de expansão. A matriz de transformação das novas aproximações para a KLT é definida por:

$$\mathbf{T} \triangleq \text{int}(\alpha \cdot \mathbf{K}^{(\rho)}), \quad (3.6)$$

em que $\mathbf{K}^{(\rho)}$ é a matriz da KLT exata de comprimento $N = 8$ e um valor do coeficiente de correlação ρ definido. A função $\text{int}(\cdot)$ é definida por uma coleção de funções de arredondamento inteiras, apresentadas na Equação (2.45).

A busca computacional por novas aproximações para a KLT baseadas em funções inteiras de arredondamento foi feita de modo que as novas transformadas apresentassem as seguintes características:

- (i) a matriz $\widehat{\mathbf{K}}$ deve apresentar elementos definidos no conjunto $\mathcal{C} = \{0, \pm 1, \pm 2, \pm 3\}$;
- (ii) $\widehat{\mathbf{K}} \cdot \widehat{\mathbf{K}}^\top$ deve ser uma matriz diagonal, ou apresentar um desvio de diagonalidade muito pequeno conforme apresentado na Equação (2.30) na Condição (2.31).

A fim de que os elementos de $\widehat{\mathbf{K}}$ estejam definidos em \mathcal{C} devemos restringir o intervalo dos valores de α . Para isto basta resolver a seguinte inequação para α :

$$0 \leq \text{int}(\alpha \cdot \gamma) \leq 3, \quad (3.7)$$

em que γ será o maior elemento da matriz da KLT exata dado um valor definido de ρ . Desta forma, os intervalos de α estão definidos na Tabela 7.

Tabela 7 – Intervalos de α

Função	Intervalo de α
<i>Floor</i>	$\left[\frac{1}{\gamma}, \frac{3}{\gamma}\right]$
<i>Ceil</i>	$\left[0, \frac{3}{\gamma}\right]$
<i>Trunc</i>	$\left[\frac{1}{\gamma}, \frac{4}{\gamma}\right]$
<i>Round</i>	$\left[\frac{1}{\gamma}, \frac{3}{\gamma}\right]$

Por exemplo, os valores de γ para a KLT considerando os valores de $\rho = 0.1, 0.5$ e 0.9 são, respectivamente, $0.469, 0.481$ e 0.489 . A Figura 2 apresenta o gráfico dos valores de γ para os valores de ρ variando de 0.1 até 0.9 , conforme considerado neste trabalho.

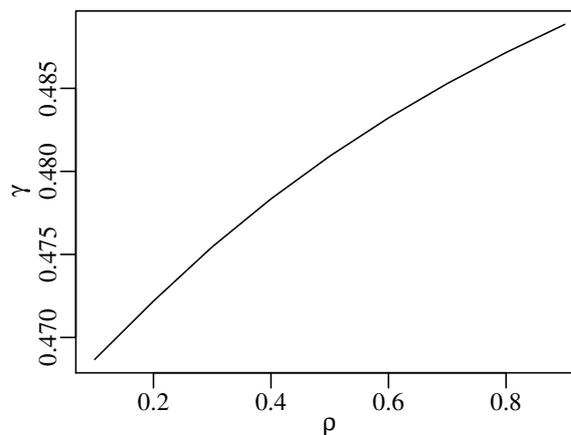


Figura 2 – Valores de γ considerando a KLT para diferentes valores de $\rho \in (0, 1)$.

De modo a encontrarmos as transformadas ótimas, precisamos buscar os valores de α dentro destes intervalos que otimizem as figuras de mérito utilizadas neste

trabalho. Este problema de otimização é apresentado na subseção seguinte.

3.4 OTIMIZAÇÃO

A otimização numérica é uma ferramenta muito importante na tomada de decisões em diversas áreas. Para que possamos utilizar esta ferramenta, precisamos identificar o objetivo da busca, uma medida quantitativa a ser otimizada [78]. Matematicamente, a otimização é a minimização ou a maximização de uma função sujeita a restrições de acordo com suas variáveis.

A busca computacional das transformadas foi realizada com o objetivo de otimizar as medidas utilizadas como figuras de mérito apresentadas na Seção 2.4, quais sejam:

- Ganho de codificação unificado ($\text{Cg}(\hat{\mathbf{K}})$);
- Eficiência da transformada ($\eta(\hat{\mathbf{K}})$);
- Erro quadrático médio ($\text{MSE}(\mathbf{K}^{(\rho)}, \hat{\mathbf{K}})$);
- Erro de energia total ($\epsilon(\mathbf{K}^{(\rho)}, \hat{\mathbf{K}})$).

As métricas ganho de codificação unificado e eficiência da transformada medem o poder de codificação das transformadas propostas; neste caso, queremos maximizar estas medidas. Já as métricas de similaridade, erro quadrático médio e erro de energia total, medem o quão distante as transformadas estão da KLT exata para dado valor de ρ . Considerando estas figuras de mérito, objetivamos minimizar estas medidas.

Como os elementos das matrizes $\hat{\mathbf{K}}$, transformadas aproximadas propostas, são valores no conjunto $\mathcal{C} = \{0, \pm 1, \pm 2, \pm 3\}$ e, como estamos considerando $N = 8$, temos $7^8 = 5764801$ matrizes candidatas a serem consideradas no problema de otimização.

3.4.1 Função objetiva

Para realizar a busca pelas transformadas ótimas segundo as métricas consideradas, o seguinte problema de otimização foi proposto:

$$\hat{\mathbf{K}}^* = \arg \underset{\hat{\mathbf{K}}}{\text{opt}} f(\hat{\mathbf{K}}), \quad (3.8)$$

em que $\hat{\mathbf{K}}$ são as matrizes candidatas a solução do problema, e

$$f \in \{\text{Cg}(\hat{\mathbf{K}}), \eta(\hat{\mathbf{K}}), \text{MSE}(\mathbf{K}^{(\rho)}, \hat{\mathbf{K}}), \epsilon(\mathbf{K}^{(\rho)}, \hat{\mathbf{K}})\}, \quad (3.9)$$

que são as figuras de mérito a serem otimizadas. Se considerarmos as medidas de codificação, ganho de codificação unificado e eficiência da transformada, o problema de otimização será:

$$\widehat{\mathbf{K}}^* = \arg \max_{\widehat{\mathbf{K}}} f(\widehat{\mathbf{K}}). \quad (3.10)$$

Já se considerarmos as medidas de similaridade com a KLT exata dado um valor de ρ , erro quadrático médio e erro de energia total, a função objetivo será dada da seguinte maneira:

$$\widehat{\mathbf{K}}^* = \arg \min_{\widehat{\mathbf{K}}} f(\widehat{\mathbf{K}}). \quad (3.11)$$

3.4.2 Espaço de busca

O espaço de busca das transformadas foi definido como segue. Seja

$$\widehat{\mathbf{K}} = \sqrt{\text{diag}(\mathbf{T} \cdot \mathbf{T}^\top)^{-1}} \cdot \mathbf{T}. \quad (3.12)$$

Em princípio, $\mathbf{T} \in \mathcal{M}_{\mathcal{C}}(8)$ em que $\mathcal{M}_{\mathcal{C}}(8)$ é o espaço de matrizes 8×8 com elementos pertencentes ao conjunto $\mathcal{C} = \{0, \pm 1, \pm 2, \pm 3\}$. Propomos a busca em um subconjunto de $\mathcal{M}_{\mathcal{C}}(8)$:

$$\mathcal{E}_\alpha = \{\mathbf{T} \in \mathcal{M}_{\mathcal{C}}(8) : \mathbf{T} = \text{int}(\alpha \cdot \mathbf{K}^{(\rho)})\}, \quad (3.13)$$

em que $\text{int} \in \{\text{floor}, \text{ceil}, \text{trunc}, \text{round}\}$ e α é definido nos intervalos descritos na Tabela 7. Consideramos o fator de expansão α assim como em [14, 28, 79]. Portanto, o espaço de busca é:

$$\mathcal{E} = \bigcup_{\alpha} \mathcal{E}_\alpha. \quad (3.14)$$

3.4.3 Restrições

O seguinte problema de otimização é um problema restrito. As matrizes candidatas à transformadas ótimas foram sujeitas às seguintes condições, que já foram previamente discutidas na Seção 3.3:

- Os elementos da matriz \mathbf{T} devem estar definidos no conjunto $\mathcal{C} = \{0, \pm 1, \pm 2, \pm 3\}$. Esta condição já foi implicitamente adotada na escolha do espaço de busca das matrizes.
- \mathbf{T} deve ser ortogonal ou quase-ortogonal, conforme a Definição 3.

O Algoritmo 1 apresenta o pseudocódigo de como foi realizada essa busca computacional das transformadas ótimas dado o problema de otimização descrito acima. O motivo de considerarmos apenas as matrizes da KLT para valores de ρ variando de 0.1 até 0.9 com passos de 10^{-1} é o fato de que esta transformada é sensível a este parâmetro. Para cada passo de ρ obtemos uma matriz da KLT que, por sua vez, pode resultar em diferentes aproximações.

Algoritmo 1: Pseudocódigo para a busca de transformadas ótimas

```

início
  para  $\rho$  variando de 0.1 até 0.9 com passos de  $10^{-1}$  faça
    Determinar a matriz da KLT exata,  $\mathbf{K}^{(\rho)}$ ;
    para  $\alpha$  variando no intervalo definido pela função com passos de  $10^{-2}$ 
      faça
         $\mathbf{T} = \text{int}(\alpha \cdot \mathbf{K}^{(\rho)})$ ;
         $\widehat{\mathbf{K}} = \widehat{\mathbf{S}} \cdot \mathbf{T}$ ;
        Cálculo das figuras de mérito de  $\widehat{\mathbf{K}}$ ;
        Seleção dos intervalos de  $\alpha$  que retornam as transformadas ótimas
          segundo cada figura de mérito
      fim
    fim
  fim
fim

```

Como o objetivo deste trabalho é encontrar novas transformadas aproximadas de baixa complexidade que possam ser implementadas em *hardware*, por exemplo, não é interessante termos um número muito grande de transformadas, mas sim, transformadas que representem bem todas as suas matrizes vizinhas em relação ao coeficiente de correlação ρ . Ou seja, queremos encontrar aproximações que sejam uma boa representação de transformadas para um sinal que apresente baixa correlação, e outras que sejam uma boa representação de uma transformada para um sinal altamente correlacionado, por exemplo.

De modo que estamos considerando quatro figuras de mérito e avaliando cada uma delas de maneira separada, poderemos ter até quatro transformadas ótimas para cada intervalo de ρ e função inteira fixados.

3.5 RESULTADOS

A partir da solução do problema de otimização descrito previamente, obtivemos os seguintes resultados, fixando $N = 8$ e considerando cada função de arredondamento e cada figura de mérito.

As Tabelas 8, 9, 10 e 11 apresentam os intervalos de α que retornam as transformadas ótimas para as funções piso, teto, de truncamento e arredondamento, respectivamente, de acordo com um intervalo de ρ prefixado. Cada tabela está organizada de acordo com as figuras de mérito utilizadas para a avaliação das transformadas: ganho de codificação unificado, eficiência da transformada, erro de energia total e erro quadrático médio.

Tabela 8 – Transformadas encontradas considerando a função piso

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	2.18	(0, 0.1]	(5.82, 6.03)
(0.1, 0.2]	(2.57, 2.62)	(0.1, 0.2]	(4.7, 4.83)
(0.2, 0.3]	(4.88, 4.97)	(0.2, 0.3]	(4.7, 4.83)
(0.3, 0.4]	(4.65, 4.69)	(0.3, 0.4]	(4.7, 4.83)
(0.4, 0.5]	(4.86, 5.06)	(0.4, 0.5]	(4.93, 5.13)
(0.5, 0.6]	(4.93, 5.13)	(0.5, 0.6]	(4.93, 5.13)
(0.6, 0.7]	(5.09, 5.21)	(0.6, 0.7]	(5.09, 5.21)
(0.7, 1)	5.73	(0.7, 1)	5.73
(a) Ganho de Codificação		(b) Eficiência da Transformada	
Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(5.82, 6.03)	(0, 0.1]	(5.82, 6.03)
(0.1, 0.3]	(5.66, 5.77)	(0.1, 0.3]	(5.66, 5.77)
(0.3, 0.4]	(5.85, 6.12)	(0.3, 0.4]	(5.85, 6.12)
(0.4, 0.5]	(6.09, 6.12)	(0.4, 0.5]	(5.11, 5.29)
(0.5, 0.6]	(6.04, 6.12)	(0.5, 0.6]	(5.22, 5.40)
(0.6, 0.7]	(5.22, 5.40)	(0.6, 0.7]	(5.22, 5.40)
(0.7, 0.8]	(5.83, 6.12)	(0.7, 0.8]	(5.74, 5.88)
(0.8, 1)	(5.02, 5.27)	(0.8, 1)	(5.74, 5.88)
(c) Erro de Energia Total		(d) Erro Quadrático Médio	

Tabela 9 – Transformadas encontradas considerando a função teto

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(2.38, 2.40)	(0, 0.1]	(5.4, 5.81)
(0.1, 0.3]	(5.78, 6.12)	(0.1, 0.2]	(5.14, 5.25)
(0.3, 0.4]	(5.68, 5.84)	(0.2, 0.3]	(5.78, 6.12)
(0.4, 0.6]	(5.89, 5.98)	(0.3, 0.4]	5.67
(0.6, 0.7]	(5.45, 5.89)	(0.4, 0.7]	(5.45, 5.89)
(0.7, 1)	(6, 6.12)	(0.7, 1)	(6, 6.12)

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(6.04, 6.12)	(0, 0.1]	(6.04, 6.12)
(0.1, 0.2]	(5.96, 6.12)	(0.1, 0.2]	(5.96, 6.12)
(0.2, 0.4]	(5.68, 5.84)	(0.2, 0.4]	(5.68, 5.84)
(0.4, 0.6]	(5.89, 5.98)	(0.4, 0.6]	(5.89, 5.98)
(0.6, 0.7]	(5.45, 5.89)	(0.6, 0.7]	(5.45, 5.89)
(0.7, 1)	(6, 6.12)	(0.7, 1)	(6, 6.12)

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(2.38, 2.39)	(0, 0.1]	(3.47, 4.26)
(0.1, 0.3]	4.55	(0.1, 0.3]	(7.71, 7.92)
(0.3, 0.4]	4.55	(0.3, 0.4]	(8.06, 8.16)
(0.4, 0.5]	(6.09, 6.13)	(0.4, 0.5]	(7.66, 7.93)
(0.5, 0.6]	(5.99, 6.03)	(0.5, 0.6]	7.26
(0.6, 0.7]	7.26	(0.6, 0.7]	7.26
(0.7, 0.8]	(6.83, 6.92)	(0.7, 0.8]	(6.83, 6.92)
(0.8, 1)	(6.92, 6.98)	(0.8, 1)	(6.92, 6.98)

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(7.62, 8.16)	(0, 0.1]	(7.62, 8.16)
(0.1, 0.2]	(7.94, 8.16)	(0.1, 0.2]	(7.94, 8.16)
(0.2, 0.3]	(7.71, 7.92)	(0.2, 0.3]	(7.71, 7.92)
(0.3, 0.4]	(8.04, 8.05)	(0.3, 0.4]	(8.04, 8.05)
(0.4, 0.5]	(7.94, 8.16)	(0.4, 0.5]	(7.5, 7.6)
(0.5, 0.6]	(8.05, 8.16)	(0.5, 0.6]	(8.02, 8.04)
(0.6, 0.7]	8.16	(0.6, 0.7]	7.26
(0.7, 0.8]	8.16	(0.7, 0.8]	(8.03, 8.16)
(0.8, 1)	(7.25, 7.65)	(0.8, 1)	(6.92, 6.98)

Tabela 10 – Transformadas encontradas considerando a função de truncamento

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(2.38, 2.39)	(0, 0.1]	(3.47, 4.26)
(0.1, 0.3]	4.55	(0.1, 0.3]	(7.71, 7.92)
(0.3, 0.4]	4.55	(0.3, 0.4]	(8.06, 8.16)
(0.4, 0.5]	(6.09, 6.13)	(0.4, 0.5]	(7.66, 7.93)
(0.5, 0.6]	(5.99, 6.03)	(0.5, 0.6]	7.26
(0.6, 0.7]	7.26	(0.6, 0.7]	7.26
(0.7, 0.8]	(6.83, 6.92)	(0.7, 0.8]	(6.83, 6.92)
(0.8, 1)	(6.92, 6.98)	(0.8, 1)	(6.92, 6.98)

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(7.62, 8.16)	(0, 0.1]	(7.62, 8.16)
(0.1, 0.2]	(7.94, 8.16)	(0.1, 0.2]	(7.94, 8.16)
(0.2, 0.3]	(7.71, 7.92)	(0.2, 0.3]	(7.71, 7.92)
(0.3, 0.4]	(8.04, 8.05)	(0.3, 0.4]	(8.04, 8.05)
(0.4, 0.5]	(7.94, 8.16)	(0.4, 0.5]	(7.5, 7.6)
(0.5, 0.6]	(8.05, 8.16)	(0.5, 0.6]	(8.02, 8.04)
(0.6, 0.7]	8.16	(0.6, 0.7]	7.26
(0.7, 0.8]	8.16	(0.7, 0.8]	(8.03, 8.16)
(0.8, 1)	(7.25, 7.65)	(0.8, 1)	(6.92, 6.98)

Tabela 11 – Transformadas encontradas considerando a função de arredondamento

Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(1.19, 1.2)	(0, 0.1]	(1.74, 2.69)
(0.1, 0.4]	(1.22, 1.25)	(0.1, 0.4]	(6.09, 6.12)
(0.4, 0.5]	(3.75, 3.8)	(0.4, 0.5]	(6.08, 6.12)
(0.5, 0.6]	3.64	(0.5, 0.6]	(6.08, 6.12)
(0.6, 0.7]	(3.69, 3.81)	(0.6, 0.7]	(6.05, 6.12)
(0.7, 0.8]	(5.34, 5.35)	(0.7, 0.8]	6.03
(0.8, 1)	(4.01, 4.08)	(0.8, 1)	(4.01, 4.08)
(a) Ganho de Codificação		(b) Eficiência da Transformada	
Intervalo de ρ	Intervalo de α	Intervalo de ρ	Intervalo de α
(0, 0.1]	(5.46, 5.92)	(0, 0.1]	(5.46, 5.92)
(0.1, 0.2]	6.12	(0.1, 0.2]	6.12
(0.2, 0.4]	(6.09, 6.12)	(0.2, 0.4]	(6.09, 6.12)
(0.4, 0.5]	(6.08, 6.12)	(0.4, 0.5]	(6.08, 6.12)
(0.5, 0.6]	(6.08, 6.12)	(0.5, 0.6]	(6.08, 6.12)
(0.6, 0.7]	(6.05, 6.12)	(0.6, 0.7]	(6.05, 6.12)
(0.7, 0.8]	(6.04, 6.12)	(0.7, 0.8]	6.03
(0.8, 1)	(6.03, 6.12)	(0.8, 1)	(5.56, 5.82)
(c) Erro de Energia Total		(d) Erro Quadrático Médio	

3.5.1 Refinamento dos resultados

A metodologia acima resultou em um número relativamente grande de novas matrizes aproximadas (144 matrizes). Entretanto, no específico contexto de *hardware* de baixo consumo e baixo custo, não é viável a realização/implementação de várias matrizes. Deve-se, portanto, de algum modo refinar o resultado anterior, buscando critérios para selecionar a partir das transformadas propostas as transformadas candidatas à eventual implementação.

A fim de refinar o número de transformadas, propomos um procedimento em duas etapas. Na primeira etapa, apenas consideramos dentre as 144 transformadas aquelas que obtiveram melhor desempenho, para cada intervalo de ρ , de acordo com as figuras de mérito. Este procedimento causou uma redução de 13.89% no número de transformadas. A Tabela 12 apresenta as 20 transformadas ótimas encontradas na primeira etapa do refinamento e suas respectivas medidas de codificação e similaridade. As matrizes apresentadas nessa tabela estão listadas no Apêndice A.

Na segunda etapa do refinamento, objetivamos encontrar intervalos de ρ em que as transformadas obtivessem desempenho similar entre si de acordo com o ganho de

Tabela 12 – Medidas de codificação e similaridade das transformadas ótimas obtidas

Transformada	Intervalo de ρ	$Cg(\widehat{\mathbf{K}})$	$\eta(\widehat{\mathbf{K}})$	$\epsilon(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$	$MSE(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$
$\widehat{\mathbf{K}}_6$	(0, 0.1]	0.0308	93.4298	1.5331	0.0608
$\widehat{\mathbf{K}}_7$	(0, 0.1]	0.1325	79.5971	0.3173	0.0128
$\widehat{\mathbf{K}}_8$	(0, 0.1]	0.0588	88.3104	0.093	0.0036
$\widehat{\mathbf{K}}_9$	(0.1, 0.2]	0.1754	83.7756	0.2265	0.0094
$\widehat{\mathbf{K}}_{10}$	(0.2, 0.3]	0.3461	80.8238	0.2999	0.0132
$\widehat{\mathbf{K}}_{11}$	(0.3, 0.4]	0.6725	83.0728	0.3104	0.0095
$\widehat{\mathbf{K}}_{12}$	(0.3, 0.4]	0.7618	63.712	2.1348	0.0785
$\widehat{\mathbf{K}}_{13}$	(0.3, 0.4]	0.6532	83.3729	0.2823	0.0115
$\widehat{\mathbf{K}}_{14}$	(0.4, 0.5]	1.1063	87.2737	0.3487	0.0094
$\widehat{\mathbf{K}}_{15}$	(0.4, 0.5]	1.153	77.5984	0.6439	0.0163
$\widehat{\mathbf{K}}_{16}$	(0.5, 0.6]	1.7572	82.7462	0.9273	0.0197
$\widehat{\mathbf{K}}_{17}$	(0.5, 0.6]	1.6743	86.4929	0.275	0.0089
$\widehat{\mathbf{K}}_{18}$	(0.6, 0.7]	2.5736	84.7636	0.7505	0.0153
$\widehat{\mathbf{K}}_{19}$	(0.6, 0.7]	2.5308	89.7579	0.2299	0.0065
$\widehat{\mathbf{K}}_{20}$	(0.7, 0.8]	3.8534	84.1782	0.6043	0.0087
$\widehat{\mathbf{K}}_{21}$	(0.7, 0.8]	3.8484	87.7103	0.2418	0.0043
$\widehat{\mathbf{K}}_{22}$	(0.7, 0.8]	3.8146	86.6308	0.1884	0.0049
$\widehat{\mathbf{K}}_{23}$	(0.8, 1)	6.2462	88.1734	0.6746	0.0102
$\widehat{\mathbf{K}}_{24}$	(0.8, 1)	6.1727	85.8301	0.1948	0.0055
$\widehat{\mathbf{K}}_{25}$	(0.8, 1)	6.2335	86.827	0.4439	0.005

codificação unificado, uma vez que essa métrica apresenta informações sobre a capacidade de codificação da transformação ortogonal para aplicações de compressão de dados. A Figura 3 apresenta a ideia gráfica do que pretende-se encontrar: transformadas $\widehat{\mathbf{K}}_*$ que representem as transformadas de algum intervalo de ρ .

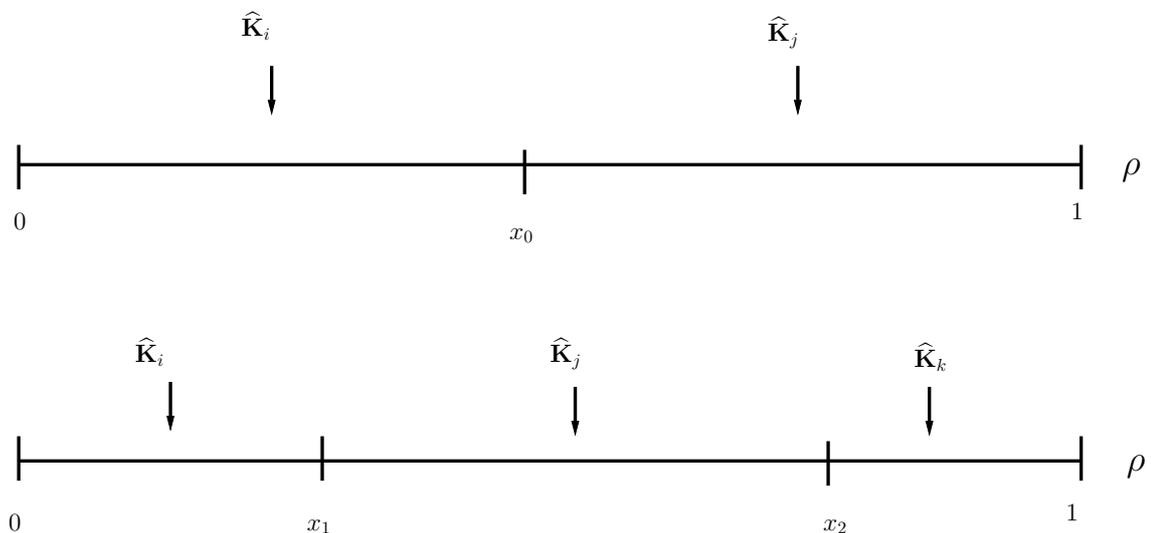


Figura 3 – Separação dos intervalos de ρ em grupos.

Uma possibilidade de se realizar esse refinamento é através de agrupamento, como por exemplo, agrupamento *k-means* [76]. O uso de agrupamento pode resultar em um número reduzido de grupos em que apenas uma matriz em cada grupo pode ser escolhida como representativa do grupo. Dessa forma, a segunda etapa do refinamento foi utilizar o método de agrupamento *k-means*, que está apresentado no Algoritmo 2 [4], para separar as transformadas em grupos distintos. A escolha do algoritmo de agrupamento *k-means* se deu pelo fato deste método de agrupamento ser um dos algoritmos mais simples que usa o método de aprendizado não supervisionado para resolver problemas de *cluster* [4].

Algoritmo 2: Agrupamento *k-means* utilizando distância Euclidiana

Entrada: dados $D \subseteq \mathbb{R}^d$; número de clusters $C \in \mathbb{N}$.

Saída: média dos C clusters $\mu_1, \dots, \mu_c \in \mathbb{R}^d$

início

 inicializar aleatoriamente os C vetores $\mu_1, \dots, \mu_c \in \mathbb{R}^d$

repita

 atribuir cada $x \in D$ para $\arg \min_j \text{Dis}(x, \mu_j)$;

para $j = 1$ até C **faça**

$D_j \leftarrow \{x \in D | x \text{ atribuído ao cluster } C\}$;

$\mu_j = \frac{1}{|D_j|} \sum_{x \in D_j} x$;

fim

até não haja mais mudanças em μ_1, \dots, μ_c ;

retorna μ_1, \dots, μ_c

fim

Para auxiliar na escolha do número de grupos do algoritmo *k-means*, foi utilizado o método hierárquico [80], apresentado na Figura 4 pelo dendrograma. Um dendrograma é um diagrama representado por uma árvore. O dendrograma mostra uma coluna de 20 nós representando os dados iniciais (ganho de codificação unificado de cada transformada), e os nós restantes representam os *clusters* aos quais os dados pertencem. As caixas vermelhas exibidas no dendrograma representam os grupos definidos. Inicialmente, cada matriz é considerada um grupo. Em seguida, as matrizes são agrupadas aos vizinhos mais próximos segundo a medida de distância euclidiana. Dessa maneira, obtivemos dois grupos distintos C_1 e C_2 . O primeiro grupo apresentou transformadas obtidas com os valores de ρ variando de $(0, 0.7]$ e o segundo grupo transformadas obtidas com os valores de $\rho \in (0.7, 1)$.

A Tabela 13 apresenta as 20 transformadas ótimas e suas figuras de mérito

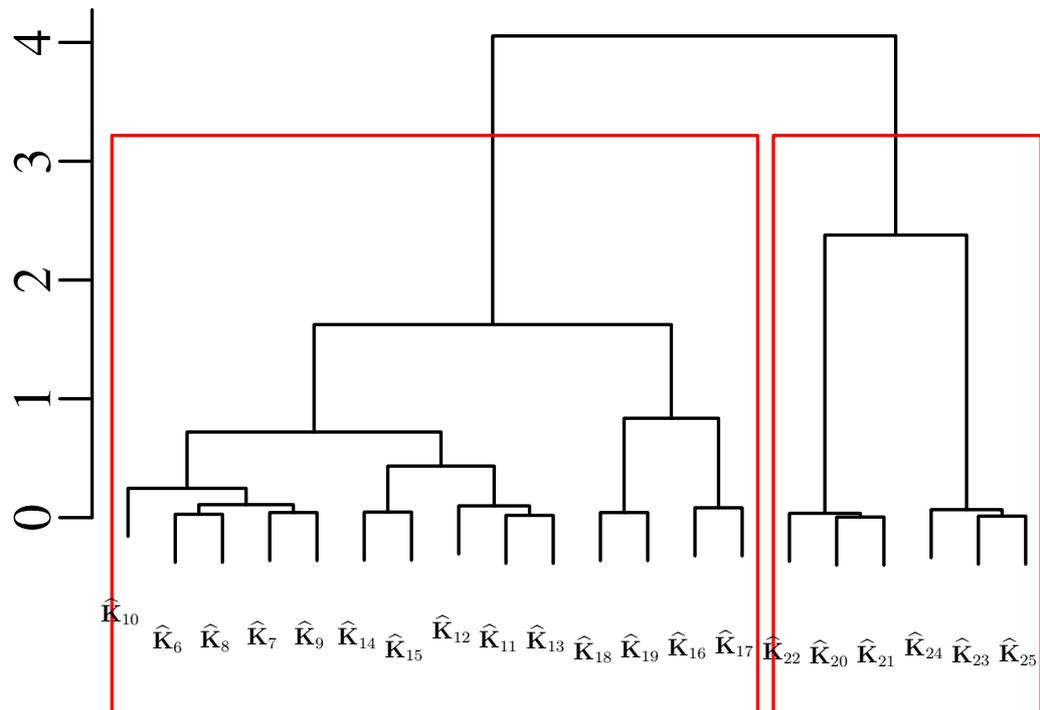


Figura 4 – Dendrograma considerando o ganho de codificação unificado das transformadas propostas.

e os dois grupos definidos pelo agrupamento. A fim de elegermos a transformada ótima que representará as demais transformadas do grupo, decidimos considerar aquela que apresentou a melhor medida de ganho de codificação em cada grupo, que foram as matrizes $\hat{\mathbf{K}}_{18}$ no grupo C_1 e $\hat{\mathbf{K}}_{23}$ no grupo C_2 . Decidimos considerar também, em cada grupo, as transformadas que apresentaram melhor desempenho considerando as demais figuras de mérito. Estas medidas estão apresentadas em negrito na Tabela 13.

Tabela 13 – Medidas de codificação e similaridade das transformadas ótimas considerando os dois grupos

Transformada	Intervalo de ρ	Cg($\widehat{\mathbf{K}}$)	$\eta(\widehat{\mathbf{K}})$	$\epsilon(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$	MSE($\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}}$)
$\widehat{\mathbf{K}}_6$	(0, 0.1]	0.0308	93.4298	1.5331	0.0608
$\widehat{\mathbf{K}}_7$	(0, 0.1]	0.1325	79.5971	0.3173	0.0128
$\widehat{\mathbf{K}}_8$	(0, 0.1]	0.0588	88.3104	0.093	0.0036
$\widehat{\mathbf{K}}_9$	(0.1, 0.2]	0.1754	83.7756	0.2265	0.0094
$\widehat{\mathbf{K}}_{10}$	(0.2, 0.3]	0.3461	80.8238	0.2999	0.0132
$\widehat{\mathbf{K}}_{11}$	(0.3, 0.4]	0.6725	83.0728	0.3104	0.0095
$\widehat{\mathbf{K}}_{12}$	(0.3, 0.4]	0.7618	63.712	2.1348	0.0785
$\widehat{\mathbf{K}}_{13}$	(0.3, 0.4]	0.6532	83.3729	0.2823	0.0115
$\widehat{\mathbf{K}}_{14}$	(0.4, 0.5]	1.1063	87.2737	0.3487	0.0094
$\widehat{\mathbf{K}}_{15}$	(0.4, 0.5]	1.153	77.5984	0.6439	0.0163
$\widehat{\mathbf{K}}_{16}$	(0.5, 0.6]	1.7572	82.7462	0.9273	0.0197
$\widehat{\mathbf{K}}_{17}$	(0.5, 0.6]	1.6743	86.4929	0.275	0.0089
$\widehat{\mathbf{K}}_{18}$	(0.6, 0.7]	2.5736	84.7636	0.7505	0.0153
$\widehat{\mathbf{K}}_{19}$	(0.6, 0.7]	2.5308	89.7579	0.2299	0.0065
$\widehat{\mathbf{K}}_{20}$	(0.7, 0.8]	3.8534	84.1782	0.6043	0.0087
$\widehat{\mathbf{K}}_{21}$	(0.7, 0.8]	3.8484	87.7103	0.2418	0.0043
$\widehat{\mathbf{K}}_{22}$	(0.7, 0.8]	3.8146	86.6308	0.1884	0.0049
$\widehat{\mathbf{K}}_{23}$	(0.8, 1)	6.2462	88.1734	0.6746	0.0102
$\widehat{\mathbf{K}}_{24}$	(0.8, 1)	6.1727	85.8301	0.1948	0.0055
$\widehat{\mathbf{K}}_{25}$	(0.8, 1)	6.2335	86.827	0.4439	0.005

Assim, o grupo C_1 representa as matrizes obtidas para valores de $\rho \in (0, 0.7]$ e as transformadas que retornaram as melhores medidas foram $\widehat{\mathbf{K}}_6$, $\widehat{\mathbf{K}}_8$ e $\widehat{\mathbf{K}}_{18}$. As matrizes \mathbf{T}_6 , \mathbf{T}_8 e \mathbf{T}_{18} das respectivas transformadas são dadas a seguir:

$$\mathbf{T}_6 = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 1 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & 0 & -1 & 1 & 1 & -1 & 0 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 1 & -1 & 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}, \quad (3.15)$$

$$\mathbf{T}_8 = \begin{bmatrix} 1 & 2 & 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 3 & 3 & 1 & -1 & -3 & -3 & -2 \\ 3 & 3 & 0 & -3 & -3 & 0 & 3 & 3 \\ 3 & 1 & -3 & -2 & 2 & 3 & -1 & -3 \\ 3 & -1 & -3 & 2 & 2 & -3 & -1 & 3 \\ 3 & -3 & 0 & 3 & -3 & 0 & 3 & -3 \\ 2 & -3 & 3 & -1 & -1 & 3 & -3 & 2 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}, \quad (3.16)$$

$$\mathbf{T}_{18} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 & -1 & -2 & -2 \\ 2 & 1 & 0 & -2 & -2 & 0 & 1 & 2 \\ 2 & 0 & -2 & -1 & 1 & 2 & 0 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 2 & -2 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix}. \quad (3.17)$$

Já o grupo C_2 representa as matrizes obtidas para valores de $\rho \in (0.7, 1)$ e as transformadas que obtiveram melhores medidas neste grupo foram $\widehat{\mathbf{K}}_{21}$, $\widehat{\mathbf{K}}_{22}$ e $\widehat{\mathbf{K}}_{23}$. As matrizes \mathbf{T}_{21} , \mathbf{T}_{22} e \mathbf{T}_{23} das respectivas transformadas são dadas a seguir:

$$\mathbf{T}_{21} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 2 & -1 & -3 & -3 & -1 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 1 & 2 & -2 & -1 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}, \quad (3.18)$$

$$\mathbf{T}_{22} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 2 & -1 & -3 & -3 & -1 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 1 & 3 & -3 & -1 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}, \quad (3.19)$$

$$\mathbf{T}_{23} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 1 & 0 & 0 & -1 & -2 & -2 \\ 2 & 1 & -1 & -2 & -2 & -1 & 1 & 2 \\ 2 & 0 & -2 & -1 & 1 & 2 & 0 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 2 & -2 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix}. \quad (3.20)$$

A Tabela 14 apresenta as transformadas, suas respectivas diagonais $\hat{\mathbf{S}}$ e a medida de desvio de diagonalidade. Apenas a transformada \mathbf{T}_6 é ortogonal. Podemos perceber que todas as matrizes possuem um desvio de diagonalidade bem mais baixo do que a condição $\delta(\mathbf{T} \cdot \mathbf{T}^\top) \leq 1 - \frac{2}{\sqrt{5}} \approx 0.1056$ dada na Equação (2.31), o que implica dizer que as matrizes são quase-ortogonais.

Tabela 14 – Aproximadas propostas e suas respectivas matrizes diagonais

Transformada	Ortogonal?	$\hat{\mathbf{S}}$	$\delta(\mathbf{K} \cdot \mathbf{K}^\top)$
\mathbf{T}_6	Sim	$\text{diag} \left(\frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{6}} \right)$	-
\mathbf{T}_8	Não	$\text{diag} \left(\frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{54}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{54}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}} \right)$	0.0056
\mathbf{T}_{18}	Não	$\text{diag} \left(\frac{1}{\sqrt{14}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{18}} \right)$	0.0395
\mathbf{T}_{21}	Não	$\text{diag} \left(\frac{1}{\sqrt{32}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{44}}, \frac{1}{\sqrt{32}}, \frac{1}{\sqrt{36}}, \frac{1}{\sqrt{40}}, \frac{1}{\sqrt{46}} \right)$	0.0133
\mathbf{T}_{22}	Não	$\text{diag} \left(\frac{1}{\sqrt{32}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{44}}, \frac{1}{\sqrt{32}}, \frac{1}{\sqrt{46}}, \frac{1}{\sqrt{40}}, \frac{1}{\sqrt{46}} \right)$	0.0094
\mathbf{T}_{23}	Não	$\text{diag} \left(\frac{1}{\sqrt{14}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{18}}, \frac{1}{\sqrt{20}}, \frac{1}{\sqrt{18}} \right)$	0.0360

A Tabela 15 apresenta as medidas de codificação e similaridade das transformadas propostas em comparação a KLT para $\rho = 0.4$ e $\rho = 0.9$ ($\mathbf{K}^{(0.4)}$ e $\mathbf{K}^{(0.9)}$), com a DCT e as transformadas aproximadas SDCT e RDCT, e com a SKLT₁ que é a aproximação para a KLT baseada na proposta de Haweel [31] para $\rho \leq 0.6$. A medida de ganho de codificação unificado depende do valor de ρ . Podemos perceber isso ao avaliar esta medida considerando a KLT para $\rho = 0.4$. Por este motivo, não podemos comparar as transformadas aproximadas que são obtidas para valores mais baixos de ρ com a DCT e as aproximações da DCT (que consideram $\rho \rightarrow 1$), por exemplo. Analisando as outras medidas, podemos perceber que as transformadas aproximadas propostas têm um comportamento muito bom, ganhando em muitos casos de transformadas clássicas da literatura como a SDCT e a RDCT.

Tabela 15 – Comparação das medidas de codificação e similaridade entre as transformadas listadas na literatura e as transformadas propostas

	$Cg(\widehat{\mathbf{K}})$	$\eta(\widehat{\mathbf{K}})$	$\epsilon(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$	$MSE(\mathbf{K}^{(\rho)}, \widehat{\mathbf{K}})$
$\mathbf{K}_8^{(0.4)}$	0.6625	100	0	0
$\mathbf{K}_8^{(0.9)}$	6.3109	100	0	0
DCT	8.8259	93.9911	0.0088	0.0007
SDCT	7.7870	82.5525	3.3065	0.0215
RDCT	8.1826	87.4297	1.7658	0.0101
SKLT ₁	1.2884	61.4431	3.5782	0.1144
$\widehat{\mathbf{K}}_6$	0.0308	93.4298	1.5331	0.0608
$\widehat{\mathbf{K}}_8$	0.0588	88.3104	0.093	0.0036
$\widehat{\mathbf{K}}_{18}$	2.5736	84.7636	0.7505	0.0153
$\widehat{\mathbf{K}}_{21}$	3.8484	87.7103	0.2418	0.0043
$\widehat{\mathbf{K}}_{22}$	3.8146	86.6308	0.1884	0.0049
$\widehat{\mathbf{K}}_{23}$	6.2462	88.1734	0.6746	0.0102

4 ALGORITMOS RÁPIDOS

A implementação direta de algumas transformadas pode requerer um número grande de operações de multiplicação, adição e deslocamento de bits. Por exemplo, considerando $N = 8$ e $\rho = 0.95$, a matriz da KLT é dada por:

$$\mathbf{K}^{(0.95)} = \begin{bmatrix} 0.338 & 0.351 & 0.360 & 0.364 & 0.364 & 0.360 & 0.351 & 0.338 \\ 0.481 & 0.420 & 0.286 & 0.101 & -0.101 & -0.286 & -0.420 & -0.481 \\ 0.467 & 0.207 & -0.179 & -0.456 & -0.456 & -0.179 & 0.207 & 0.467 \\ 0.423 & -0.085 & -0.487 & -0.278 & 0.278 & 0.487 & 0.085 & -0.423 \\ 0.360 & -0.347 & -0.356 & 0.351 & 0.351 & -0.356 & -0.347 & 0.360 \\ 0.283 & -0.488 & 0.094 & 0.415 & -0.415 & -0.094 & 0.488 & -0.283 \\ 0.195 & -0.462 & 0.460 & -0.190 & -0.190 & 0.460 & -0.462 & 0.195 \\ 0.100 & -0.279 & 0.416 & -0.490 & 0.490 & -0.416 & 0.279 & -0.100 \end{bmatrix}. \quad (4.1)$$

Aplicando $\mathbf{K}^{(0.95)}$ a um sinal de entrada \mathbf{x} , obtemos o sinal de saída \mathbf{y} :

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_7 \end{bmatrix} = \begin{bmatrix} k_{0,0} & k_{0,1} & \dots & k_{0,7} \\ k_{1,0} & k_{1,1} & \dots & k_{1,7} \\ \vdots & \vdots & \ddots & \vdots \\ k_{7,0} & k_{7,1} & \dots & k_{7,7} \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_7 \end{bmatrix} = \begin{bmatrix} k_{0,0} \cdot x_0 + k_{0,1} \cdot x_1 + \dots + k_{0,7} \cdot x_7 \\ k_{1,0} \cdot x_0 + k_{1,1} \cdot x_1 + \dots + k_{1,7} \cdot x_7 \\ \vdots \\ k_{7,0} \cdot x_0 + k_{7,1} \cdot x_1 + \dots + k_{7,7} \cdot x_7 \end{bmatrix}, \quad (4.2)$$

em que $k_{i,j}$ é (i, j) -ésimo elemento de $\mathbf{K}^{(0.95)}$. Note que aplicar a transformada $\mathbf{K}^{(0.95)}$ em um sinal de entrada \mathbf{x} demanda 64 multiplicações e 56 adições. Como o cômputo da KLT depende do sinal de entrada, é inviável a construção de algoritmos rápidos para ela. Por isso, surge a necessidade de trabalharmos com aproximações para a KLT, que independam do sinal de entrada.

Por não depender do sinal de entrada, a DCT ameniza este problema, mas seu cálculo ainda pode ser muito custoso. O cômputo direto da DCT em um sinal de entrada \mathbf{x} pode requerer até N^2 multiplicações e $N(N - 1)$ adições. Para endereçar essa dificuldade, são propostos algoritmos rápidos para a DCT, que consistem de procedimentos de custo computacional reduzido para realizar o cômputo da transformação. A literatura traz uma ênfase em métodos para o comprimendo de bloco $N = 8$ [19–26] devido a importância destas transformadas no contexto de compressão de imagens. A Tabela 16 apresenta os principais algoritmos rápidos propostos para a DCT de comprimento oito encontrados na literatura. Loeffler *et al.* [25] derivaram o algoritmo rápido que requer apenas 11 multiplicações, o que é considerado o mínimo teórico de complexidade multiplicativa [27]. Mesmo

reduzindo a complexidade aritmética através dos algoritmos, o uso da DCT ainda pode ser proibitivo pelo fato de que sua forma exata requer multiplicações por números reais. É por este motivo que aproximações de baixa complexidade para a DCT e KLT se tornam tão importantes em contextos que disponham de baixo poder de processamento.

Tabela 16 – Complexidade aritmética dos algoritmos rápidos da DCT-8

Algoritmo	Multiplicação	Adição
Loeffler <i>et al.</i> [25, 65, 81]	11	29
Suehiro e Hatori [82]	12	29
Yuan <i>et al.</i> [83, 84]	12	29
Lee [23, 85, 86]	12	29
Vetterli <i>et al.</i> [21]	12	29
Hou [20, 87, 88]	12	29
Wang [22]	13	29
Arai <i>et al.</i> [24]	13	29
Chen <i>et al.</i> [19, 89]	16	26
Feig e Winograd [26, 90]	22	28

Os algoritmos rápidos visam reduzir esse custo computacional – número de operações aritméticas – exigidos na aplicação de uma transformada. Um algoritmo pode ser definido como uma sequência finita de regras, que dado um conjunto de entradas, retorna uma saída [13]. Algoritmos rápidos visam processar estas entradas de maneira eficiente, resolvendo tarefas com o mínimo de cálculos necessários. Por exemplo, suponha que queremos calcular A , dado por [13]:

$$A = ac + ad + bc + bd. \quad (4.3)$$

Esta operação requer quatro multiplicações e três adições. Porém, podemos reescrever A da seguinte forma:

$$A = (a + b)(c + d), \quad (4.4)$$

o que requer apenas uma multiplicação e duas adições. Ambas Equações (4.3) e (4.4) retornam o mesmo valor de A , porém na Equação (4.4) podemos rearranjar adições e multiplicações de forma a reduzir o custo envolvido neste cálculo.

4.1 ALGORITMOS RÁPIDOS PROPOSTOS

Mesmo trabalhando com as transformadas aproximadas para a KLT, quando queremos aplicá-las a um sinal de entrada, ainda teremos um grande número de operações

As constantes m_k , $k = 0, 1, \dots, 17$ dependem da escolha da matriz \mathbf{T} e são apresentadas na Tabela 17.

Tabela 17 – Constantes necessárias para os algoritmos rápidos

Constantes	\mathbf{T}_6	\mathbf{T}_8	\mathbf{T}_{18}	\mathbf{T}_{21}	\mathbf{T}_{22}	\mathbf{T}_{23}
m_0	0	1	1	2	2	1
m_1	1	2	1	2	2	1
m_2	1	3	1	2	2	0
m_3	1	3	2	0	0	2
m_4	0	0	0	-1	-1	1
m_5	-1	-3	-2	3	3	-2
m_6	1	3	1	2	2	1
m_7	0	1	1	2	2	1
m_8	1	2	1	0	0	1
m_9	1	3	2	3	3	0
m_{10}	1	3	2	3	3	-2
m_{11}	0	-1	-1	1	1	-1
m_{12}	0	1	0	1	1	0
m_{13}	1	3	2	3	3	2
m_{14}	1	2	2	3	3	2
m_{15}	0	1	0	0	0	0
m_{16}	1	3	2	2	3	2
m_{17}	0	0	0	1	1	0

As Figuras 5, 6 e 7 apresentam os diagramas de fluxo e sinal dos algoritmos rápidos. Os diagramas relacionam os dados de entrada x_n , $n = 0, 1, \dots, 7$, aos coeficientes da saída X_k , $k = 0, 1, \dots, 7$, resultando em $\mathbf{X} = \mathbf{K} \cdot \mathbf{x}$. No diagrama, as setas tracejadas representam multiplicações por -1 . Além disso, quando há o encontro de duas ou mais setas, seus valores são somados [13]. É importante ressaltar que os Blocos A e B são os mesmos em todos os diagramas e são apresentados na Figura 8.

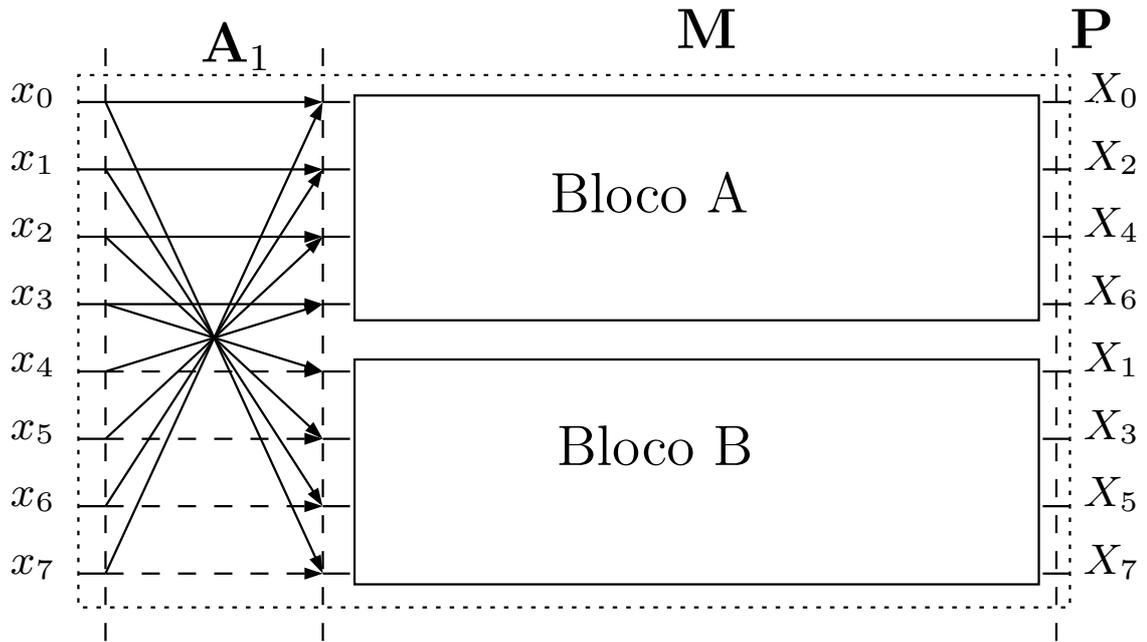


Figura 5 – Diagrama de fluxo de sinal geral para T_6 , T_8 e T_{18} .

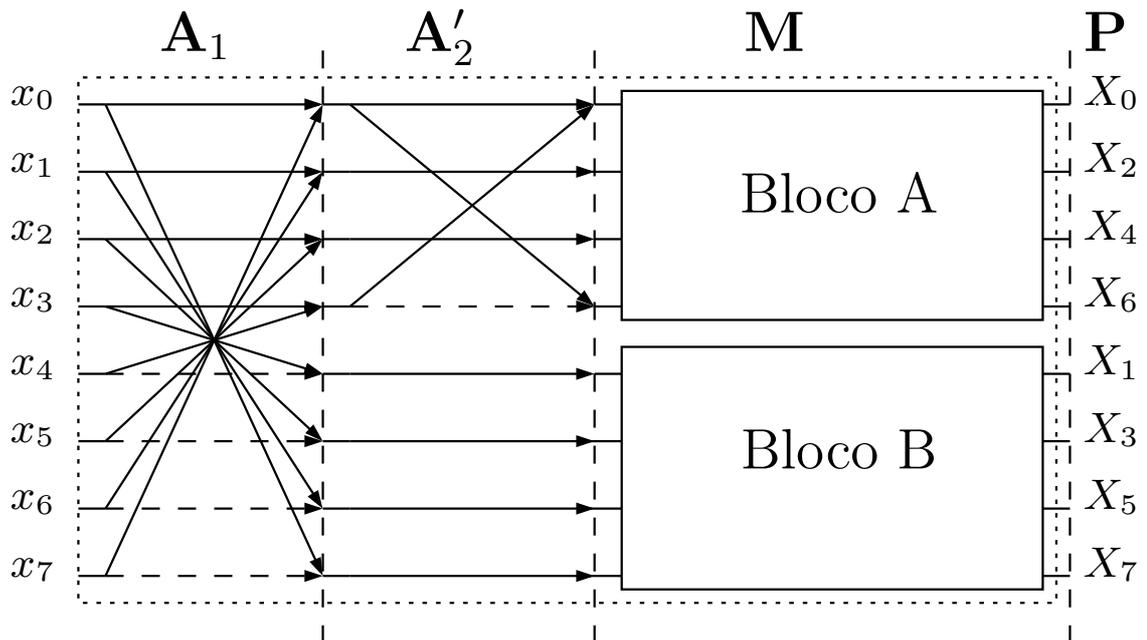


Figura 6 – Diagrama de fluxo de sinal geral para T_{21} e T_{22} .

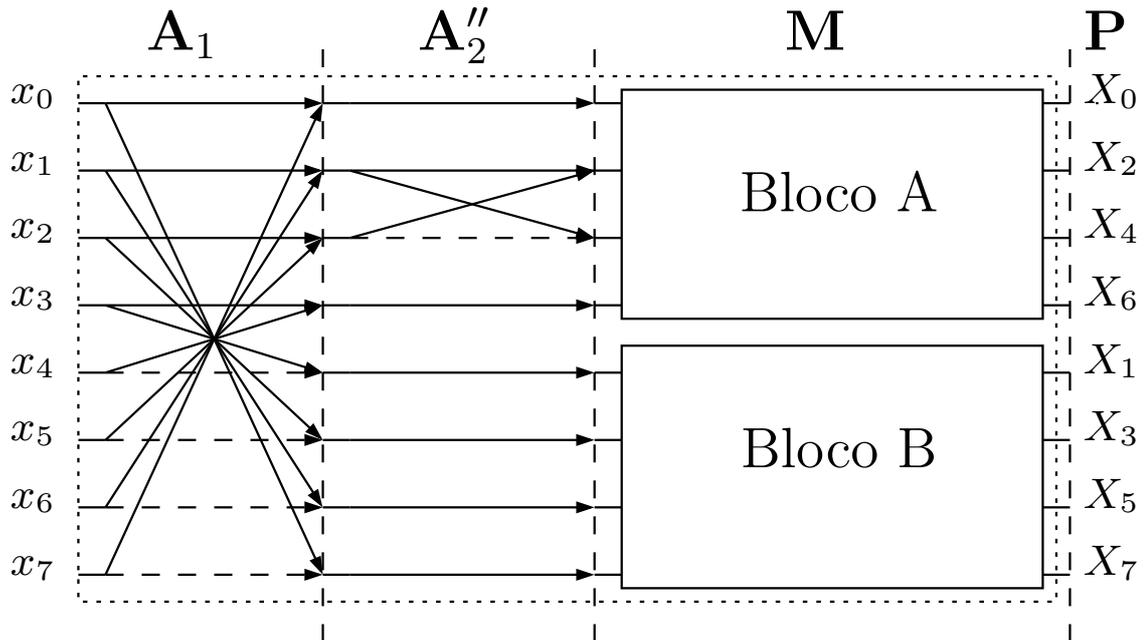
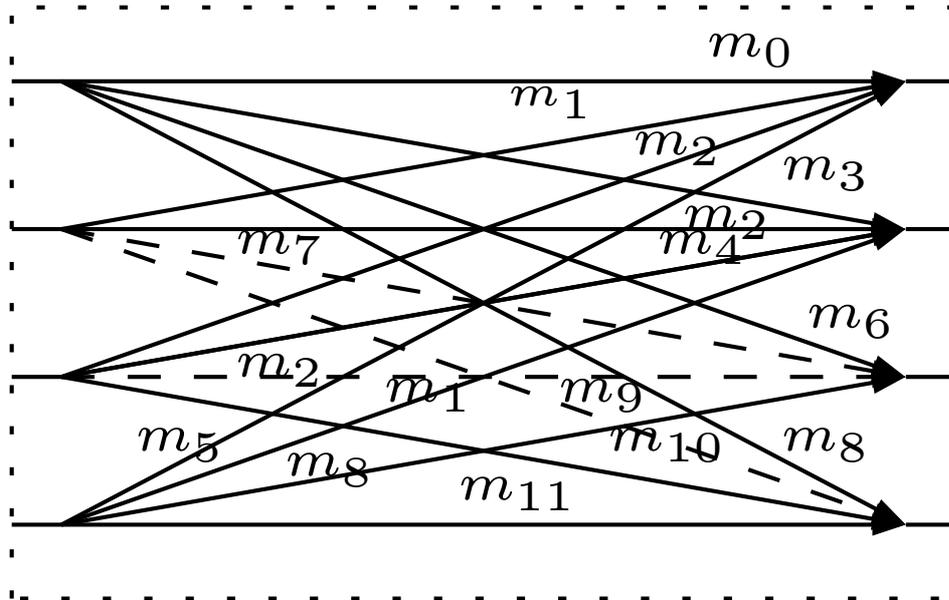


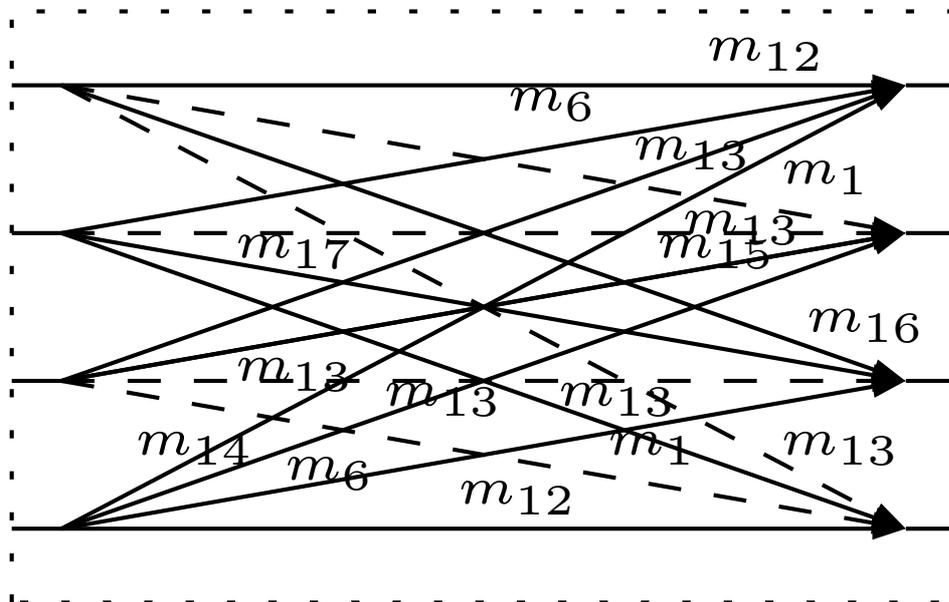
Figura 7 – Diagrama de fluxo de sinal geral para T_{23} .

4.2 COMPLEXIDADE COMPUTACIONAL

A complexidade computacional de transformadas discretas pode depender de muitos fatores diferentes, desde aspectos puramente matemáticos até detalhes de sua realização física em *hardware* ou implementação em *software*. No livro “*The DFT: An Owner’s Manual for the Discrete Fourier Transform*”, Briggs e Henson [91] citam diversos fatores que podem determinar a complexidade computacional de algoritmos rápidos. Alguns desses fatores são: (i) a **simetria** do sinal de entrada; (ii) o cômputo de **transformações múltiplas** que são calculadas mais de uma vez em casos multidimensionais; (iv) **operações adicionais** que são muitas vezes negligenciadas e afetam o cômputo da transformada. Como exemplo destas operações podemos citar matrizes de armazenamento adicionais, permutações e movimentação de dados; (v) **estratégias híbridas** que consistem em computar transformadas de certo comprimento (N) a partir de transformadas de comprimentos menores; (vi) o desempenho de transformadas rápidas pode depender também de problemas de *hardware* e **arquitetura**; (vii) e também o **software** pois seu custo se torna imensurável dado que nenhum pacote único estará em conformidade com todos os aplicativos e ambientes de computação. Combinar estes fatores para a redução do custo computacional pode ser considerada uma tarefa intratável, pois há muitos fatores de difícil quantificação ou controle. Porém, há um fator que não está



(a) Bloco A



(b) Bloco B

Figura 8 – Blocos A e B dos diagramas de fluxo de sinal.

listado acima e que podemos controlar, que é a **complexidade aritmética** da transformada. A complexidade aritmética da transformada é composta pelas complexidades multiplicativa, aditiva e outras operações, como deslocamentos de *bits*. Nosso interesse é entender e controlar a quantidade de operações aritméticas necessárias para avaliar uma determinada transformada [92]. A complexidade aritmética pode ser definida como uma função que relaciona as complexidades aditivas, multiplicativas e deslocamento de *bits*. Dentre estas complexidades, a complexidade multiplicativa é a que possui as operações mais custosas [13, 27]. Sendo assim, podemos aproximar a complexidade aritmética pela complexidade multiplicativa:

$$\text{complexidade aritmética} \approx \text{complexidade multiplicativa.} \quad (4.13)$$

As operações multiplicativas podem ser definidas de duas maneiras diferentes: multiplicações triviais e não triviais. As multiplicações triviais são ditas assim, pois elas podem ser realizadas sem o uso de um algoritmo de multiplicação numérica, como por exemplo, multiplicações pelos elementos do conjunto $\{0, \pm 1/2, \pm 1, \pm 2\}$. Multiplicações por potência de dois podem ser facilmente realizadas por simples deslocamentos de *bit* em uma arquitetura computacional binária. Portanto, apenas multiplicações não triviais incrementam a complexidade multiplicativa [13].

4.2.1 Complexidade aritmética dos algoritmos rápidos propostos

A complexidade computacional das transformadas propostas pode ser calculada pela complexidade aritmética, dada pelo número de multiplicações, adições e deslocamentos de *bits* necessários para sua implementação. O número de multiplicações é dado pelos elementos m_k da matriz \mathbf{M} (4.10). No entanto, como a busca das matrizes foi construída de maneira a termos transformadas de baixa complexidade, com $m_k \in \{0, \pm 1, \pm 2, \pm 3\}$, nesse caso são necessárias apenas operações de adição e deslocamento de *bit*.

Nesse sentido, o número de adições, $\mathcal{A}(m)$, e de deslocamento de *bits*, $\mathcal{S}(m)$, dos algoritmos rápidos propostos podem ser representados por:

$$\mathcal{A}(m) = 32 + \sum_{i=0}^{17} \omega_i \mathcal{I}_{\{3\}}(m_i) - \sum_{i=0}^{17} \omega_i \mathcal{I}_{\{0\}}(m_i), \quad (4.14)$$

$$\mathcal{S}(m) = \sum_{i=0}^{17} \omega_i \gamma(m_i), \quad (4.15)$$

em que $\omega = [1, 2, 3, 1, 1, 3, 1, 2, 1, 1, 1, 2, 5, 1, 1, 1, 1, 1]^\top$, $\mathcal{I}_{\{i\}}$ é a função indicadora e $\gamma(m_i)$ é definida por:

$$\gamma(m_i) = \begin{cases} 1, & \text{se } |m_i| = 2 \text{ ou } |m_i| = 3, \\ 0, & \text{caso contrário.} \end{cases} \quad (4.16)$$

A Tabela 18 apresenta as complexidades aritméticas dos algoritmos rápidos das transformadas propostas. São apresentadas também as complexidades aritméticas da KLT, da DCT com algoritmo rápido proposto por Loeffler *et al.* [25], da SDCT e da RDCT. Podemos observar que as transformadas que não são aproximações exigem multiplicações, já as transformadas aproximadas têm complexidade multiplicativa nula. A menos de operações de deslocamento de *bit*, as transformadas propostas competem muito bem em relação à complexidade aritmética com as transformadas aproximadas para a DCT da literatura, principalmente \mathbf{T}_{18} e \mathbf{T}_{23} . As quantidades de adições e deslocamento de *bit* das transformadas \mathbf{T}_8 , \mathbf{T}_{21} e \mathbf{T}_{22} têm um adicional pelo fato de que estas transformadas possuem elementos ± 3 em suas matrizes.

Tabela 18 – Comparação da complexidade aritmética das transformadas de comprimento 8

Transformada	Adição	Multiplicação	Deslocamento de <i>bit</i>
KLT	56	64	0
DCT [25]	29	11	0
<hr style="border-top: 1px dashed black;"/>			
SDCT [31]	24	0	0
RDCT [30]	22	0	0
\mathbf{T}_6	24	0	0
\mathbf{T}_8	30 + 18	0	6 + 18
\mathbf{T}_{18}	26	0	13
\mathbf{T}_{21}	28 + 10	0	12 + 10
\mathbf{T}_{22}	27 + 11	0	11 + 11
\mathbf{T}_{23}	26	0	12

5 RESULTADOS NUMÉRICOS

Para avaliar as transformadas propostas em compressão de imagens foram utilizadas imagens retiradas do Banco de Imagens Público “*The USC-SIPI Image Database*” [93]. Todas as imagens armazenadas neste banco encontram-se em tamanhos 256×256 e 512×512 e, portanto, vão ao encontro da condição abordada neste trabalho, ou seja, a dimensão das imagens ser $N \times N$.

A metodologia aplicada para comprimir as imagens, bem como as métricas utilizadas para avaliar a qualidade das imagens comprimidas, são apresentadas nas seções a seguir.

5.1 COMPRESSÃO DE IMAGENS

Podemos avaliar o desempenho das transformadas propostas quando aplicadas na compressão de imagens, assim como em [28, 29, 32]. Por simplicidade, mas sem perda de generalidade, serão consideradas imagens de 8 *bits* em escala de cinza. Uma imagem digital de 8 *bits* pode ser definida como uma matriz, em que cada pixel representa um nível de escala de cinza com valores de zero à $255 = 2^8 - 1$.

A metodologia utilizada para a compressão de imagens é apresentada nos passos seguintes [6]:

- (i) a imagem é dividida em blocos \mathbf{A}_j de ordem $N \times N$;
- (ii) é aplicada a transformação bidimensional \mathbf{T} em cada bloco, por $\mathbf{B}_j = \mathbf{T} \cdot \mathbf{A}_j \cdot \mathbf{T}^\top$;
- (iii) empregando a sequência zig-zag padrão [6], conforme Figura 9, retém-se apenas os r coeficientes iniciais em cada bloco \mathbf{B}_j e os $N^2 - r$ coeficientes restantes são anulados, obtendo os blocos truncados $\bar{\mathbf{B}}_j$ (nesta etapa acontece a compressão);
- (iv) aplica-se a transformada bidimensional inversa, dada por $\bar{\mathbf{A}}_j = \mathbf{T}^{-1} \cdot \bar{\mathbf{B}}_j \cdot (\mathbf{T}^{-1})^\top$;
- (v) recompõem-se os blocos comprimidos $\bar{\mathbf{A}}_j$ no lugar dos blocos originais \mathbf{A}_j da imagem. A imagem final comprimida $\bar{\mathbf{A}}$ pode então ser comparada com a imagem original \mathbf{A} para avaliar a perda de qualidade imposta pela compressão.

A Figura 9 apresenta a sequência zig-zag padrão utilizada na compressão. Esse padrão zig-zag pode ser otimizado para as transformadas propostas. Porém, neste caso, mantemos a sequência padrão para que possamos fazer uma comparação justa entre as

transformadas. A Figura 10 ilustra o diagrama da operação de compressão de imagem utilizado neste trabalho.

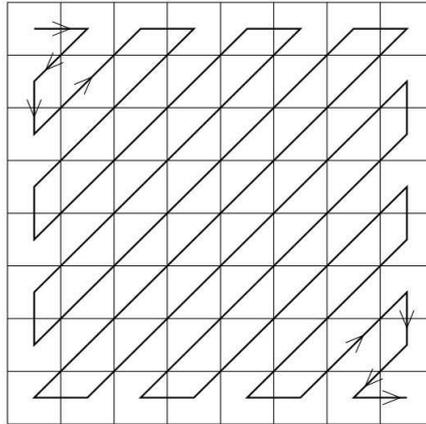


Figura 9 – Padrão zig-zag.

O número de coeficientes r retidos na etapa de quantização indicam a taxa de compressão (TC) da imagem. A taxa de compressão é calculada por:

$$TC = 1 - \frac{r}{N^2}. \quad (5.1)$$

5.2 MEDIDAS DE QUALIDADE DE IMAGEM

No processo de compressão a imagem recuperada não é a mesma imagem original. Para avaliar a qualidade das imagens comprimidas, algumas figuras de méritos usuais podem ser utilizadas, tais como: o erro quadrático médio (EQM) [14], a relação sinal-ruído de pico (PSNR) [94] e o índice de similaridade estrutural médio (MSSIM) [95].

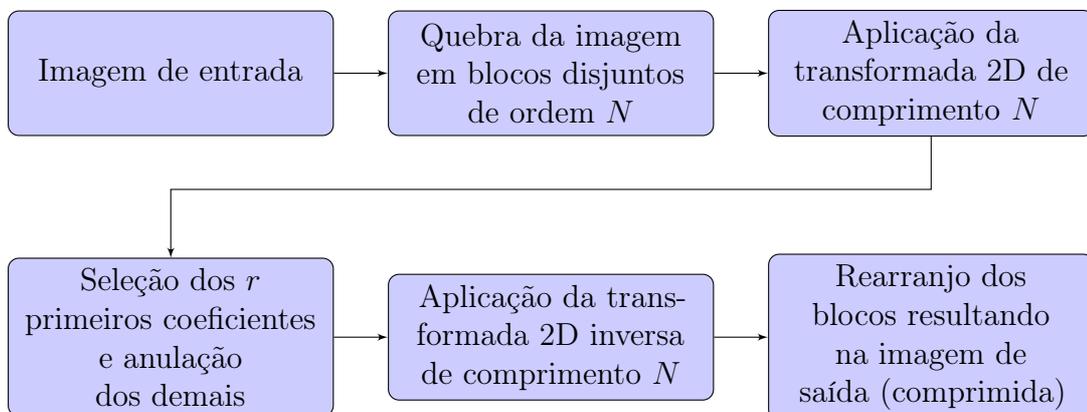


Figura 10 – Diagrama de blocos da operação de compressão de imagem.

5.2.1 Erro quadrático médio

O erro quadrático médio entre a imagem original \mathbf{A} e a imagem comprimida $\bar{\mathbf{A}}$ é definido por [14]:

$$\text{EQM} = \frac{\sum_{i=1}^N \sum_{j=1}^N (a_{i,j} - \bar{a}_{i,j})^2}{N^2}, \quad (5.2)$$

em que $a_{i,j}$ são os elementos (*pixels*) da imagem original, $\bar{a}_{i,j}$ os elementos da imagem reconstituída e N é a dimensão de ambas imagens.

5.2.2 Relação sinal-ruído de pico

A relação sinal-ruído de pico (PSNR) é definida em *dB* e é dada por [94]:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{EQM}} \right), \quad (5.3)$$

em que EQM é a Equação (5.2) e MAX é o valor máximo dos elementos de \mathbf{A} . Como são consideradas imagens em escala de cinza em 8 bits, $\text{MAX} = 255$.

5.2.3 Índice de similaridade estrutural médio

Uma outra medida que pode ser utilizada para avaliação da qualidade da imagem é o índice de similaridade estrutural médio (MSSIM) [95]. O MSSIM considera o funcionamento da visão humana para determinar sua formalização matemática e é dado por:

$$\text{MSSIM}(\mathbf{A}, \bar{\mathbf{A}}) = \frac{1}{M} \sum_{j=1}^M \text{SSIM}(\mathbf{A}_j, \bar{\mathbf{A}}_j), \quad (5.4)$$

em que \mathbf{A}_j e $\bar{\mathbf{A}}_j$, são respectivamente, os j -ésimos blocos de \mathbf{A} e $\bar{\mathbf{A}}$ e M é o total de blocos. O cálculo do j -ésimo bloco, SSIM, é dado por:

$$\text{SSIM}(\mathbf{A}_j, \bar{\mathbf{A}}_j) = \frac{[2 \cdot \text{mean}(\mathbf{A}_j) \cdot \text{mean}(\bar{\mathbf{A}}_j) + C_1][2 \cdot \text{cov}(\mathbf{A}_j, \bar{\mathbf{A}}_j) + C_2]}{[\text{mean}(\mathbf{A}_j)^2 + \text{mean}(\bar{\mathbf{A}}_j)^2 + C_1][\text{var}(\mathbf{A}_j) + \text{var}(\bar{\mathbf{A}}_j) + C_2]}, \quad (5.5)$$

em que \mathbf{A}_j e $\bar{\mathbf{A}}_j$ são blocos de \mathbf{A} e $\bar{\mathbf{A}}$ respectivamente, $C_1 = 6.5025$ e $C_2 = 58.5225$ [95], $\text{mean}(\mathbf{A}_j)$ é a média aritmética de \mathbf{A}_j , $\text{mean}(\bar{\mathbf{A}}_j)$ é a média aritmética de $\bar{\mathbf{A}}_j$, $\text{var}(\mathbf{A}_j)$ é a variância de \mathbf{A}_j , $\text{var}(\bar{\mathbf{A}}_j)$ é a variância de $\bar{\mathbf{A}}_j$ e $\text{cov}(\mathbf{A}_j, \bar{\mathbf{A}}_j)$ é a covariância entre \mathbf{A}_j e $\bar{\mathbf{A}}_j$.

Para melhor visualização dos resultados, foi considerado o erro absoluto percentual (APE) relativo à KLT exata (para diferentes valores de ρ) das medidas de PSNR, MSSIM e MSE. Por exemplo, o APE, considerando o PSNR, é calculado pela seguinte expressão:

$$\text{APE(PSNR)} = \left| \frac{\text{PSNR}_{\mathbf{K}(\rho)} - \text{PSNR}_{\widehat{\mathbf{K}}}}{\text{PSNR}_{\mathbf{K}(\rho)}} \right|, \quad (5.6)$$

em que $\text{PSNR}_{\mathbf{K}(\rho)}$ e $\text{PSNR}_{\widehat{\mathbf{K}}}$ são os valores de PSNR considerando a KLT exata, dado um valor de ρ , e uma aproximação $\widehat{\mathbf{K}}$, respectivamente. Os valores de APE(MSSIM) e APE(MSE) são calculados de forma similar.

5.2.4 Resultados

Para a avaliação do desempenho da transformadas propostas via compressão de imagens, utilizaram-se imagens retiradas do Banco de Imagens Público [93]. Nesta subseção apresentaremos dois tipos de análises: (i) uma quantitativa, variando o valor de r para a compressão, considerando os valores de MSSIM e PSNR da imagem comprimida e (ii) uma outra qualitativa, baseada em imagens comprimidas com aproximadamente 85% de compressão. Em ambas as análises, as imagens serão comprimidas utilizando as transformadas propostas e também utilizando a DCT e a KLT (para $\rho = 0.95$). No contexto de processamento de imagens, os dados dos pixels adjacentes são altamente correlacionados [47], usualmente $\rho = 0.95$, e por este motivo utilizamos a $\mathbf{K}^{(0.95)}$ comparativamente nos experimentos de compressão.

Para a análise quantitativa, foi utilizado um conjunto de 45 imagens obtidas de [93], de dimensão 512×512 de 8-bits em escala de cinza, assim como em [32]. Algumas das imagens retiradas do banco de imagens são coloridas, neste caso, consideramos apenas a luminância das mesmas. Os valores médios de PSNR, MSSIM de todas as 45 imagens foram considerados. As 45 imagens utilizadas nesta etapa estão no Apêndice B.

A Figura 11 apresenta as imagens *Lena*, *Boat* e *Grass*, retiradas deste banco de imagens, que foram utilizadas para a avaliação qualitativa.

As Figuras 12, 14, 16, apresentam as imagens comprimidas utilizando as transformadas propostas, a DCT e a KLT (para $\rho = 0.95$). As Figuras 13, 15 e 17 exibem a diferença entre as imagens comprimidas e as imagens originais. As transformadas $\widehat{\mathbf{K}}_6$ e $\widehat{\mathbf{K}}_8$ são transformadas propostas considerando valores de ρ baixo. Como os *pixels* de uma imagem natural são altamente correlacionados [47] a compressão da imagem utilizando

(a) *Lena*(b) *Boat*(c) *Grass*

Figura 11 – Imagens originais.

estas transformadas não apresenta bons resultados, como esperado. Porém, considerando as demais transformadas propostas podemos notar que, qualitativamente, não há uma apreciável diferença entre as imagens comprimidas.

A Tabela 19 apresenta os valores de EQM, PSNR e MSSIM das imagens comprimidas segundo cada transformada. Nesta tabela, adicionamos as medidas considerando também a SDCT e RDCT, a fim de comparação. Além das medidas para a KLT, destacamos o desempenho das três melhores transformadas segundo cada métrica, para as três imagens comprimidas diferentes. As transformadas propostas tiveram desempenho melhor que a SDCT e a RDCT (que também são transformadas de baixa complexidade) para todos os casos. Considerando a imagem *Grass*, as transformadas \hat{K}_{21} e \hat{K}_{22} apresentaram melhores medidas do que a própria DCT em termos de PSNR. Estes resultados são muito bons considerando que as transformadas propostas são transformadas de baixa complexidade.

Tabela 19 – Medidas de qualidade das imagens

Imagem	<i>Lena</i>			<i>Boat</i>			<i>Grass</i>		
Transformada	EQM	PSNR	MSSIM	EQM	PSNR	MSSIM	EQM	PSNR	MSSIM
KLT	41.0898	31.9935	0.9019	83.0899	28.9353	0.8536	659.543	19.9384	0.7864
DCT	40.2664	32.0814	0.9136	82.3894	28.9721	0.8602	666.4695	19.893	0.7839
SDCT	117.1966	27.4416	0.8577	180.3947	25.5686	0.7975	1292.029	17.0181	0.6777
RDCT	58.7272	30.4424	0.8932	106.3967	27.8615	0.8372	789.5017	19.1573	0.7426
\hat{K}_6	5505.235	10.7230	0.1086	5925.529	10.4035	0.1335	6122.267	10.2617	0.3440
\hat{K}_8	982.5026	18.2075	0.2698	1088.367	17.7631	0.3082	1568.127	16.1770	0.6216
\hat{K}_{18}	57.6005	30.5265	0.8093	104.6983	27.9314	0.7884	707.104	19.6360	0.7797
\hat{K}_{21}	42.6135	31.8353	0.8942	85.1470	28.8291	0.8477	656.7466	19.9568	0.7884
\hat{K}_{22}	43.5124	31.7447	0.8934	86.5637	28.7574	0.8470	662.1343	19.9213	0.7874
\hat{K}_{23}	44.055	31.6908	0.9091	88.8640	28.6436	0.8555	713.8527	19.59472	0.7776

Para análises mais gerais relativas ao desempenho das transformadas propostas em compressão de imagens, extendemos este experimento para as 45 imagens apresenta-



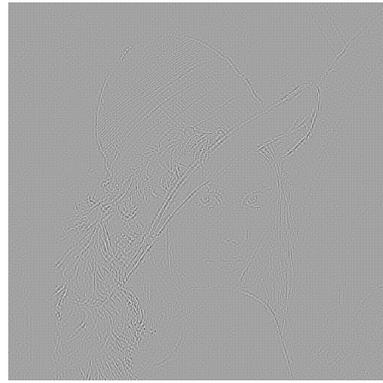
(a) KLT



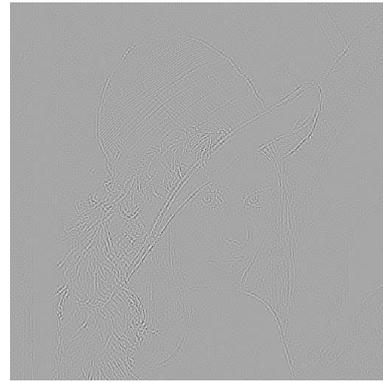
(b) DCT

(c) \hat{K}_6 (d) \hat{K}_8 (e) \hat{K}_{18} (f) \hat{K}_{21} (g) \hat{K}_{22} (h) \hat{K}_{23}

Figura 12 – Imagem *Lena* comprimida utilizando a DCT, a KLT e as transformadas propostas, com $r = 10$.



(a) KLT



(b) DCT

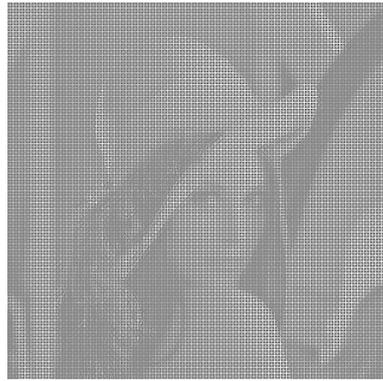
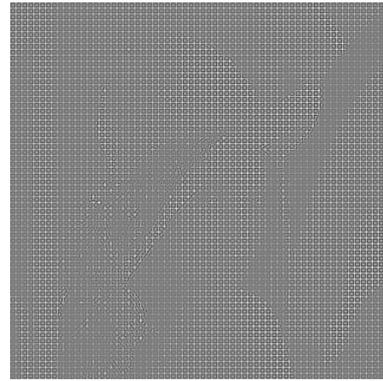
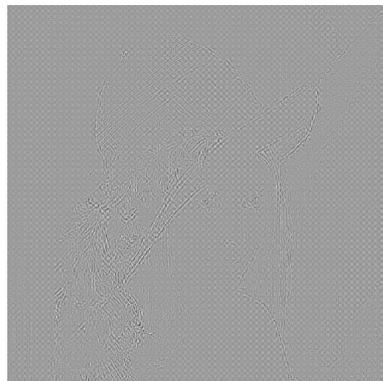
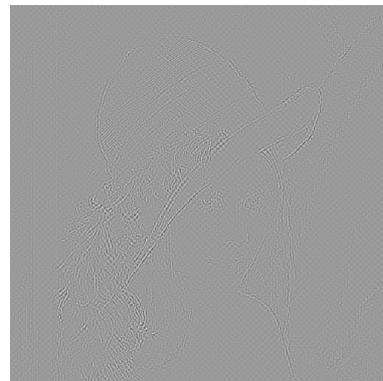
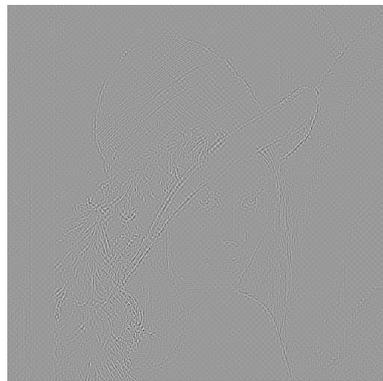
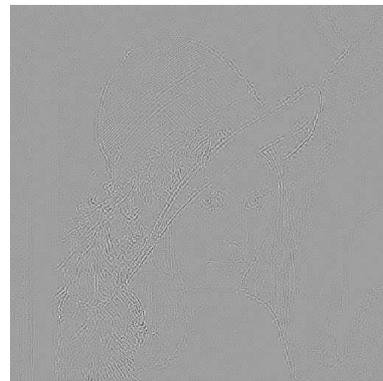
(c) \hat{K}_6 (d) \hat{K}_8 (e) \hat{K}_{18} (f) \hat{K}_{21} (g) \hat{K}_{22} (h) \hat{K}_{23}

Figura 13 – Diferença entre a Imagem *Lena* comprimida e a original.



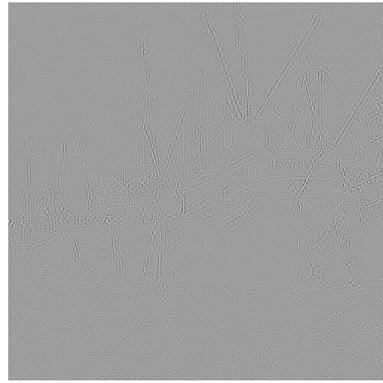
(a) KLT



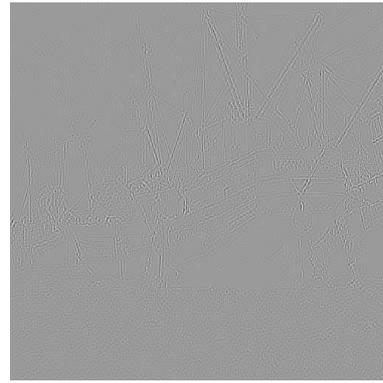
(b) DCT

(c) \hat{K}_6 (d) \hat{K}_8 (e) \hat{K}_{18} (f) \hat{K}_{21} (g) \hat{K}_{22} (h) \hat{K}_{23}

Figura 14 – Imagem *Boat* comprimida utilizando a DCT, a KLT e as transformadas propostas, com $r = 10$.



(a) KLT



(b) DCT

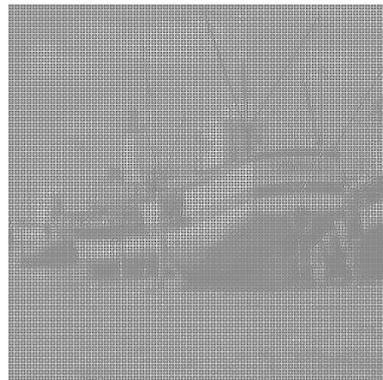
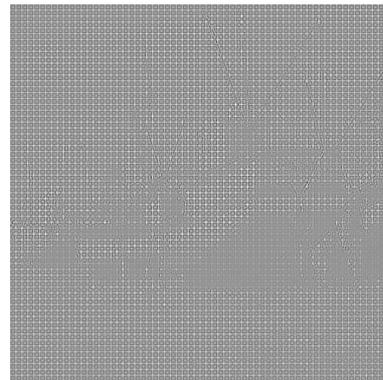
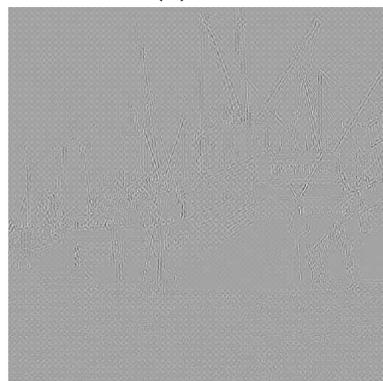
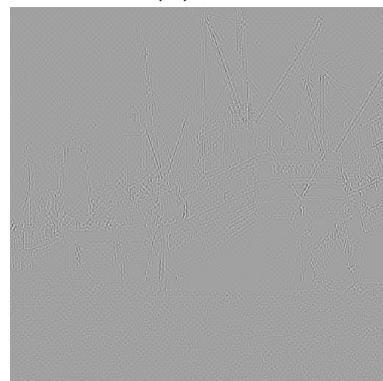
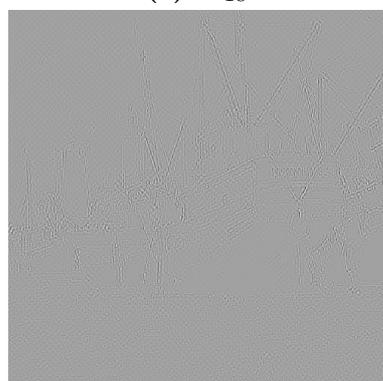
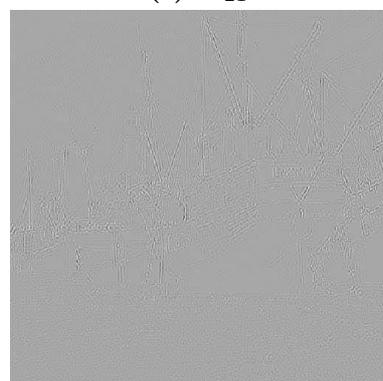
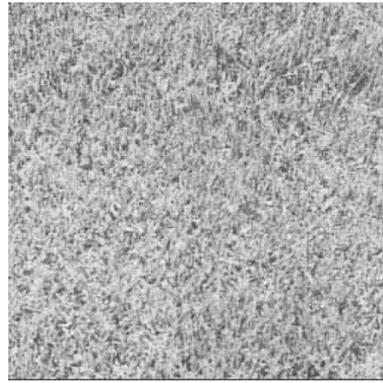
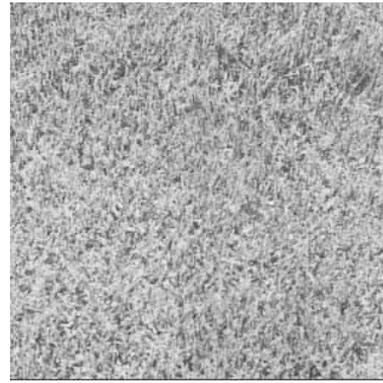
(c) \hat{K}_6 (d) \hat{K}_8 (e) \hat{K}_{18} (f) \hat{K}_{21} (g) \hat{K}_{22} (h) \hat{K}_{23}

Figura 15 – Diferença entre a Imagem *Boat* comprimida e a original.



(a) KLT



(b) DCT

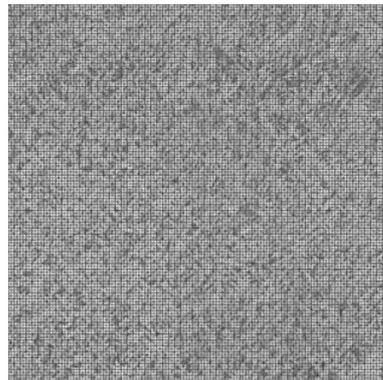
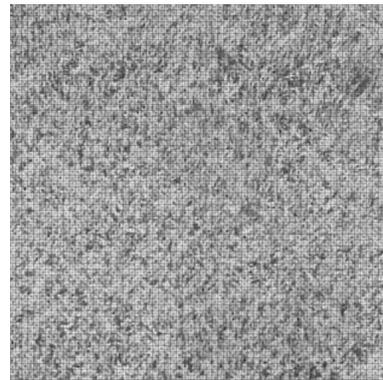
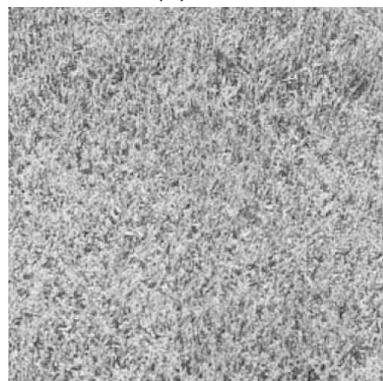
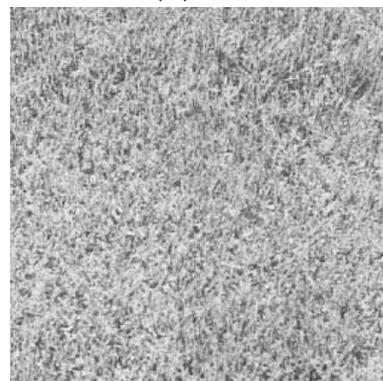
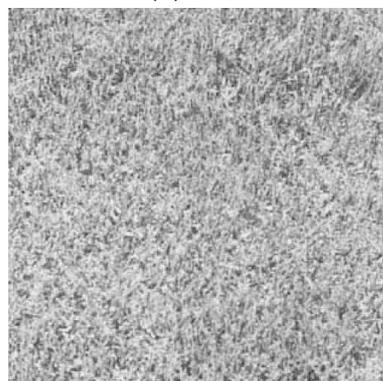
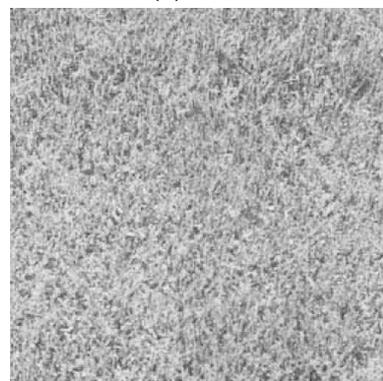
(c) \hat{K}_6 (d) \hat{K}_8 (e) \hat{K}_{18} (f) \hat{K}_{21} (g) \hat{K}_{22} (h) \hat{K}_{23}

Figura 16 – Imagem *Grass* comprimida utilizando a DCT, a KLT e as transformadas propostas, com $r = 10$.

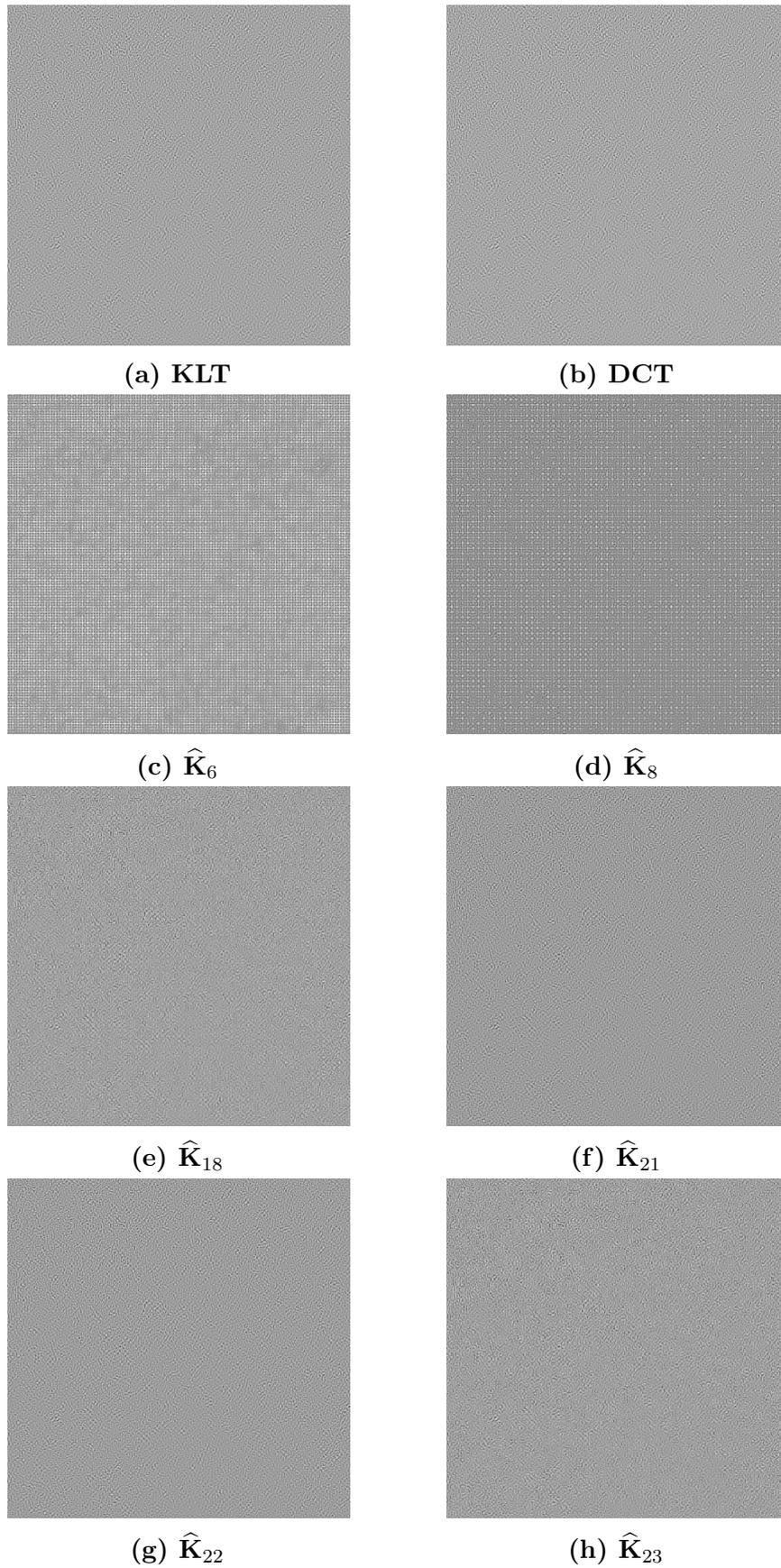


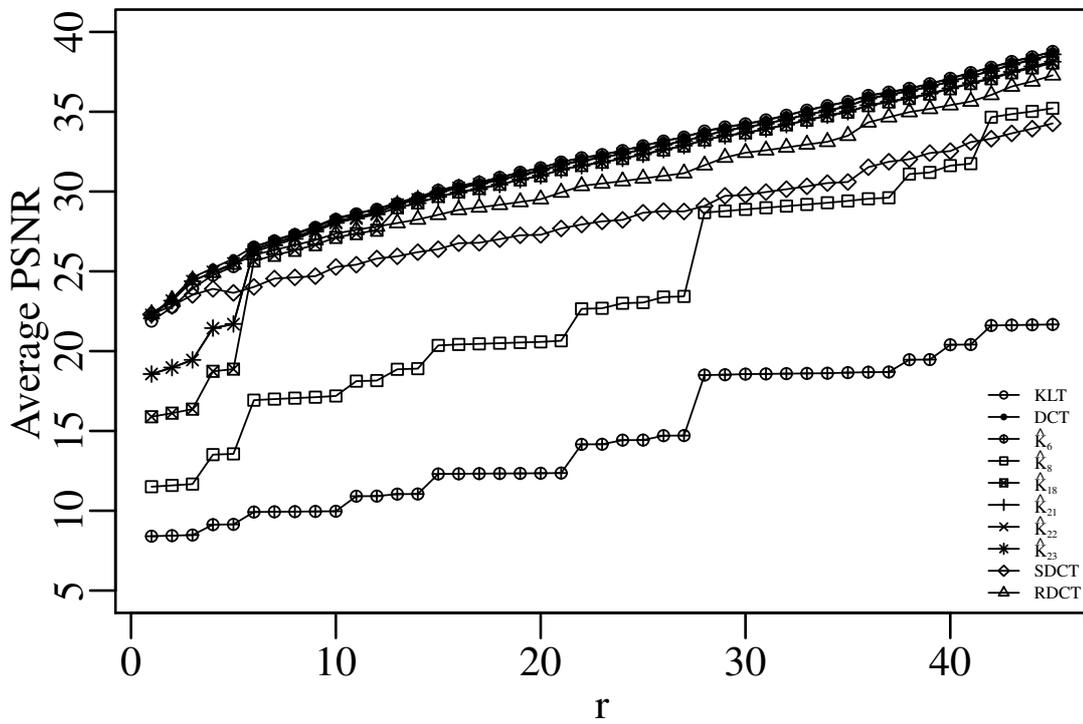
Figura 17 – Diferença entre a Imagem *Grass* comprimida e a original.

das no Apêndice B considerando diferentes taxas de compressão ($1 \leq r \leq 45$). Foram calculadas as medidas de PSNR e MSSIM para cada imagem. As Figuras 18a e 19a apresentam os gráficos dos valores médios destas medidas. Nas Figuras 18 e 19 e também são exibidos os gráficos considerando o erro absoluto percentual (APE) de cada medida. Novamente podemos perceber que algumas das transformadas propostas têm um comportamento muito semelhante as transformadas de complexidade multiplicativa não nula, KLT e DCT, principalmente a medida que aumentamos o número de coeficientes retidos r . Já para $r = 10$, o que implica em uma taxa de compressão de 85%, as transformadas $\widehat{\mathbf{K}}_{21}$, $\widehat{\mathbf{K}}_{22}$ e $\widehat{\mathbf{K}}_{23}$ apresentam um desempenho muito parecido ao da DCT e da KLT em termos de PSNR e MSSIM. A Tabela 20 apresenta os valores de PSNR e MSSIM médios das transformadas considerando esta taxa de compressão.

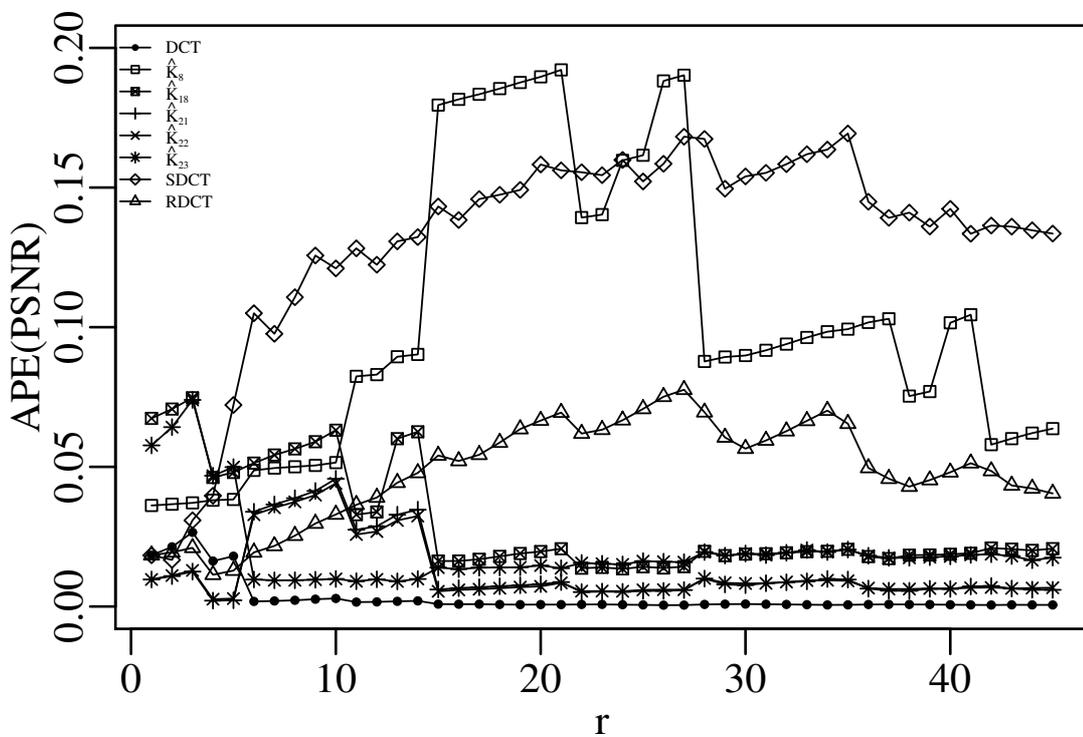
Tabela 20 – Valores do PSNR e MSSIM médios, considerando $r = 10$

Transformada	PSNR	MSSIM
KLT	28.2605	0.8288
DCT	28.3371	0.8361
SDCT	25.2740	0.7642
RDCT	27.3699	0.8092
$\widehat{\mathbf{K}}_6$	9.9677	0.1221
$\widehat{\mathbf{K}}_8$	17.1911	0.3011
$\widehat{\mathbf{K}}_{18}$	27.1251	0.7634
$\widehat{\mathbf{K}}_{21}$	28.1392	0.8238
$\widehat{\mathbf{K}}_{22}$	28.1003	0.8232
$\widehat{\mathbf{K}}_{23}$	28.0347	0.8308

Podemos destacar, após estas análises, a transformada $\widehat{\mathbf{K}}_{23}$ como sendo a melhor entre as transformadas propostas. Ela possui a menor complexidade aritmética, como apresentado na Tabela 18, e tem um desempenho muito bom quando aplicada em compressão de imagens – ganhando da SDCT e da RDCT.

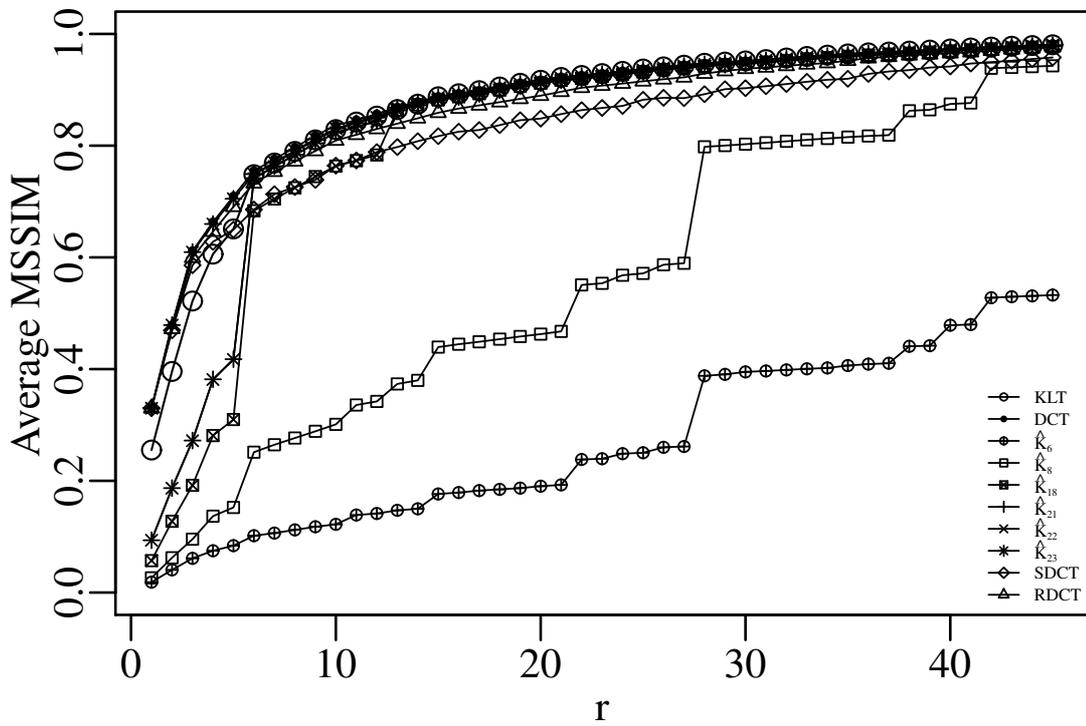


(a) PSNR

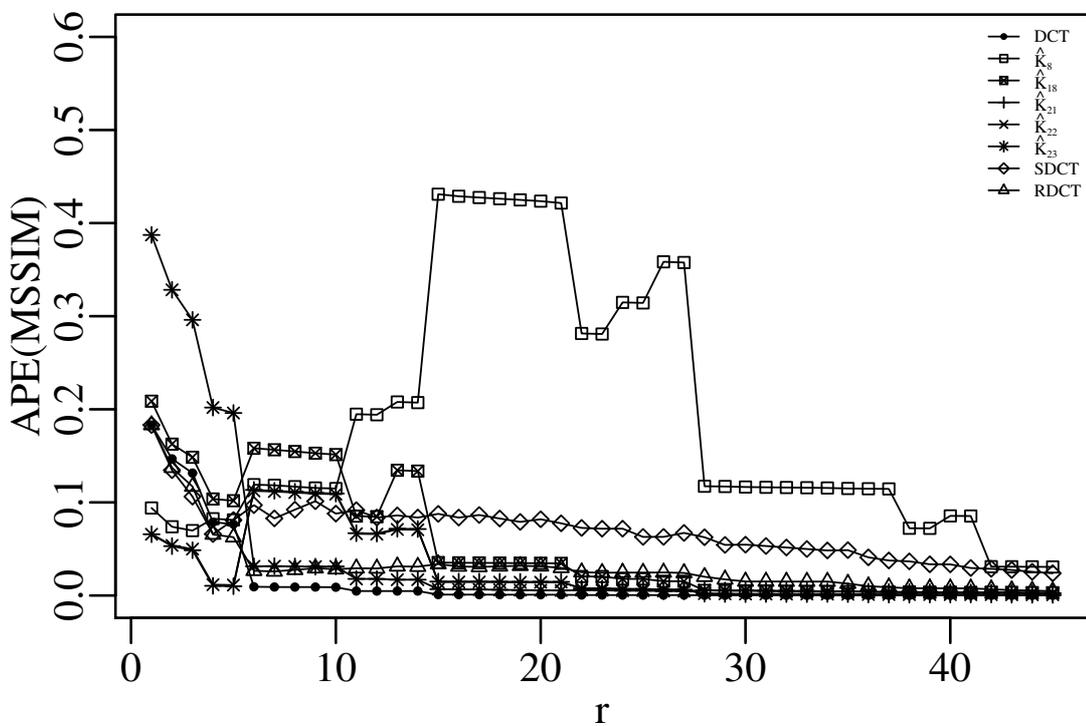


(b) APE(PSNR)

Figura 18 – PSNR médio das transformadas ótimas propostas com relação a compressão de imagem variando o valor de r .



(a) MSSIM



(b) APE(MSSIM)

Figura 19 – MSSIM médio das transformadas ótimas propostas com relação a compressão de imagem variando o valor de r .

6 CONCLUSÕES

No presente trabalho, foram derivadas novas aproximações de baixa complexidade para a transformada de Karhunen-Loève (KLT) de comprimento $N = 8$. A obtenção destas transformadas foi baseada em transformadas clássicas na literatura que são aproximações para a DCT, como a SDCT [31], a RDCT [32] e a classe de aproximações baseadas em funções inteiras proposta em [28]. O interesse em propor novas transformadas de baixa complexidade se dá principalmente pelo fato de que estas transformadas podem ser implementadas em *hardware* e são atrativas por requerer baixo custo e baixa demanda de potência, como nos casos de transmissão e processamento de imagem e vídeo em tempo real e aplicações de cômputo portátil.

Nos Capítulos 1 e 2, foram apresentadas uma introdução e o formalismo teórico da KLT e de outras transformadas usualmente utilizadas no contexto de processamento de sinais e imagens. Foi apresentada também a relação da DCT com a KLT, e algumas de suas aproximações de baixa complexidade clássicas da literatura, como a SDCT e a RDCT, entre outras. Também são evidenciadas neste capítulo figuras de mérito utilizadas para avaliação de transformadas aproximadas.

No Capítulo 3, são apresentadas as transformadas de baixa complexidade propostas bem como a metodologia utilizada para a derivação destas. É neste capítulo que se encontram as principais contribuições deste trabalho. Baseadas em diferentes procedimentos, foram encontradas 26 aproximações para a KLT, algumas delas são aproximações da DCT já conhecidas na literatura. Dentre estas transformadas, evidenciamos seis delas, que apresentamos como as transformadas ótimas segundo as figuras de mérito consideradas para avaliação. As transformadas ótimas propostas estão separadas em dois grupos: transformadas obtidas considerando um sinal em que seus elementos sejam fracamente correlacionados (ρ baixo); e transformadas obtidas considerando um sinal cujos elementos sejam altamente correlacionados (ρ alto). Algoritmos rápidos para estas transformadas foram propostos e estão apresentados no Capítulo 4, juntamente com uma análise de suas complexidades aritméticas. Por fim, experimentos de compressão de imagens são apresentados no Capítulo 5, evidenciando o bom desempenho das transformadas propostas em comparação com a KLT exata, por exemplo.

A complexidade computacional das novas transformadas é consideravelmente baixa, independentemente do sinal de entrada e sendo livre de multiplicações. As transfor-

madras propostas mostraram-se boas aproximações para KLT nas medidas de qualidade elegidas. Ainda, no contexto de compressão de imagens, elas se mostram bastante úteis, evidenciando imagens adequadas a um custo computacional associado bastante baixo.

Para trabalhos futuros, deseja-se:

- Explorar a aplicabilidade da KLT para sinais com baixo valor de ρ .
- Expandir a busca de novas transformadas considerando diferentes comprimentos, $N = 4, 16$ e 32 , pela importância destes comprimentos em codificação em vídeo HEVC.
- Considerar mais funções, como as funções inteiras de arredondamento para o inteiro mais próximo, para a derivação de novas transformadas.
- Utilizar as transformadas para a codificação em vídeo HEVC [18].
- Utilizar as transformadas para aplicação em outro contexto além de compressão de imagem, como por exemplo aplicação de marcas d'água [96].

6.1 PRODUÇÃO ACADÊMICA

Baseadas nessa dissertação, foram realizadas as seguintes produções acadêmicas:

- **Depósito de Patente:** Radünz, A. P., Cintra, R. J., Bayer, F. M., Canterle, D. R., 2018. Aproximação para a Transformada de Karhunen-Loève com Baixo Custo Aritmético com Aplicações em Compressão de Imagens. **BR 10 2018 006344 8**
- **Trabalho em Conferência:** Radünz, A. P., Cintra, R. J., Bayer, F. M., 2018. Componentes Principais Sinalizadas com Aplicação em Processamento de Imagens. 23º Simpósio Nacional de Probabilidade de Estatística (SINAPE). São Pedro. Anais do 23º SINAPE, 2018.
- **Manuscrito:** Radünz, A. P., Cintra, R. J., Bayer, F. M., Silveira, T., L., T. *Signed Principal Components Applied to Image Processing*.

REFERÊNCIAS

- 1 GOLDSTON, D. Big data: Data wrangling. **Nature News**, Nature Publishing Group, v. 455, n. 7209, p. 15–15, 2008.
- 2 MERVIS, J. **Agencies rally to tackle big data**. [S.l.]: American Association for the Advancement of Science, 2012.
- 3 WU, X. *et al.* Data mining with big data. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 1, p. 97–107, 2014.
- 4 FLACH, P. **Machine learning: the art and science of algorithms that make sense of data**. [S.l.]: Cambridge University Press, 2012.
- 5 MOSTELLER, F. On some useful “inefficient” statistics. In: **Breakthroughs in Statistics**. [S.l.]: Springer, 1992. p. 209–228.
- 6 SALOMON, D. **Data compression: the complete reference**. [S.l.]: Springer Science & Business Media, 2004.
- 7 MOURA, J. What is signal processing?[president’s message]. **IEEE Signal Processing Magazine**, v. 26, n. 6, p. 6–6, 2009.
- 8 JOHNSON, R. A.; WICHERN, D. W. *et al.* **Applied multivariate statistical analysis**. [S.l.]: Prentice Hall Upper Saddle River, NJ, 2002. v. 5.
- 9 JOLLIFFE, I. Principal component analysis. In: **International Encyclopedia of Statistical Science**. [S.l.]: Springer, 2011. p. 1094–1096.
- 10 CHIPMAN, H. A.; GU, H. Interpretable dimension reduction. **Journal of Applied Statistics**, Taylor & Francis, v. 32, n. 9, p. 969–987, 2005.
- 11 DU, Q.; FOWLER, J. E. Hyperspectral image compression using JPEG2000 and principal component analysis. **IEEE Geoscience and Remote Sensing Letters**, v. 4, n. 2, p. 201–205, 2007.
- 12 DOUGHERTY, E. R. **Random processes for image and signal processing**. [S.l.]: SPIE Optical Engineering Press, 1999.
- 13 BLAHUT, R. E. **Fast algorithms for signal processing**. [S.l.]: Cambridge University Press, 2010.
- 14 BRITANAK, V.; YIP, P. C.; RAO, K. R. **Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations**. [S.l.]: Academic Press, 2010.
- 15 AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transform. **IEEE Transactions on Computers**, n. 1, p. 90–93, 1974.
- 16 WALLACE, G. K. The JPEG still picture compression standard. **IEEE Transactions on Consumer Electronics**, v. 38, n. 1, p. xviii–xxxiv, 1992.
- 17 PURI, A.; CHEN, X.; LUTHRA, A. Video coding using the H. 264/MPEG-4 AVC compression standard. **Signal Processing: Image Communication**, Elsevier, v. 19, n. 9, p. 793–849, 2004.

- 18 POURAZAD, M. T. *et al.* HEVC: The new gold standard for video compression: How does HEVC compare with H. 264/AVC? **IEEE Consumer Electronics Magazine**, v. 1, n. 3, p. 36–46, 2012.
- 19 CHEN, W.-H.; SMITH, C.; FRALICK, S. A fast computational algorithm for the discrete cosine transform. **IEEE Transactions on Communications**, v. 25, n. 9, p. 1004–1009, 1977.
- 20 HOU, H. A fast recursive algorithm for computing the discrete cosine transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 35, n. 10, p. 1455–1461, 1987.
- 21 VETTERLI, M.; NUSSBAUMER, H. J. *et al.* Simple FFT and DCT algorithms with reduced number of operations. **Signal Processing**, Elsevier, v. 6, n. 4, p. 267–278, 1984.
- 22 WANG, Z. Fast algorithms for the discrete w transform and for the discrete Fourier transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 32, n. 4, p. 803–816, 1984.
- 23 LEE, B. A new algorithm to compute the discrete cosine transform. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 32, n. 6, p. 1243–1245, 1984.
- 24 ARAI, Y.; AGUI, T.; NAKAJIMA, M. A fast DCT-SQ scheme for images. **IEICE TRANSACTIONS (1976-1990)**, The Institute of Electronics, Information and Communication Engineers, v. 71, n. 11, p. 1095–1097, 1988.
- 25 LOEFFLER, C.; LIGTENBERG, A.; MOSCHYTZ, G. S. Practical fast 1-D DCT algorithms with 11 multiplications. In: **IEEE. Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on.** [S.l.], 1989. p. 988–991.
- 26 FEIG, E.; WINOGRAD, S. Fast algorithms for the discrete cosine transform. **IEEE Transactions on Signal processing**, v. 40, n. 9, p. 2174–2193, 1992.
- 27 HEIDEMAN, M. T. Multiplicative complexity of linear and bilinear systems. In: **Multiplicative Complexity, Convolution, and the DFT.** [S.l.]: Springer, 1988. p. 5–26.
- 28 CINTRA, R. J.; BAYER, F. M.; TABLADA, C. Low-complexity 8-point DCT approximations based on integer functions. **Signal Processing**, Elsevier, v. 99, p. 201–214, 2014.
- 29 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. Low-complexity 8×8 transform for image compression. **Electronics Letters, IET**, v. 44, n. 21, p. 1249–1250, 2008.
- 30 BAYER, F. M.; CINTRA, R. J. Image compression via a fast DCT approximation. **IEEE Latin America Transactions**, v. 8, n. 6, p. 708–713, 2010.
- 31 HAWHEEL, T. I. A new square wave transform based on the DCT. **Signal Processing**, Elsevier, v. 81, n. 11, p. 2309–2319, 2001.
- 32 CINTRA, R. J.; BAYER, F. M. A DCT approximation for image compression. **IEEE Signal Processing Letters**, v. 18, n. 10, p. 579–582, 2011.

- 33 HNATIV, L. Integer cosine transforms: Methods to construct new order 8, 16 fast transforms and their application. **Cybernetics and Systems Analysis**, Springer, v. 50, n. 6, p. 913–929, 2014.
- 34 BAYER, F. M.; CINTRA, R. J. DCT-like transform for image compression requires 14 additions only. **Electronics Letters**, IET, v. 48, n. 15, p. 919–921, 2012.
- 35 MEYER-BAESE, U. Turbo LMS algorithm: supercharger meets adaptive filter. In: International Society for Optics and Photonics. **Independent Component Analyses, Wavelets, Unsupervised Smart Sensors, and Neural Networks IV**. [S.l.], 2006. v. 6247, p. 62470Y.
- 36 CLARKE, R. Relation between the Karhunen-Loève and cosine transforms. In: IET. **IEE Proceedings F-Communications, Radar and Signal Processing**. [S.l.], 1981. v. 128, n. 6, p. 359–360.
- 37 GANTMACHER, F. R. The theory of matrices, vol. 2. **Chelsea, New York**, v. 220, 1959.
- 38 LOÈVE, M. **Probability Theory, The University Series in Higher Mathematics**. [S.l.]: D. Van Nostrand Co., Inc., Princeton, NJ-Toronto-New York-London, 1960.
- 39 DEMIDOVICH, B.; MARON, I. Basics of numerical mathematics. **The Book**, p. 672, 1960.
- 40 KARHUNEN, K. Under lineare methoden in der wahr scheinlichkeitsrechnung. **Annales Academiae Scientiarum Fennicae Series A1: Mathematica Physica**, v. 47, 1947.
- 41 LOÈVE, M. Fonctions aléatoires de second ordre. **Processus Stochastique et Mouvement Brownien**, Gauthier-Villars, p. 366–420, 1948.
- 42 PEARSON, K. LIII. On lines and planes of closest fit to systems of points in space. **The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science**, Taylor & Francis, v. 2, n. 11, p. 559–572, 1901.
- 43 HOTELLING, H. Analysis of a complex of statistical variables into principal components. **Journal of Educational Psychology**, Warwick & York, v. 24, n. 6, p. 417, 1933.
- 44 RAO, K. R.; YIP, P. C. Discrete cosine transform, algorithm, advantage and applications. **New York: Academic**, 1990.
- 45 UNSER, M. On the approximation of the discrete Karhunen-Loève transform for stationary processes. **Signal Processing**, Elsevier, v. 7, n. 3, p. 231–249, 1984.
- 46 RAY, W.; DRIVER, R. Further decomposition of the Karhunen-Loève series representation of a stationary random process. **IEEE Transactions on Information Theory**, v. 16, n. 6, p. 663–668, 1970.
- 47 RAO, K. R.; YIP, P. C. **The transform and data compression handbook**. [S.l.]: CRC press, 2000. v. 1.

- 48 REED, I. S.; LAN, L.-S. A fast approximate Karhunen-Loève transform (AKLT) for data compression. **Journal of Visual Communication and Image Representation**, Elsevier, v. 5, n. 4, p. 304–316, 1994.
- 49 KOUASSI, R.; GOUTON, P.; PAINDAVOINE, M. Approximation of the Karhunen-Loève transformation and its application to colour images. **Signal Processing: Image Communication**, Elsevier, v. 16, n. 6, p. 541–551, 2001.
- 50 STRANG, G. **Linear Algebra and Its Applications**. Brooks Cole. [S.l.]: ISBN 0-15-551005-3, 1988.
- 51 ROMA, N.; SOUSA, L. Efficient hybrid DCT-domain algorithm for video spatial downscaling. **EURASIP Journal on Advances in Signal Processing**, Springer, v. 2007, n. 1, p. 057291, 2007.
- 52 UNION-TELECOMMUN, I. T. Generic coding of moving pictures and associated audio information-Part 2: Video. **Int. Standards Org./Int. Electrotech. Comm.(ISO/IEC) JTC 1, Rec. H. 262 and ISO/IEC 13 818-2 (MPEG-2 Video)**, 1994.
- 53 BOUGUEZEL, S.; AHMAD, M.; SWAMY, M. A multiplication-free transform for image compression. In: **2nd International Conference on Signals, Circuits and Systems (SCS)**. [S.l.: s.n.], 2008. p. 1–4.
- 54 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. A fast 8×8 transform for image compression. In: **IEEE. Microelectronics (ICM), 2009 International Conference on**. [S.l.], 2009. p. 74–77.
- 55 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. A novel transform for image compression. In: **IEEE. 2010 53rd IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)**. [S.l.], 2010. p. 509–512.
- 56 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. A low-complexity parametric transform for image compression. In: **IEEE. 2011 IEEE International Symposium on Circuits and Systems (ISCAS)**. [S.l.], 2011. p. 2145–2148.
- 57 BOUGUEZEL, S.; AHMAD, M. O.; SWAMY, M. Binary discrete cosine and hartley transforms. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 60, n. 4, p. 989–1002, 2013.
- 58 SILVEIRA, T. L. da *et al.* Multiplierless 16-point DCT approximation for low-complexity image and video coding. **Signal, Image and Video Processing**, Springer, v. 11, n. 2, p. 227–233, 2017.
- 59 HAWHEEL, R. T.; EL-KILANI, W. S.; RAMADAN, H. H. Fast approximate DCT with GPU implementation for image compression. **Journal of Visual Communication and Image Representation**, Elsevier, v. 40, p. 357–365, 2016.
- 60 JRIDI, M.; ALFALOU, A.; MEHER, P. K. A generalized algorithm and reconfigurable architecture for efficient and scalable orthogonal approximation of DCT. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 62, n. 2, p. 449–457, 2015.

- 61 LENGWEHASATIT, K.; ORTEGA, A. Scalable variable complexity approximate forward DCT. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 14, n. 11, p. 1236–1248, 2004.
- 62 PAUCHARD, Y. *et al.* Fast computation of residual complexity image similarity metric using low-complexity transforms. **IET Image Processing**, v. 9, n. 8, p. 699–708, 2015.
- 63 POTLURI, U. S. *et al.* Improved 8-point approximate DCT for image and video compression requiring only 14 additions. **IEEE Transactions on Circuits and Systems I: Regular Papers**, v. 61, n. 6, p. 1727–1740, 2014.
- 64 TABLADA, C.; BAYER, F. M.; CINTRA, R. J. A class of DCT approximations based on the Feig–Winograd algorithm. **Signal Processing**, Elsevier, v. 113, p. 38–51, 2015.
- 65 LIANG, J.; TRAN, T. D. Fast multiplierless approximation of the DCT with the lifting scheme. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Applications of Digital Image Processing XXIII**. [S.l.], 2000. v. 4115, p. 384–396.
- 66 ISHWAR, S.; MEHER, P. K.; SWAMY, M. Discrete Tchebichef transform—a fast 4x4 algorithm and its application in image/video compression. In: IEEE. **IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008**. [S.l.], 2008. p. 260–263.
- 67 CINTRA, R. J. An integer approximation method for discrete sinusoidal transforms. **Circuits, Systems, and Signal Processing**, Springer, v. 30, n. 6, p. 1481, 2011.
- 68 SEBER, G. A. **A matrix handbook for statisticians**. [S.l.]: John Wiley & Sons, 2008. v. 15.
- 69 BAYER, F. M. *et al.* A digital hardware fast algorithm and FPGA-based prototype for a novel 16-point approximate DCT for image compression applications. **Measurement Science and Technology**, IOP Publishing, v. 23, n. 11, p. 114010, 2012.
- 70 FLURY, B. N.; GAUTSCHI, W. An algorithm for simultaneous orthogonal transformation of several positive definite symmetric matrices to nearly diagonal form. **SIAM Journal on Scientific and Statistical Computing**, v. 7, n. 1, p. 169–184, 1986.
- 71 POTLURI, U. *et al.* Multiplier-free DCT approximations for RF multi-beam digital aperture-array space imaging and directional sensing. **Measurement Science and Technology**, IOP Publishing, v. 23, n. 11, p. 114003, 2012.
- 72 GRAHAM, R. L. *et al.* Concrete mathematics: a foundation for computer science. **Computers in Physics**, AIP, v. 3, n. 5, p. 106–107, 1989.
- 73 OLDHAM, K. B.; MYLAND, J.; SPANIER, J. **An atlas of functions: with equator, the atlas function calculator**. [S.l.]: Springer Science & Business Media, 2010.
- 74 JAYANT, N. S.; NOLL, P. Digital coding of waveforms: principles and applications to speech and video. **Englewood Cliffs, NJ**, p. 115–251, 1984.
- 75 TAKALA, J.; NIKARA, J. Unified pipeline architecture for discrete sine and cosine transforms of type IV. In: **Proceedings of the 3rd Internacional Conference on Information Communication and Signal Processing**. [S.l.: s.n.], 2001.

- 76 HARTIGAN, J. A.; WONG, M. A. Algorithm as 136: A k-means clustering algorithm. **Journal of the Royal Statistical Society. Series C (Applied Statistics)**, Wiley Online Library, v. 28, n. 1, p. 100–108, 1979.
- 77 DOOB, J. L. **Stochastic processes**. [S.l.]: John Wiley & Sons, 1990.
- 78 WRIGHT, S.; NOCEDAL, J. Numerical optimization. **Springer Science**, v. 35, n. 67-68, p. 7, 1999.
- 79 PLONKA, G. A global method for invertible integer dct and integer wavelet algorithms. **Applied and Computational Harmonic Analysis**, Academic Press, v. 16, n. 2, p. 90–110, 2004.
- 80 HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. Unsupervised learning. In: **The elements of statistical learning**. [S.l.]: Springer, 2009. p. 485–585.
- 81 MASERA, M.; MARTINA, M.; MASERA, G. Odd type DCT/DST for video coding: Relationships and low-complexity implementations. In: IEEE. **2017 IEEE International Workshop on Signal Processing Systems (SiPS)**. [S.l.], 2017. p. 1–6.
- 82 SUEHIRO, N.; HATORI, M. Fast algorithms for the DFT and other sinusoidal transforms. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 34, n. 3, p. 642–644, 1986.
- 83 YUAN, W.; HAO, P.; XU, C. Matrix factorization for fast DCT algorithms. In: IEEE. **ICASSP 2006 Proceedings. IEEE International Conference on Acoustics, Speech and Signal Processing, 2006**. [S.l.], 2006. v. 3, p. III–III.
- 84 WANG, Z. Combined DCT and companding for PAPR reduction in OFDM signals. **Journal of Signal and Information Processing**, Scientific Research Publishing, v. 2, n. 2, p. 100–104, 2011.
- 85 SNIGDHA, F. S. *et al.* Optimal design of JPEG hardware under the approximate computing paradigm. In: ACM. **Proceedings of the 53rd Annual Design Automation Conference**. [S.l.], 2016. p. 106.
- 86 SUZUKI, T.; IKEHARA, M. Integer DCT based on direct-lifting of DCT-IDCT for lossless-to-lossy image coding. **IEEE Transactions on Image Processing**, v. 19, n. 11, p. 2958–2965, 2010.
- 87 FONG, C.-K.; CHAM, W.-K. LLM integer cosine transform and its fast algorithm. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 22, n. 6, p. 844–854, 2012.
- 88 CHOI, K.; LEE, S.; JANG, E. S. Zero coefficient-aware IDCT algorithm for fast video decoding. **IEEE Transactions on Consumer Electronics**, v. 56, n. 3, 2010.
- 89 PARK, J.-S. *et al.* 2-D large inverse transform (16×16 , 32×32) for HEVC (High Efficiency Video Coding). **JSTS: Journal of Semiconductor Technology and Science**, The Institute of Electronics Engineers of Korea, v. 12, n. 2, p. 203–211, 2012.
- 90 XU, X. *et al.* Multiple morphological component analysis based decomposition for remote sensing image classification. **IEEE Transactions on Geoscience and Remote Sensing**, v. 54, n. 5, p. 3083–3102, 2016.

- 91 BRIGGS, W. L.; HENSON, V. E. **The DFT: an owner's manual for the discrete Fourier transform**. [S.l.]: SIAM, 1995. v. 45.
- 92 WINOGRAD, S. **Arithmetic complexity of computations**. [S.l.]: SIAM, 1980. v. 33.
- 93 SIPI, U. **The USC-SIPI image database**. 1977. Disponível em: <<http://sipi.usc.edu/database/>>.
- 94 HUYNH-THU, Q.; GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. **Electronics Letters**, IET, v. 44, n. 13, p. 800–801, 2008.
- 95 WANG, Z. *et al.* Image quality assessment: from error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, n. 4, p. 600–612, 2004.
- 96 BOTTA, M.; CAVAGNINO, D.; POMPONIU, V. Fragile watermarking using Karhunen–Loève transform: the KLT-F approach. **Soft Computing**, Springer, v. 19, n. 7, p. 1905–1919, 2015.

APÊNDICE A – TRANSFORMADAS DE COMPRIMENTO OITO

No presente apêndice, são exibidas todas as transformadas de comprimento 8 obtidas aplicando as funções inteiras de arredondamento. Seguem listadas abaixo:

$$\widehat{\mathbf{K}}_7 = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 2 & 2 & 1 \\ 2 & 3 & 2 & 1 & -1 & -2 & -3 & -2 \\ 2 & 2 & 0 & -2 & -2 & 0 & 2 & 2 \\ 3 & 1 & -2 & -2 & 2 & 2 & -1 & -3 \\ 3 & -1 & -2 & 2 & 2 & -2 & -1 & 3 \\ 2 & -2 & 0 & 2 & -2 & 0 & 2 & -2 \\ 2 & -3 & 2 & -1 & -1 & 2 & -3 & 2 \\ 1 & -2 & 2 & -3 & 3 & -2 & 2 & -1 \end{bmatrix}, \quad \widehat{\mathbf{K}}_9 = \begin{bmatrix} 1 & 2 & 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 3 & 3 & 1 & -1 & -3 & -3 & -2 \\ 3 & 3 & 0 & -3 & -3 & 0 & 3 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 3 & -1 & -3 & 2 & 2 & -3 & -1 & 3 \\ 3 & -3 & 0 & 3 & -3 & 0 & 3 & -3 \\ 2 & -3 & 3 & -1 & -1 & 3 & -3 & 2 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix},$$

$$\widehat{\mathbf{K}}_{10} = \begin{bmatrix} 1 & 2 & 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 3 & 2 & 1 & -1 & -2 & -3 & -2 \\ 3 & 2 & 0 & -3 & -3 & 0 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 3 & -1 & -2 & 2 & 2 & -2 & -1 & 3 \\ 2 & -3 & 0 & 3 & -3 & 0 & 3 & -2 \\ 2 & -3 & 3 & -1 & -1 & 3 & -3 & 2 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}, \quad \widehat{\mathbf{K}}_{11} = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 2 & 2 & 1 \\ 2 & 3 & 2 & 1 & -1 & -2 & -3 & -2 \\ 3 & 2 & 0 & -3 & -3 & 0 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 3 & -2 & -2 & 2 & 2 & -2 & -2 & 3 \\ 2 & -3 & 0 & 3 & -3 & 0 & 3 & -2 \\ 2 & -3 & 3 & -1 & -1 & 3 & -3 & 2 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix},$$

$$\widehat{\mathbf{K}}_{12} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 2 & 1 & 0 & 0 & -1 & -2 & -1 \\ 2 & 1 & 0 & -1 & -1 & 0 & 1 & 2 \\ 2 & 0 & -1 & -1 & 1 & 1 & 0 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 1 & -1 & 0 & 2 & -1 \\ 1 & -2 & 1 & 0 & 0 & 1 & -2 & 1 \\ 0 & -1 & 1 & -2 & 2 & -1 & 1 & 0 \end{bmatrix}, \quad \widehat{\mathbf{K}}_{13} = \begin{bmatrix} 1 & 2 & 3 & 3 & 3 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 2 & 0 & -3 & -3 & 0 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 3 & -2 & -3 & 2 & 2 & -3 & -2 & 3 \\ 2 & -3 & 0 & 3 & -3 & 0 & 3 & -2 \\ 2 & -3 & 3 & -1 & -1 & 3 & -3 & 2 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix},$$

$$\widehat{\mathbf{K}}_{14} = \begin{bmatrix} 1 & 2 & 2 & 3 & 3 & 2 & 2 & 1 \\ 2 & 3 & 2 & 1 & -1 & -2 & -3 & -2 \\ 3 & 2 & 0 & -3 & -3 & 0 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 3 & -2 & -2 & 2 & 2 & -2 & -2 & 3 \\ 2 & -3 & 0 & 3 & -3 & 0 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}, \quad \widehat{\mathbf{K}}_{15} = \begin{bmatrix} 1 & 1 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 0 & 0 & -2 & -2 & -2 \\ 2 & 1 & 0 & -2 & -2 & 0 & 1 & 2 \\ 2 & 0 & -2 & -1 & 1 & 2 & 0 & -2 \\ 2 & -1 & -2 & 2 & 2 & -2 & -1 & 2 \\ 2 & -2 & 0 & 2 & -2 & 0 & 2 & -2 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix},$$

$$\widehat{\mathbf{K}}_{16} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 0 & 0 & -1 & -2 & -1 \\ 2 & 1 & 0 & -2 & -2 & 0 & 1 & 2 \\ 2 & 0 & -2 & -1 & 1 & 2 & 0 & -2 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -2 & 0 & 2 & -2 & 0 & 2 & -1 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 0 & -1 & 2 & -2 & 2 & -2 & 1 & 0 \end{bmatrix}, \quad \widehat{\mathbf{K}}_{17} = \begin{bmatrix} 2 & 2 & 2 & 3 & 3 & 2 & 2 & 2 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 2 & -1 & -3 & -3 & -1 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 0 & 3 & -3 & 0 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix},$$

$$\widehat{\mathbf{K}}_{19} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 2 & -1 & -3 & -3 & -1 & 2 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 0 & 3 & -3 & 0 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}, \quad \widehat{\mathbf{K}}_{20} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 1 & -1 & -2 & -2 & -2 \\ 3 & 1 & -1 & -2 & -2 & -1 & 1 & 3 \\ 2 & 0 & -3 & -1 & 1 & 3 & 0 & -2 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 0 & 2 & -2 & 0 & 3 & -2 \\ 1 & -2 & 2 & -1 & -1 & 2 & -2 & 1 \\ 1 & -2 & 2 & -3 & 3 & -2 & 2 & -1 \end{bmatrix},$$

$$\widehat{\mathbf{K}}_{24} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 2 & 1 & -1 & -2 & -3 & -3 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\ 3 & 0 & -3 & -2 & 2 & 3 & 0 & -3 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 1 & 3 & -3 & -1 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 3 & -3 & 3 & -3 & 2 & -1 \end{bmatrix}, \quad \widehat{\mathbf{K}}_{25} = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 2 & 2 & 1 & -1 & -2 & -2 & -3 \\ 3 & 1 & -1 & -3 & -3 & -1 & 1 & 3 \\ 2 & 0 & -3 & -2 & 2 & 3 & 0 & -2 \\ 2 & -2 & -2 & 2 & 2 & -2 & -2 & 2 \\ 2 & -3 & 1 & 2 & -2 & -1 & 3 & -2 \\ 1 & -3 & 3 & -1 & -1 & 3 & -3 & 1 \\ 1 & -2 & 2 & -3 & 3 & -2 & 2 & -1 \end{bmatrix}.$$

APÊNDICE B – BANCO DE IMAGENS

As imagens retiradas do Banco de Imagens Público “*The USC-SIPI Image Database*” [93] utilizadas nos experimentos de compressão de imagens são apresentadas neste apêndice para referência.

