UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DO CABO DE SANTO AGOSTINHO
GRADUAÇÃO EM ENGENHARIA ELETRÔNICA

MANASSIES WILLYAN LEMOS FELIX

Um estudo de topologias para aplicação da transformada de Fourier sobre grafos no contexto de compressão de imagens de *light field*.

Cabo de Santo Agostinho - PE

2026

MANASSIES WILLYAN LEMOS FELIX

Um estudo de topologias para aplicação da transformada de Fourier sobre grafos no contexto de compressão de imagens de *light field*.

Monografia apresentada ao curso de Graduação em Engenharia Eletrônica da Unidade Acadêmica do Cabo de Santo Agostinho, Universidade Federal Rural de Pernambuco, como requisito para obtenção do grau de Bacharel em Engenharia Eletrônica.

Orientador (a): Prof. Dr. Felipe Alberto Barbosa Simão Ferreira.

Cabo de Santo Agostinho - PE

2026

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Rosimeri Couto – CRB-4 1395

F316u Felix, Manassies Willyan Lemos.
Um estudo de topologias para aplicação da transformada de Fourier sobre grafos no contexto de compressão de imagens de *light field*. / Manassies Willyan Lemos Felix. – Cabo de Santo Agostinho, 2026.
75 f.; il.

Orientador(a): Felipe Alberto Barbosa Simão Ferreira.

Trabalho de Conclusão de Curso (Graduação) – Universidade Federal Rural de Pernambuco, Unidade Acadêmica Cabo de Santo Agostinho - UACSA, Bacharelado em Engenharia Eletrônica, Cabo de Santo Agostinho, BR-PE, 2026.

Inclui referências e apêndice(s).

1. Compressão de dados. 2. Imagens de *light field*. 3. Transformada de Fourier sobre grafos. 4. Processamento digital de imagens I. Ferreira, Felipe Alberto Barbosa Simão, orient. II. Título

CDD 621.3

MANASSIES WILLYAN LEMOS FELIX

Um estudo de topologias para aplicação da transformada de Fourier sobre grafos no contexto de compressão de imagens de *light field*.

Monografia apresentada ao Curso de Graduação em Engenharia Eletrônica da Universidade Federal Rural de Pernambuco, como requisito parcial para a obtenção do grau de Bacharel em Engenharia Eletrônica.

Aprovada em: 13/07/2026.

BANCA EXAMINADORA

_____

Felipe Alberto Barbosa Simão Ferreira
Universidade Federal Rural de Pernambuco

_____

Elias Marques Ferreira de Oliveira
Universidade Federal Rural de Pernambuco

_____

Fernando Gonçalves de Almeida Neto
Universidade Federal Rural de Pernambuco

Dedico este trabalho à minha mãe, Lindacy de Andrade Lemos, por todo o apoio e amor ao longo dessa jornada. Sem ela, certamente, nenhuma dessas palavras seria escrita.

AGRADECIMENTOS

À minha mãe, Lindacy de Andrade Lemos, por ter sido meu pilar e porto seguro durante toda esta longa jornada — desde o momento em que decidi qual caminho seguir até a consolidação deste sonho.

Ao meu orientador, Professor Felipe Alberto Barbosa Simão Ferreira, por todo o apoio, paciência, dedicação e genuína vontade de ensinar.

Aos amigos que a vida me deu, antes e durante a graduação. Muito obrigado por tornarem o fardo mais leve, por dividirem as angústias e por celebrarem as vitórias comigo.

Aos professores que fizeram parte da minha formação acadêmica. Ensinar é, sem dúvida, uma das formas mais bonitas e generosas de demonstrar amor.

“Eu não falhei. Apenas descobri dez mil maneiras que não funcionam.”

Thomas Edison.

RESUMO

Imagens de *light field* (LF) representam uma tecnologia em ascensão. Sua aplicação é vasta e pode variar desde aplicações mais óbvias, como para entretenimento, até aplicações mais profissionais. Estas imagens são capazes de variar o ângulo da cena capturada, focar em outros objetos da fotografia mesmo após a captura, entre outras funcionalidades. Um dos grandes problemas para a aplicabilidade dessa tecnologia é a sua estrutura de dados complexa e volumosa, que gera arquivos gigantescos, exigindo alta capacidade de armazenamento em disco e de banda para transmissão. Atualmente, ainda não existe um método definitivo ou um padrão universal de compressão voltado especificamente para as LFs. É comum o uso de métodos de compressão convencionais para contornar problemas de tamanho, porém, essas soluções lidam mal com a natureza multidimensional da LF. Uma imagem de LF é visualizada como uma matriz de subimagens, e as abordagens tradicionais, que exploram apenas a similaridade espacial 2D, falham ao tentar comprimir a transição muitas vezes abrupta entre essas subimagens, gerando perdas de qualidade. Por outro lado, as subimagens de uma imagem de LF podem parecer repetitivas entre si, já que, na maioria das vezes, as variações de ângulo são bastante sutis. Dessa forma, é possível explorar essa redundância de forma mais otimizada utilizando a transformada de Fourier sobre grafos (GFT, do inglês, *Graph Fourier Transform*), já que, considerando vértices de um grafo como pixels de uma imagem, os pixels semelhantes (de intensidade parecida) entre as subimagens podem ser ligados entre si. Neste trabalho, diferentes topologias de grafos foram desenvolvidas e avaliadas para estruturar essas conexões. Os resultados demonstraram que a aplicação da GFT, especificamente utilizando uma topologia híbrida que une proximidade espacial e correspondência angular (Pixel Vizinho + Gêmeo), superou a tradicional 2D-DCT em cenários de alta taxa de compressão. A abordagem proposta garantiu uma qualidade visual superior das imagens reconstruídas, comprovada por ganhos significativos nas métricas PSNR e SSIM sob um mesmo limite teórico ditado pela Entropia de Shannon, validando o uso de grafos como uma alternativa promissora e eficiente para a compressão de mídias plenópticas.

Palavras-chave: compressão de dados; imagens de *light field*; transformada de Fourier sobre grafos; processamento digital de imagens.

ABSTRACT

Light field (LF) images represent an emerging technology. Their application is vast and can range from obvious uses, such as entertainment, to more professional applications. These images are capable of varying the angle of the captured scene and focusing on different objects in the photograph even after capture, among other features. One of the major obstacles to the applicability of this technology is its complex and voluminous data structure, which generates massive files, demanding high storage capacity and transmission bandwidth. Currently, there is still no definitive method or universal standard for compression specifically designed for LFs. It is common to use conventional compression methods to bypass size issues, however, these solutions handle the multidimensional nature of LF poorly. An LF image can be visualized as a matrix of sub-images, and traditional approaches, which exploit only 2D spatial similarity, fail when attempting to compress the often abrupt transition between these sub-images, leading to quality loss. On the other hand, sub-images within an LF image can appear repetitive, as angle variations are mostly subtle. Thus, it is possible to exploit this redundancy more optimally by using the Graph Fourier Transform (GFT), since, by treating graph vertices as image pixels, similar pixels (with similar intensity) across sub-images can be linked together. In this work, different graph topologies were developed and evaluated to structure these connections. The results demonstrated that the application of GFT, specifically using a hybrid topology that combines spatial proximity and angular correspondence (Neighbor + Twin Pixel), outperformed the traditional 2D-DCT in high compression rate scenarios. The proposed approach ensured superior visual quality of the reconstructed images, supported by significant gains in the PSNR and SSIM metrics under the same theoretical limit dictated by Shannon Entropy, validating the use of graphs as a promising and efficient alternative for the compression of plenoptic media.

Keywords: data compression; light field imaging; graph Fourier transform; digital image processing.

LISTA DE ILUSTRAÇÕES

Figura 1 - Subabertura de uma imagem de light field. 17
Figura 2 - Grafos (a) direcional e (b) não direcional. 19
Figura 3 - Matriz de adjacência do grafo da (a) Figura 2(a) e (b) Figura 2(b). 19
Figura 4 - (a) Um grafo proporcional e (b) sua matriz de pesos. 20
Figura 5 - Exemplo de imagem de light field em um array 8x8. 21
Figura 6 - Uma matriz (a) decimal, (b) em binário e (c) a imagem formada por ela. 24
Figura 7 - Produzindo uma imagem digital. (a) Imagem contínua. (b) Linha de varredura de A e B na imagem contínua para ilustrar os conceitos de amostragem e quantização. (c) Amostragem e quantização. (d) Linha de varredura digital. 25
Figura 8 - (a) Um sinal contínuo e (b) seu espectro de Fourier. 26
Figura 9 - (a) A imagem de entrada (uma versão reescalonada em 64x64 da capa do álbum "Da Lama Ao Caos" de Chico Science & Nação Zumbi), (b) coeficientes 1D e (c) 2D da imagem usando a 2D-DCT ordenando os coeficientes no formato zig-zag. 27
Figura 10 - Ilustração do funcionamento do método de zig-zag. 29
Figura 11 - Exemplo de (a) um array de câmeras utilizado para capturar imagens de light field e (b) uma imagem capturada por este tipo de sistema. 30
Figura 12 - (a) Uma modelo sendo capturada por uma câmera de microlentes Raytrix e (b) um exemplo de visualização 3D da imagem pós-processada. 30
Figura 13 - Uma imagem de light field em escala de cinza de uma rua com prédios. 31
Figura 14 - Uma imagem de light field em escala de cinza de uma tabela de cores. 31
Figura 15 - Uma imagem de light field em escala de cinza de um grupo de pessoas. 31
Figura 16 - Visualização de todas as vistas de uma imagem de light field capturada por um dispositivo com microlentes (lenslet), em que cada vista é referenciada por um sistema de coordenadas (s, t), e a vista central expandida, em que cada pixel é referenciado por um sistema de coordenadas (u, v). 32
Figura 17 - (a) Um exemplo de imagem e (b) sua representação como grafo. 33
Figura 18 - (a) imagem de referência, (b) topologia que liga pixels ao redor e (c) topologia que liga pixels de mesma intensidade 34
Figura 19 – Gráfico dos coeficientes da GFT da Figura 18 usando (a) a topologia que lida pixels ao redor e (b) a topologia que liga pixels de mesma intensidade. 35
Figura 20 – Pixel art criada para ilustrar uma imagem de light field. 38
Figura 21 - (a) Mostrando a área capturada em cada subimagem e (b) a imagem resultante do processo de captura das áreas. 38

Figura 22 - (a) Exemplo discreto para a topologia do pixel vizinho e (b) exemplo da topologia usada em Python. 39
Figura 23 - (a) Exemplo discreto para a topologia do pixel gêmeo e (b) exemplo da topologia usada em Python. 40
Figura 24 - (a) Exemplo discreto para a topologia do pixel vizinho + pixel gêmeo e (b) exemplo da topologia usada em Python. 41
Figura 25 - Exemplo para a topologia do pixel gêmeo proporcional. 42
Figura 26 - (a) Exemplo discreto para a topologia da vizinhança gêmea e (b) exemplo da topologia usada em Python. 43
Figura 27 - Exemplo para a topologia da vizinhança gêmea proporcional. 44
Figura 28 - Fluxograma do CODEC elaborado. 44
Figura 29 - Complexidades comuns de Big-O. 46
Figura 30 -Valores médios de PSNR (dB) obtidos pelas topologias propostas para diferentes níveis de entropia. 50
Figura 31 - Valores médios de SSIM obtidos pelas topologias propostas para diferentes níveis de entropia. 50
Figura 32 - Comparativo do PSNR e da entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT. 52
Figura 33 - Comparativo do SSIM e da entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT. 52
Figura 34 - Consumo de RAM por método utilizado. 54
Figura 35 - Uso de CPU por método utilizado. 54
Figura 36 - Média do tempo de processamento por método utilizado. 56

LISTA DE TABELAS

Tabela A. 1 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Gêmeo. 62
Tabela A. 2 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Vizinho. 62
Tabela A. 3 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Vizinho + Gêmeo. 63
Tabela A. 4 - Valores de PSNR e SSIM das três imagens de teste com a topologia Vizinhança Gêmea. 63
Tabela A. 5 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Gêmeo Proporcional. 64
Tabela A. 6 - Valores de PSNR e SSIM das três imagens de teste com a topologia Vizinhança Gêmea Proporcional. 64
Tabela A. 7 - Valores de PSNR e SSIM das três imagens de teste com aplicação da 2D-DCT. 65
Tabela A. 8 - Valores médios de PSNR (dB) obtidos pelas topologias propostas para diferentes níveis de entropia. 65
Tabela A. 9 - Valores médios de SSIM obtidos pelas topologias propostas para diferentes níveis de entropia. 66
Tabela A. 10 - Comparativo do PSNR e entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT. 66
Tabela A. 11 - Comparativo do SSIM e entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT. 67
Tabela A. 12 - Medições de consumo de RAM e CPU na topologia do Pixel Gêmeo. 67
Tabela A. 13 - Medições de consumo de RAM e CPU na topologia do Pixel Vizinho. 68
Tabela A. 14 - Medições de consumo de RAM e CPU na topologia do Pixel Vizinho + Gêmeo. 68
Tabela A. 15 - Medições de consumo de RAM e CPU na topologia da Vizinhança Gêmea. 69
Tabela A. 16 - Medições de consumo de RAM e CPU na topologia do Pixel Gêmeo Proporcional. 69
Tabela A. 17 - Medições de consumo de RAM e CPU na topologia da Vizinhança Gêmea Proporcional. 70
Tabela A. 18 - Medições de consumo de RAM e CPU na 2D-DCT. 70

Tabela A. 19 - Média das medições de RAM e CPU das três imagens de teste para diferentes métodos. 71
Tabela A. 20 - Medições do tempo usado em segundos para a topologia do Pixel Gêmeo nas diferentes imagens. 71
Tabela A. 21 - Medições do tempo usado em segundos para a topologia do Pixel Vizinho nas diferentes imagens. 72
Tabela A. 22 - Medições do tempo usado em segundos para a topologia do Pixel Vizinho + Gêmeo nas diferentes imagens..... 72
Tabela A. 23 - Medições do tempo usado em segundos para a topologia da Vizinhança Gêmea nas diferentes imagens. 73
Tabela A. 24 - Medições do tempo usado em segundos para a topologia do Pixel Gêmeo Proporcional nas diferentes imagens. 73
Tabela A. 25 - Medições do tempo usado em segundos para a topologia da Vizinhança Gêmea Proporcional nas diferentes imagens. 74
Tabela A. 26 - Medições do tempo usado em segundos para o método da 2D-DCT nas diferentes imagens. 74
Tabela A. 27 - Média das medições de tempo das três imagens de teste para diferentes métodos. 75

LISTA DE EQUAÇÕES

Equação 1 - Transformada linear bidimensional direta. 25
Equação 2 - Transformada linear bidimensional inversa. 25
Equação 3 - Transformada discreta do cosseno em duas dimensões (2D-DCT). 28
Equação 4 - Definição da constante da transformada discreta do cosseno. 28
Equação 5 - Transformada discreta do cosseno em duas dimensões inversa. 28
Equação 6 - Transformada de Fourier sobre grafos (GFT). 33
Equação 7 - Transformada de Fourier sobre grafos inversa (IGFT). 33
Equação 8 - Variação total de um sinal sobre o grafo. 34
Equação 9 - Relação Sinal-Ruído de Pico (PSNR). 35
Equação 10 - Índice de Similaridade Estrutural (SSIM). 36
Equação 11 - Entropia de Shannon. 36
Equação 12 - Regra de conexão para a topologia do Pixel Vizinho. 40
Equação 13 - Regra de conexão para a topologia do Pixel Gêmeo. 41
Equação 14 - Regra de conexão para a topologia do Pixel Vizinho + Pixel Gêmeo. 41
Equação 15 - Cálculo do peso da aresta por distância euclidiana. 42
Equação 16 - Regra de conexão para a topologia da Vizinhança Gêmea. 43

LISTA DE ABREVIATURAS E SIGLAS

1D – Unidimensional

2D – Bidimensional

2D-DCT – *Two-Dimensional Discrete Cosine Transform* (Transformada Discreta de Cosseno Bidimensional)

3D – Tridimensional

4D – Quatridimensional

CMYK – *Cyan, Magenta, Yellow, Key/Black* (Ciano, Magenta, Amarelo e Preto)

CPU – *Central Processing Unit* (Unidade Central de Processamento)

DCT – *Discrete Cosine Transform* (Transformada Discreta de Cosseno)

FFT – *Fast Fourier Transform* (Transformada Rápida de Fourier)

FGFT – *Fast Graph Fourier Transform* (Transformada Rápida de Fourier em Grafos)

GFT – *Graph Fourier Transform* (Transformada de Fourier em Grafos)

GPU – *Graphics Processing Unit* (Unidade de Processamento Gráfico)

IGFT – *Inverse Graph Fourier Transform* (Transformada Inversa de Fourier em Grafos)

LF – *Light Field* (Campo de Luz)

PDI – Processamento Digital de Imagens

PSNR – *Peak Signal-to-Noise Ratio* (Relação Sinal-Ruído de Pico)

RAM – *Random Access Memory* (Memória de Acesso Aleatório)

RGB – *Red, Green, Blue* (Vermelho, Verde, Azul)

SSIM – *Structural Similarity Index Measure* (Índice de Similaridade Estrutural)

SUMÁRIO

1 INTRODUÇÃO 17

1.1 CONTEXTUALIZAÇÃO 17
1.2 A PROBLEMÁTICA 20
1.3 OBJETIVOS 21
1.3.1 Objetivo geral 21
1.3.2 Objetivos específicos 22

2 FUNDAMENTAÇÃO TEÓRICA 23

2.1 PROCESSAMENTO DIGITAL DE SINAIS E IMAGENS 23
2.1.1 Processamento de imagens no domínio da transformada 25
2.1.2 A transformada discreta do cosseno em duas dimensões (2D-DCT) 27
2.2 IMAGENS DE *LIGHT FIELD* 29
2.3 TEORIA DOS GRAFOS 32
2.4 A TRANSFORMADA DE FOURIER SOBRE GRAFOS (GFT) 33
2.4.1 Como a topologia influencia na distribuição dos coeficientes 34
2.5 MÉTRICAS DE QUALIDADE E COMPRESSÃO DE IMAGEM 35

3 METODOLOGIA DE COMPRESSÃO E TOPOLOGIAS PROPOSTAS 37

3.1 DIVISÃO DE BLOCOS 37
3.2 TOPOLOGIAS 38
3.2.1 Pixel Vizinho 39
3.2.3 Pixel vizinho + Pixel gêmeo 41
3.2.4 Pixel gêmeo proporcional 41
3.2.5 Vizinhança gêmea 42
3.2.6 Vizinhança gêmea proporcional 43
3.3 FLUXO DE CODEC 44
3.3.1 Codificação 44
3.3.2 Decodificação 45

4 ANÁLISE DE COMPLEXIDADE E CUSTO COMPUTACIONAL 46

4.1 ANÁLISE DE COMPLEXIDADE ALGORÍTMICA 46
4.2 GARGALOS DE PROCESSAMENTO 47

5 RESULTADOS E DISCUSSÃO 49

5.1 DESEMPENHO DE COMPRESSÃO 49
5.2 ANÁLISE DE HARDWARE 53
5.3 ANÁLISE DE TEMPO 55

6 CONCLUSÃO 57

6.1 CONSIDERAÇÕES GERAIS 57
6.2 LIMITAÇÕES E TRABALHOS FUTUROS 58
REFERÊNCIAS 60
APÊNDICE A – Tabelas 62

1 INTRODUÇÃO

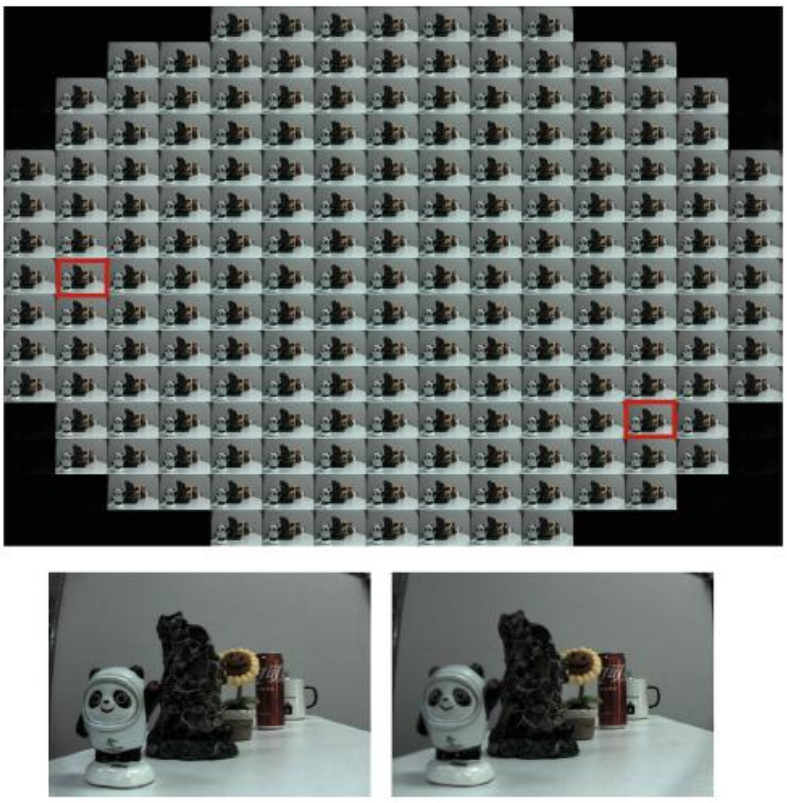
Nesta seção, são apresentados os conceitos preliminares sobre imagens de *light field*, a problemática atrelada ao armazenamento dessas mídias e os objetivos traçados para a realização deste trabalho.

1.1 CONTEXTUALIZAÇÃO

As imagens de *light field* (LF) fazem parte de uma tecnologia ainda muito recente. Baseadas em funções plenópticas, essas imagens representam uma evolução em relação às imagens bidimensionais (2D) comuns e às exibições de imagens planas. (Pereira; Silva; Lafruit, 2018)

Diferentes das imagens 2D, ou “padrão”, as imagens de LF não se baseiam somente em informações de duas dimensões espaciais (*x, y)* e comprimento de onda (λ), mas, assim como em funções plenópticas, em componentes de posição (*x, y, z*), ângulo (θ, ϕ) e comprimento de onda (λ) (Fang, 2025). A Figura 1 mostra a subabertura de uma imagem de *light field*.

Figura 1 - Subabertura de uma imagem de *light field*.



Fonte: Fang, 2025.

A grande quantidade de informação capturada na aquisição de uma imagem de *light field* permite uma ampliação da percepção humana sobre mídias digitais, possibilitando uma experiência mais completa e realista em imagens (Pereira; Silva; Lafruit, 2018).

Praticamente todos os tipos de mídia digital podem ser transmitidos e exibidos com o uso da internet, o que impulsionou diretamente a sua popularização no cotidiano das pessoas. Nesse sentido, um fator crucial e comum à viabilidade desses arquivos é a sua capacidade de compressão e descompressão.

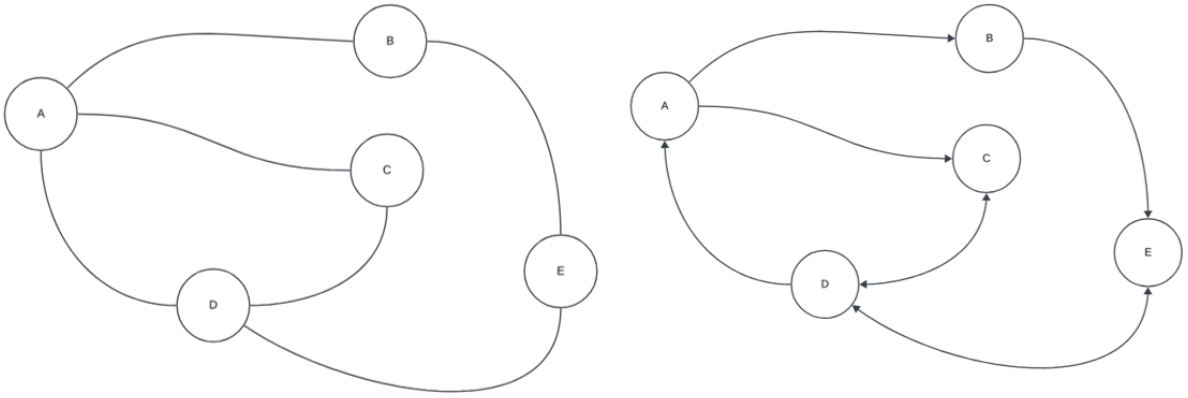
A compressão de dados ocorre a partir de um processo de CODEC (sistema codificador-decodificador) que pode utilizar diversos métodos. Em outras palavras, tendo um arquivo no formato *raw data* (arquivo bruto), ele passa por um processo chamado *encoder* (codificador) no qual a saída é um arquivo comprimido. Após isso, o arquivo pode ser transmitido e, no seu destino, passar por um processo chamado *decoder* (decodificador). O papel deste é descomprimir o arquivo e reconstruí-lo a partir do sinal comprimido. O CODEC é o *software* ou *hardware* que realiza os processos de *encoder* e *decoder* (Gonzalez; Woods, 2010).

Para imagens usuais (2D), a técnica de compressão mais utilizada atualmente é a transformada discreta do cosseno bidimensional (2D-DCT), que é um caso especial da transformada de Fourier. Esse é o principal método utilizado pelo JPEG (*Joint Photographic Experts Group*) e, por isso, será usado como comparação para as técnicas desenvolvidas neste trabalho.

A transformada de Fourier permite analisar sinais no domínio da frequência, onde cada sinal é representado por componentes de frequência senoidais, com amplitude e fase (Ell; Le Bihan; Sangwine, 2014). Quando aplicada a imagens, é possível obter uma resposta onde altas concentrações de energia são evidenciadas em determinadas faixas de frequência, a depender da imagem.

A teoria dos grafos é uma forte ferramenta matemática que permite relacionar pontos que não estão necessariamente próximos, de modo que seja possível associar vértices distantes de forma direta ou até mesmo de forma parcial. Um grafo consiste em um conjunto de vértices não nulos, cujos pares interconectados são chamados de arestas. Estruturalmente, os grafos podem ser do tipo direcional, no qual indica-se explicitamente o vértice de origem da ligação, ou não-direcional, que, por sua vez, indica apenas a relação recíproca entre dois vértices, desconsiderando a direção de partida da aresta. Exemplos desses tipos de grafos podem ser visualizados na Figura 2. Neste trabalho, serão utilizadas apenas topologias não-direcionais.

Figura 2 - Grafos (a) direcional e (b) não direcional.



(a) (b)

Fonte: elaborado pelo próprio autor.

Matematicamente falando, é comum que grafos sejam representados por meio de matrizes que indicam a ligação de vértices. A representação matemática dessas estruturas utiliza comumente dois tipos de matrizes: a de adjacência (***A***) e a de pesos (***W***). Na primeira, utiliza-se "1" para indicar a presença de uma aresta e "0" para a sua ausência, conforme exemplificado na Figura 3. Em contrapartida, a matriz de pesos transcende a mera indicação de existência da aresta, quantificando também a força (ou custo) dessa ligação, como ilustrado na Figura 4. É possível, como será mostrado posteriormente, considerar uma imagem digital como um grande grafo onde cada pixel corresponde a um vértice. Desse modo, criar arestas entre vértices seria o mesmo que criar ligações entre pixels.

Figura 3 - Matriz de adjacência do grafo da (a) Figura 2(a) e (b) Figura 2(b).

Não direcional					
	A	B	C	D	E
A	0	1	1	1	0
B	1	0	0	0	1
C	1	0	0	1	0
D	1	0	1	0	1
E	0	1	0	1	0

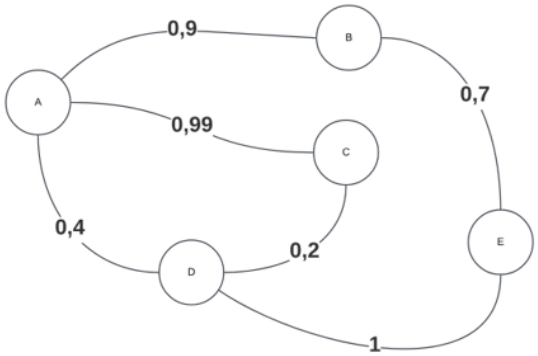
(a)

Direcional					
	A	B	C	D	E
A	0	1	1	0	0
B	0	0	0	0	1
C	0	0	0	1	0
D	1	0	1	0	1
E	0	0	0	1	0

(b)

Fonte: elaborado pelo próprio autor.

Figura 4 - (a) Um grafo proporcional e (b) sua matriz de pesos.



(a)

Matriz de pesos					
	A	B	C	D	E
A	0	0,9	0,99	0,4	0
B	0,9	0	0	0	0,7
C	0,99	0	0	0,2	0
D	0,4	0	0,2	0	1
E	0	0,7	0	1	0

(b)

Fonte: elaborado pelo próprio autor.

A Transformada de Fourier sobre grafos (GFT - *Graph Fourier Transform*) é uma extensão da Transformada de Fourier clássica para domínios não euclidianos. Enquanto a Transformada de Fourier convencional opera em sinais definidos em grades regulares (como tempo ou pixels em uma imagem), a GFT lida com sinais definidos nos vértices de um grafo, em que a relação entre os dados é dada pela topologia de conectividade (Shuman *et al*., 2013).

1.2 A PROBLEMÁTICA

Uma imagem digital comum em HD (alta definição) possui uma dimensão de 1280×720 pixels. Isso significa que a imagem possui 1280 pixels na horizontal e 720 na vertical, totalizando 921 600 pixels.

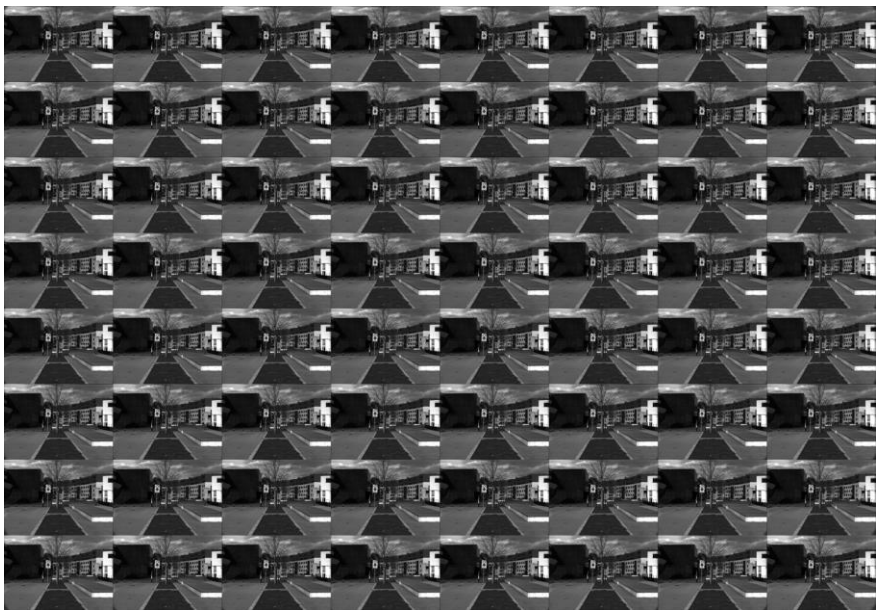
Matematicamente, é possível considerar uma imagem digital como uma matriz, sendo cada pixel um elemento. Sendo assim, uma figura HD poderia ser representada por uma matriz de 1280 colunas e 720 linhas. Cada elemento da matriz representa então a intensidade de cor do pixel.

Por sua vez, imagens de *light field*, embora derivem de funções plenópticas descritas em múltiplas dimensões, para fins de processamento digital, utiliza-se uma parametrização de dois planos paralelos. Nessa simplificação, a imagem de LF é reduzida a um sistema de coordenadas quadridimensionais (s, t, u, v) (Ferreira, 2020). Em que s e t representam as coordenadas angulares, ou seja, a posição da vista ou microlente no *array* de captura, e u e v são as coordenadas espaciais, que representam a posição do pixel dentro de cada uma das subimagens. Isso será explorado de forma mais clara na Seção 2.2.

Dessa forma, é possível pensar em uma imagem de *light field* como uma matriz de imagens planas, logo, infere-se que a demanda de armazenamento para essas imagens será

significativamente maior. A Figura 5 será uma das imagens de LF usadas ao longo do trabalho. Ela possui dimensões de $8 \times 8 \times 624 \times 432$, totalizando 17 252 352 pixels, cerca de 12 vezes maior que uma imagem HD.

Figura 5 - Exemplo de imagem de *light field* em um array 8x8.



Fonte: elaborado pelo próprio autor.

Quando se trata de tamanho de armazenamento, não se pode considerar apenas a quantidade de pixels, mas também canais de cor, taxa de *bits* etc. Esses parâmetros serão explicados posteriormente, mas por enquanto é possível basear-se na quantidade de dados a partir dessa comparação.

Apesar das grandes vantagens em usar esse tipo de imagem, é notório que sua popularização sofre com as dificuldades relacionadas ao armazenamento, envio e recebimento (transmissão) dessa grande quantidade de dados. Nesse sentido, é preciso pensar em soluções e formas de diminuir essa demanda de espaço de armazenamento, ou seja, formas de compressão para essas imagens, já que os trabalhos nessa área permanecem pouco desenvolvidos, apesar do crescente interesse.

1.3 OBJETIVOS

Esta seção estabelece as metas da pesquisa, separando-as em um objetivo geral, que define o propósito central do estudo, e em objetivos específicos, que mapeiam as etapas metodológicas necessárias para alcançá-lo.

1.3.1 Objetivo geral

O objetivo geral deste trabalho é estudar diferentes topologias de grafo que melhor geram resultados de compressão de imagens de *light field* quando aplicada a transformada de Fourier sobre grafos. As topologias estudadas serão analisadas em termos da sua capacidade de concentrar energia em baixas frequências e da qualidade da imagem recuperada após a compressão.

1.3.2 Objetivos específicos

- Estudar conceitos básicos sobre processamento digital de imagens;
- Estudar conceitos básicos para manipulação de imagens de *light field*;
- Estudar conceitos básicos sobre a transformada de Fourier sobre grafos;
- Estudar e desenvolver topologias de grafo que podem ser aplicadas em imagens de *light field;*
- Implementar topologias de grafo estudadas e realizar testes nestas;
- Avaliar as topologias em termos de sua capacidade de compressão e de manutenção da qualidade após a reconstrução e do gasto temporal e computacional utilizado.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta a base conceitual para a compreensão da pesquisa. A revisão aborda os princípios do processamento digital de imagens, a representação de sinais no domínio da frequência, a estrutura das imagens de *light field*, os fundamentos da Transformada de Fourier sobre Grafos e as métricas utilizadas para a análise de compressão.

2.1 PROCESSAMENTO DIGITAL DE SINAIS E IMAGENS

Entender o processamento de imagens passa por compreender como são capturadas, processadas e transformadas em bits.

O ser humano é capaz de, através de seus olhos, detectar luz e interpretá-la em diversos aspectos, como intensidade, comprimento de onda, profundidade, espaço, ângulo, etc. Isso permite uma facilidade na locomoção, comunicação, percepção do tempo, de perigo e de espaço. A tecnologia vem tentando aproximar-se da percepção humana e interpretar a luz de maneiras cada vez mais complexas. Essa aproximação é viabilizada através de imagens em seus diversos tipos, como serão explicados.

Uma imagem digital pode ser descrita como uma ou um conjunto de matrizes cujo tamanho depende de sua resolução. Por exemplo, uma imagem definida como HD (alta definição) possui uma resolução de 1280×720 pixels, portanto, sua matriz equivalente teria um tamanho de 1280 colunas e 720 linhas. Já uma imagem *Full* HD (alta definição completa) possui 1920×1080 pixels. Sua matriz segue esse tamanho. Cada elemento da matriz é equivalente a um pixel da imagem e o valor do elemento remete à intensidade do pixel em uma escala de cor. Imagens podem ser definidas, além de sua resolução, pela quantidade de canais e sua profundidade de cor.

Os canais de uma imagem digital definem a quantidade de matrizes necessárias para compô-la e isso depende do esquema de cores e da aplicação. Os esquemas de cores mais comuns são RGB (*Red, Green, Blue* – Vermelho, Verde, Azul), CMYK (*Cyan, Magenta, Yellow, Key/Black* – Ciano, Magenta, Amarelo, Preto) e *grayscale* (Escala de cinza). O RGB é amplamente encontrado em imagens digitais coloridas. O CMYK é mais comum em imagens destinadas à impressão gráfica. O *grayscale* é um dos tipos mais simples, que se utiliza apenas de um canal de cor, frequentemente referido como imagem “preto e branco”, e é o tipo que será usado nesse trabalho, pois a complexidade algorítmica de aplicar a GFT a mais canais de cor pode aumentar.

A profundidade de cor define a quantidade de bits necessários para armazenar cada pixel de uma imagem. Como já foi definido anteriormente, para formar uma imagem em preto e branco é necessário o uso de apenas uma matriz. No uso cotidiano, a profundidade de 8 bits por canal (*True Color* – Cor verdadeira) é uma escala suficiente para o usuário padrão. Em escala de cinza com essa profundidade, cada pixel possui 8 bits, ou seja, 256 (2^8) cores disponíveis para visualização, onde 0 é a cor preta, 255 é a cor branca e os números intermediários são tons na escala de cinza. A Figura 6 mostra uma matriz em números decimais, binários com profundidade de 8 bits e a imagem formada por ela. Em aplicações profissionais, é comum utilizar uma profundidade de bits maior para, assim, obter uma gama maior de cores.

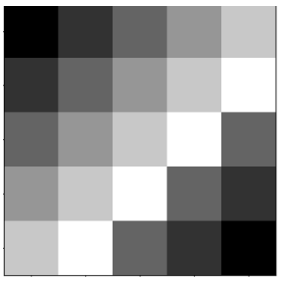
Figura 6 - Uma matriz (a) decimal, (b) em binário e (c) a imagem formada por ela.

0	50	100	150	200
50	100	150	200	255
100	150	200	255	100
150	200	255	100	50
200	255	100	50	0

(a)

00000000	00110010	01100100	10010110	11001000
00110010	01100100	10010110	11001000	11111111
01100100	10010110	11001000	11111111	01100100
10010110	11001000	11111111	01100100	00110010
11001000	11111111	01100100	00110010	00000000

(b)



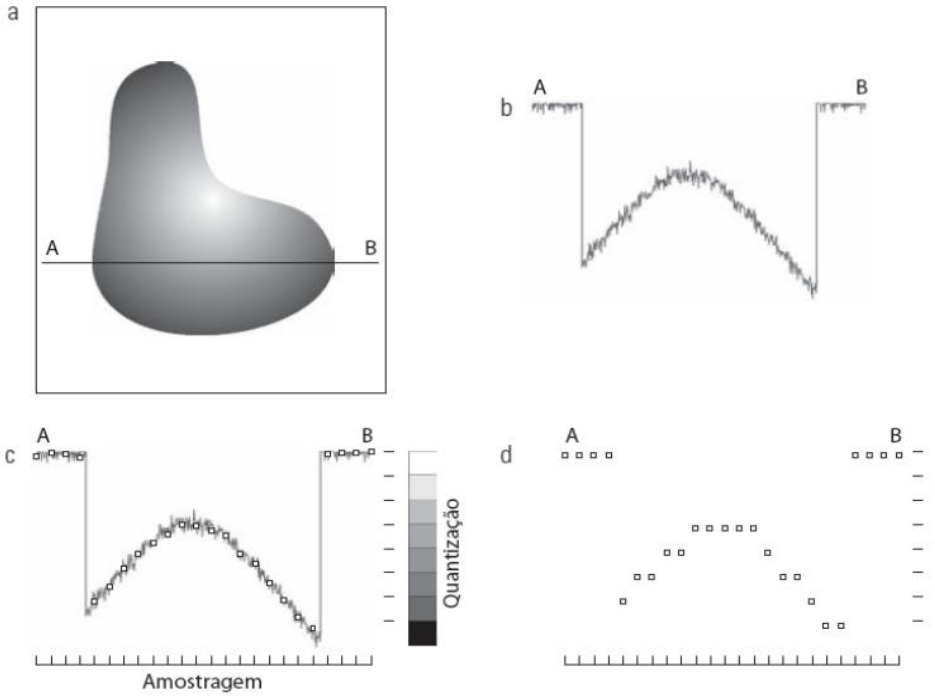
(c)

Fonte: elaborado pelo próprio autor.

Pode-se obter uma imagem digital a partir de um sensor ou um conjunto de sensores. A energia obtida por cada sensor é transformada em tensão e o sinal de entrada é obtido na forma analógica. O sensor mais conhecido é o fotodiodo, que gera uma saída correspondente à intensidade de luz. Para obter uma imagem 2D (bidimensional), pode-se utilizar métodos de deslocamento de sensor, onde a energia é capturada e o sensor é movido por uma grade, porém, o método mais comum e utilizado em câmeras digitais é por meio de uma matriz de sensores que podem incluir milhões destes. (Gonzalez; Woods, 2010)

O sinal analógico de tensão obtido pelos sensores é digitalizado através de técnicas de amostragem e quantização. A Figura 7 exemplifica os processos de digitalização de uma imagem, onde 7(a) mostra uma imagem contínua, a qual se quer digitalizar, sendo a linha AB a posição dos sensores e 7(b) mostra as tensões convertidas pelos sensores de forma contínua ao longo da linha AB. Em 7(c), o sinal passa por um processo de amostragem (onde somente alguns dos sinais dos sensores são considerados) e quantização (onde somente alguns níveis de tensão são considerados). Ao fim do processo de digitalização, obtém-se 7(d), o sinal digital da linha de sensores AB. (Gonzalez; Woods, 2010)

Figura 7 - Produzindo uma imagem digital. (a) Imagem contínua. (b) Linha de varredura de A e B na imagem contínua para ilustrar os conceitos de amostragem e quantização. (c) Amostragem e quantização. (d) Linha de varredura digital.



Fonte: Gonzalez e Woods, 2010.

2.1.1 Processamento de imagens no domínio da transformada

A aplicação do domínio da transformada no processamento de imagens é vasta. Neste trabalho, a transformada será usada no contexto de compressão de imagens.

Uma classe particularmente importante de transformadas lineares bidimensionais $T(u_f, v_f)$ pode ser expressa na forma geral conforme,

$$T(u_f, v_f) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) r(x, y, u_f, v_f), \quad (1)$$

em que $f(x,y)$ é a imagem digital de entrada, $r(x, y, u_f, v_f)$ é chamado de *kernel* (núcleo) de transformação direta e essa equação é calculada para u partindo de 0 até $M-1$ e v partindo de 0 até $N-1$ onde M e N são referentes às dimensões da imagem (linhas e colunas da matriz). A transformada inversa retoma a imagem no domínio espacial e é dada por:

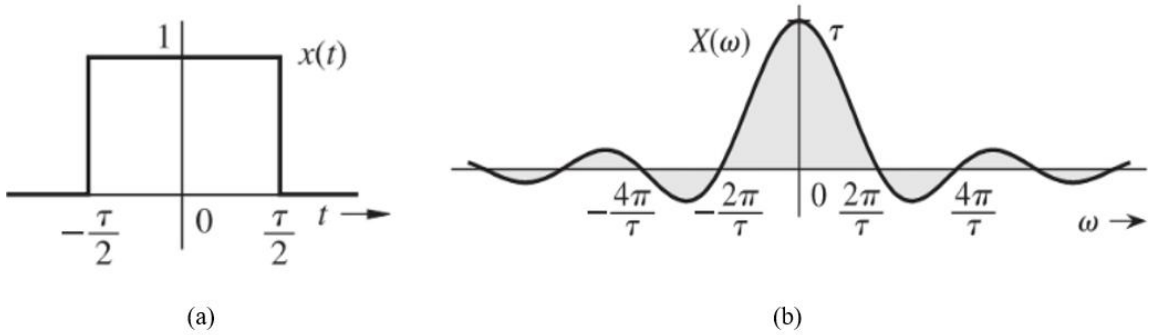
$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u_f, v_f) s(x, y, u, v), \quad (2)$$

na qual $T(u_f, v_f)$ é a transformada da imagem, $s(x, y, u_f, v_f)$ é o *kernel* de transformação inversa. As Equações 1 e 2 são chamadas de par de transformadas. (Gonzalez; Woods, 2010)

Neste trabalho, observar o domínio da frequência será bastante útil na compressão de imagens, posto que, em uma compressão simples, pode-se aplicar um filtro passa-baixas, visto que é possível, por meio da GFT, organizar as principais informações de uma imagem concentrando-as nas baixas frequências (Shuman *et al*., 2013). Ou seja, ao anular componentes de alta frequência, é possível reduzir a quantidade de informações armazenadas e ainda assim manter uma qualidade consideravelmente boa quando feita a descompressão.

Em alguns sinais unidimensionais é fácil notar as maiores energias nas baixas frequências. A Figura 8 mostra um exemplo de um sinal contínuo e seu espectro de Fourier. Nota-se que, no domínio da frequência, existem picos de energia nas baixas frequências e a intensidade diminui à medida que a frequência aumenta. (Lathi, 2014)

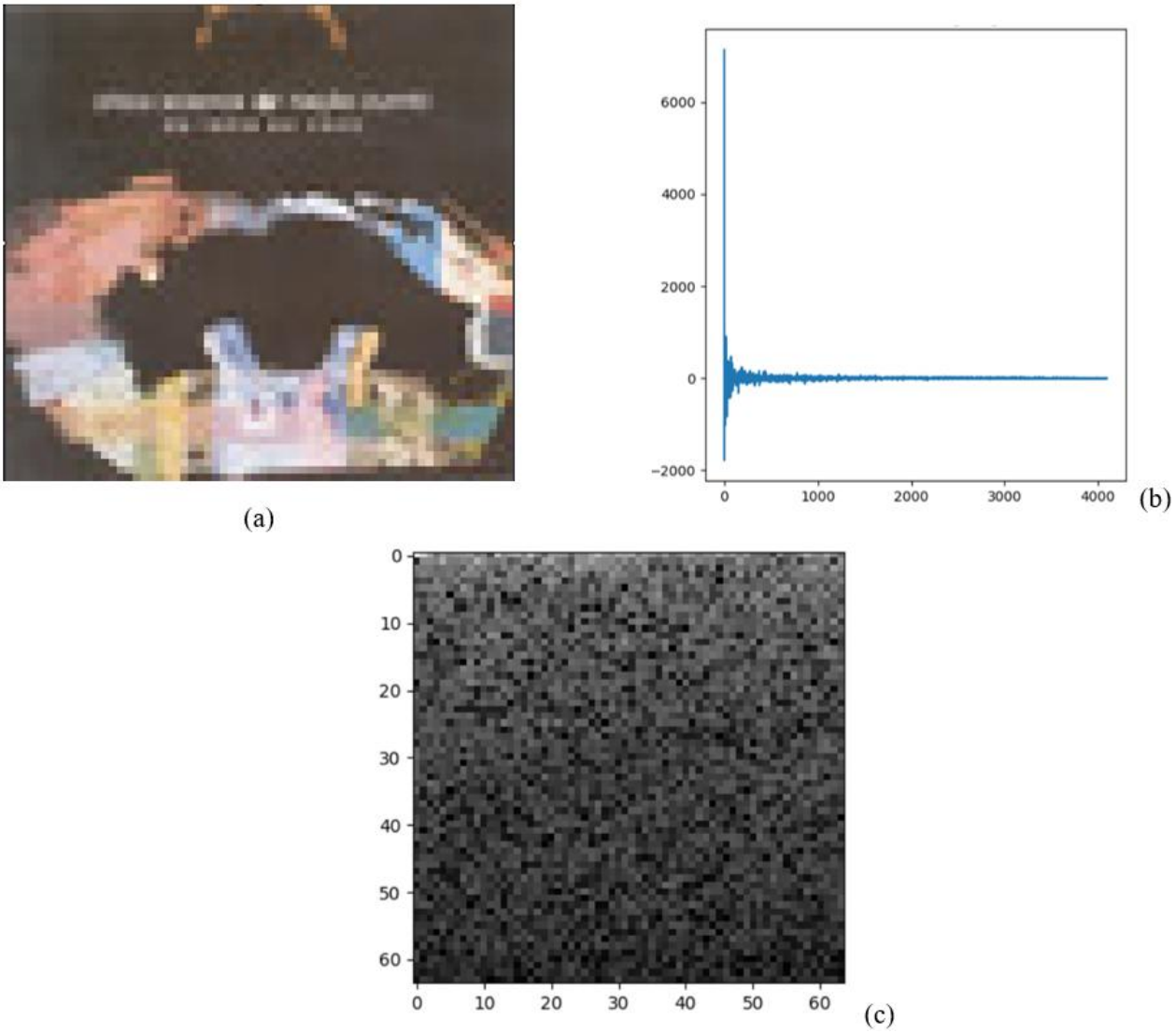
Figura 8 - (a) Um sinal contínuo e (b) seu espectro de Fourier.



Fonte: Lathi, 2014.

A Figura 9(b) mostra uma simplificação unidimensional dos coeficientes de uma imagem após a utilização da transformada discreta do cosseno em duas dimensões (2D-DCT) na ordenação *zig-zag*, fazendo com que a maioria dos coeficientes se encontre nas baixas frequências. Um conceito parecido é notado para sinais bidimensionais, porém, é importante destacar que em espectros bidimensionais, as baixas frequências costumam se localizar na parte superior esquerda da imagem. Um exemplo de espectro bidimensional é mostrado na Figura 9(c).

Figura 9 - (a) A imagem de entrada (uma versão reescalonada em 64x64 da capa do álbum “Da Lama Ao Caos” de Chico Science & Nação Zumbi), (b) coeficientes 1D e (c) 2D da imagem usando a 2D-DCT ordenando os coeficientes no formato *zig-zag*.



Fonte: elaborado pelo próprio autor.

2.1.2 A transformada discreta do cosseno em duas dimensões (2D-DCT)

O JPEG (*Joint Photographic Experts Group* – Grupo Conjunto de Especialistas em Fotografia) é o padrão comumente usado em compressões de imagens digitais. A transformada discreta do cosseno em duas dimensões (2D-DCT) é um dos métodos utilizados pelo JPEG para compressão de imagens. (Neelamani *et al*., 2006)

A 2D-DCT é uma extensão direta da transformada discreta do cosseno em uma dimensão (1D-DCT) e é dada por:

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{\pi(2x+1)u}{2N}\right]cos\left[\frac{\pi(2y+1)v}{2N}\right] \quad (3)$$

e $\alpha(u)$ e $\alpha(v)$ são definidos pela equação:

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{Para } u = 0 \\ \sqrt{\frac{1}{2N}} & \text{Para } u \neq 0 \end{cases}, \quad (4)$$

e sua transformada inversa é dada pela expressão (Khayam, 2003):

$$f(x,y) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v)cos\left[\frac{\pi(2x+1)u}{2N}\right]cos\left[\frac{\pi(2y+1)v}{2N}\right]. \quad (5)$$

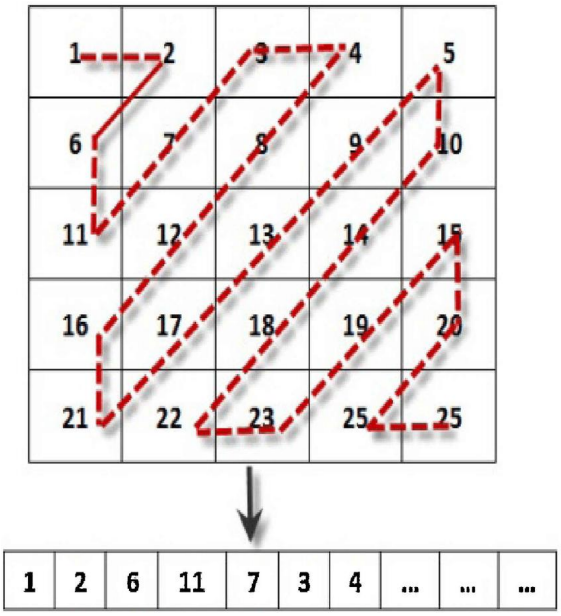
Na prática, a aplicação dessa transformada pelo JPEG em uma imagem segue um processo bem estruturado. De antemão, a imagem não é processada por completo. Ela passa por uma subdivisão em blocos de 8×8 pixels, pois a imagem tende a ser mais uniforme em áreas pequenas.

Para imagens com profundidade de 8 bits, os valores dos pixels que eram de 0 a 255 passam por uma subtração de 128, passando a ter uma faixa entre -128 e 127, ajudando a diminuir a magnitude dinâmica necessária para os cálculos da DCT. Após isso, aplica-se a 2D-DCT aos blocos 8×8. O que acontece na prática quando se aplica a transformada é que os coeficientes de baixa frequência representam as cores e a iluminação geral, já os de alta frequência representam detalhes e texturas finas. O resultado é uma nova matriz 8×8 de coeficientes semelhantes aos da Figura 9(c).

A compressão é então feita, dividindo os valores de alta frequência por números muito altos, anulando-os (ou reduzindo a zero), uma vez que o sistema visual humano não é capaz de detectar detalhes dessa magnitude de frequência.

O padrão de ordenação "*zig-zag"* faz parte do CODEC de JPEG. Ele serve para que os coeficientes sejam ordenados de tal forma que faça com que a compressão da imagem seja possível de ser desfeita corretamente no destino final. A Figura 10 ilustra o funcionamento do padrão de *zig-zag*.

Figura 10 - Ilustração do funcionamento do método de zig-zag.



Fonte: Singh, Ranade e Singh, 2016.

2.2 IMAGENS DE *LIGHT FIELD*

As imagens de *light field* representam uma tecnologia que permite capturar uma quantidade maior de informações em relação a uma imagem comum. As imagens de LF não se baseiam somente em informações de duas dimensões espaciais (*x, y)* e comprimento de onda (λ) como em imagens comuns, mas sim, inspiradas em funções plenópticas, em componentes de posição (*x, y, z*), ângulo (θ, ϕ) e comprimento de onda (λ) (Fang, 2025).

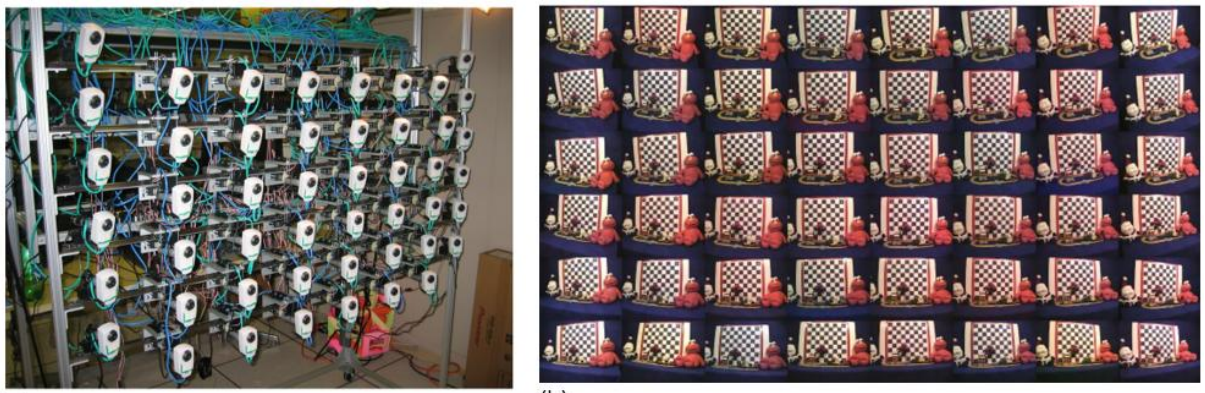
A grande quantidade de informação capturada na aquisição de uma imagem de *light field* amplia a percepção humana sobre mídias digitais, possibilitando uma experiência mais completa e realista em imagens (Pereira; Silva; Lafruit, 2018).

Essa tecnologia consegue captar raios de luz em várias direções diferentes. Isso permite que, ao observar esse tipo de imagem, se tenha uma maior percepção de profundidade, fazendo com que seja possível aplicar funções que uma imagem 2D comum não seria capaz de proporcionar.

Por exemplo, no pós-processamento de uma imagem de *light field* é possível inserir artificialmente objetos em uma cena com noção de profundidade. Também é possível focar em diferentes áreas da cena mesmo após a captura e o processamento da imagem. Em vídeos (que são basicamente uma sequência de imagens no tempo) é possível aplicar estabilização de imagem. As aplicações são diversas.

Imagens de *light field* podem ser capturadas de algumas formas. Os métodos mais comuns consistem em *arrays* (matrizes) de câmeras ou de microlentes. O método de captura por um conjunto de câmeras é o mais intuitivo. A Figura 11 mostra um desses arranjos de câmeras e a imagem de *light field* obtida por ele. Todas as câmeras capturam a cena no mesmo momento.

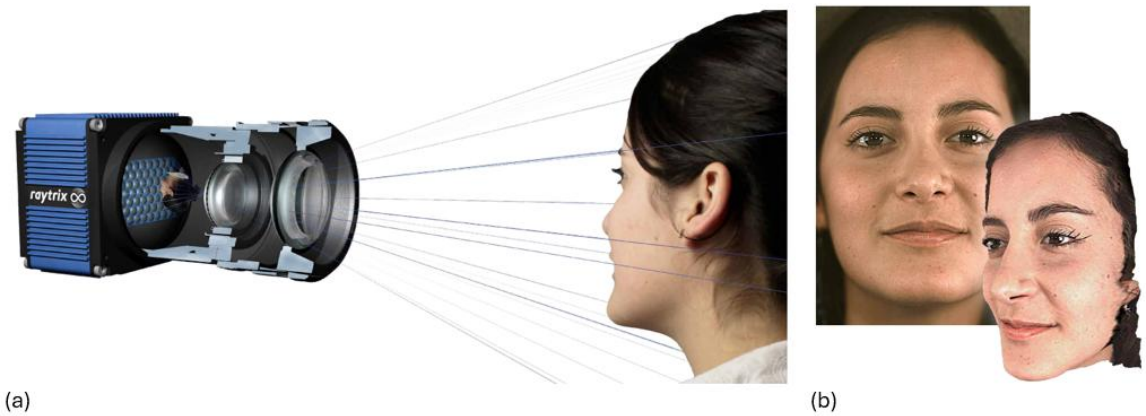
Figura 11 - Exemplo de (a) um array de câmeras utilizado para capturar imagens de *light field* e (b) uma imagem capturada por este tipo de sistema.



Fonte: Zhang e Chen, 2004.

Aplicações mais práticas envolvem câmeras específicas para imagens plenópticas, as quais incluem em sua arquitetura interna um *array* de microlentes capaz de capturar pequenos detalhes de uma cena através dos raios de luz capturados por uma lente principal. A Figura 12 exibe um exemplo de uma câmera de microlentes da marca Raytrix e uma imagem capaz de ser gerada por ela. (Raytrix GMBH, 2011)

Figura 12 - (a) Uma modelo sendo capturada por uma câmera de microlentes Raytrix e (b) um exemplo de visualização 3D da imagem pós-processada.

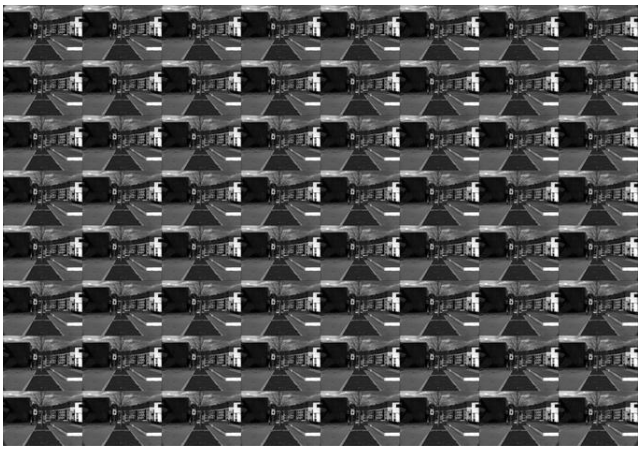


Fonte: Raytrix Gmbh[1].

[1] Imagem retirada do material de divulgação da Raytrix GmbH (2011). Disponível em: https://raytrix.de/wp-content/uploads/software/Raytrix_Deutsches_Museum_Poster.pdf. Acesso em: 23 abr. 2026.

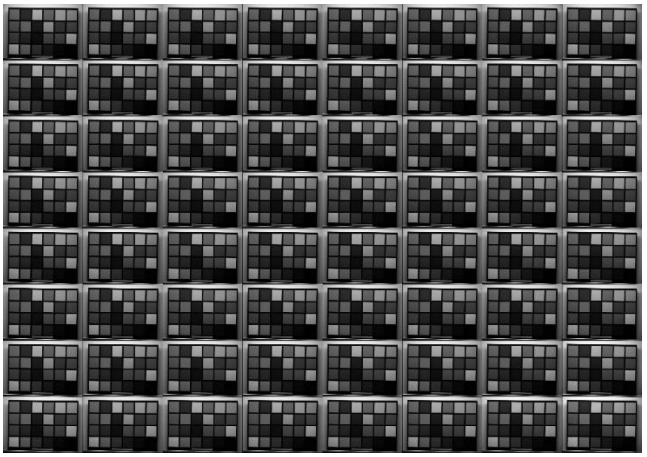
Uma maneira comum de representar imagens de *light field* é por meio de uma matriz de imagens. As Figuras 13, 14 e 15 mostram esse tipo de representação. Essas imagens também serão usadas para os testes deste trabalho.

Figura 13 - Uma imagem de *light field* em escala de cinza de uma rua com prédios.



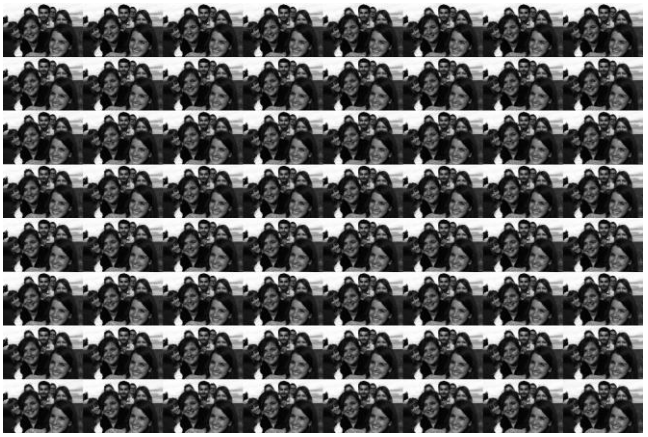
Fonte: elaborado pelo próprio autor.

Figura 14 - Uma imagem de *light field* em escala de cinza de uma tabela de cores.



Fonte: elaborado pelo próprio autor.

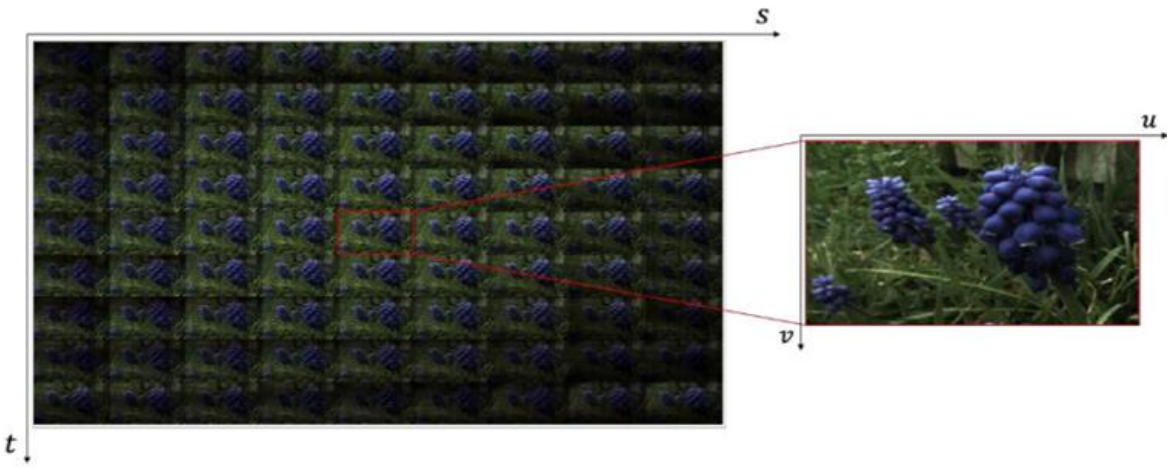
Figura 15 - Uma imagem de *light field* em escala de cinza de um grupo de pessoas.



Fonte: elaborado pelo próprio autor.

Para se referenciar matematicamente a imagens de *light field*, pode-se usar um sistema de coordenadas quadridimensionais (s, t, u, v), em que duas dimensões são usadas para as coordenadas dos pixels em uma subimagem (u, v) e duas dimensões são referentes às coordenadas das subimagens em uma imagem de *light field* completa (s, t). A Figura 16 mostra um exemplo desse sistema de coordenadas. (Ferreira, 2020)

Figura 16 - Visualização de todas as vistas de uma imagem de *light field* capturada por um dispositivo com microlentes (*lenslet*), em que cada vista é referenciada por um sistema de coordenadas (*s, t*), e a vista central expandida, em que cada pixel é referenciado por um sistema de coordenadas (*u, v*).



Fonte: Ferreira, 2020.

2.3 TEORIA DOS GRAFOS

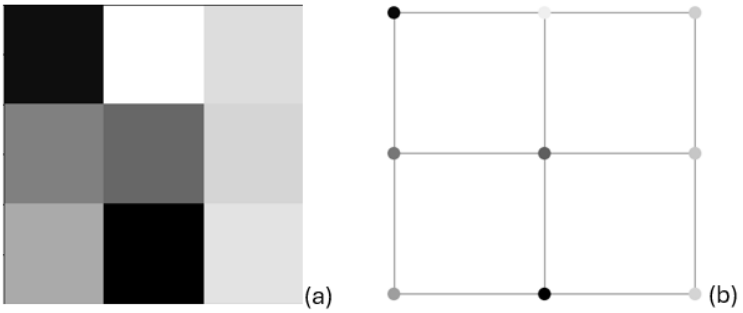
Um grafo pode ser definido como um conjunto finito não-nulo de vértices e um conjunto de pares de vértices. Esses pares são chamados de arestas. Um vértice (ou pixel) p_i é dito adjacente a um vértice p_j quando $(p_i, p_j)\ \epsilon\ \boldsymbol{A}$, onde $\boldsymbol{A}$ é a matriz de adjacência.

A matriz de adjacência é uma matriz $i \times j$ cujo tamanho varia em relação a quantidade de vértices de um grafo, portanto, como um grafo pode ser uma representação em nós de uma imagem, o tamanho da matriz $\boldsymbol{A}$ dependerá da resolução da imagem (a quantidade de pixels).

Em termos matemáticos, é possível representar um grafo através da matriz de adjacência ($\boldsymbol{A}$) ou da matriz de pesos ($\boldsymbol{W}$). Na matriz de adjacência, é usado "1" para indicar a presença de uma aresta e "0" para indicar que a aresta não existe entre os vértices. A matriz de pesos, além de indicar se existe aresta, também indica a força dessa ligação.

Ao considerar uma imagem digital como um grafo onde cada pixel é um vértice, é possível obter funções muito úteis, como, por exemplo, aplicar a GFT em imagens. A Figura 17 mostra um exemplo da aplicação desse conceito.

Figura 17 - (a) Um exemplo de imagem e (b) sua representação como grafo.



Fonte: elaborado pelo próprio autor.

As topologias de grafo são as maneiras como vértices se ligam a outros por meio de arestas em um grafo. Assim, a um grafo qualquer pode-se aplicar diversas topologias.

2.4 A TRANSFORMADA DE FOURIER SOBRE GRAFOS (GFT)

A Transformada de Fourier sobre Grafos é uma generalização da Transformada de Fourier clássica. Em vez de ser aplicada sobre domínios contínuos ou uniformes, é aplicada em sinais definidos em vértices de um grafo.

Para um grafo $G = (V, E)$, onde V é o conjunto de vértices e E é o conjunto de arestas, sendo $f: V \rightarrow \mathbb{R}$ um sinal definido nos vértices do grafo.

A matriz laplaciana do grafo é $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, onde $\boldsymbol{D}$ é a matriz diagonal das valências dos vértices e $\boldsymbol{A}$ a matriz adjacência. A decomposição espectral da matriz laplaciana resulta em $\boldsymbol{L} = \boldsymbol{U\Lambda U^T}$, onde $\boldsymbol{U}$ é a matriz cujas colunas são autovetores u_i e $\boldsymbol{\Lambda}$ é a matriz diagonal cujas entradas λ_i são os autovalores correspondentes. (Shuman *et al.*, 2013)

A GFT de um sinal f é dada por $\hat{f} = U^T f$ ou na forma

$$\hat{f}(\lambda_k) = \sum_{i=1}^{N} f(v_i) u_k(v_i). \tag{6}$$

A transformada inversa da transformada de Fourier sobre grafos (IGFT) é dada por $f = \boldsymbol{U}\hat{f}$ ou conforme,

$$f(v_i) = \sum_{i=1}^{N} \hat{f}(\lambda_k) u_k(v_i), \tag{7}$$

em que $\hat{f}(\lambda_k)$ é a componente espectral correspondente ao autovalor λ_k, $f(v_i)$ é o valor do sinal f no vértice v_i e $u_k(v_i)$ é a $i-$ésima entrada do $k-$ésimo autovetor u_k da matriz laplaciana $\boldsymbol{L}$. (Shuman *et al.*, 2013)

Um conceito fundamental para a análise de desempenho da GFT é a suavidade do sinal sobre o grafo (*graph signal smoothness*). Matematicamente, a suavidade de um sinal f mapeado em um grafo é quantificada pela sua variação total através da forma quadrática da matriz laplaciana ($\boldsymbol{L}$), conforme expresso na Equação 8. (Shuman *et al.*, 2013)

$$Variação\ Total_G(f) = f^T \boldsymbol{L} f = \frac{1}{2}\sum_{i,j} A_{i,j}\left(f_i - f_j\right)^2. \tag{8}$$

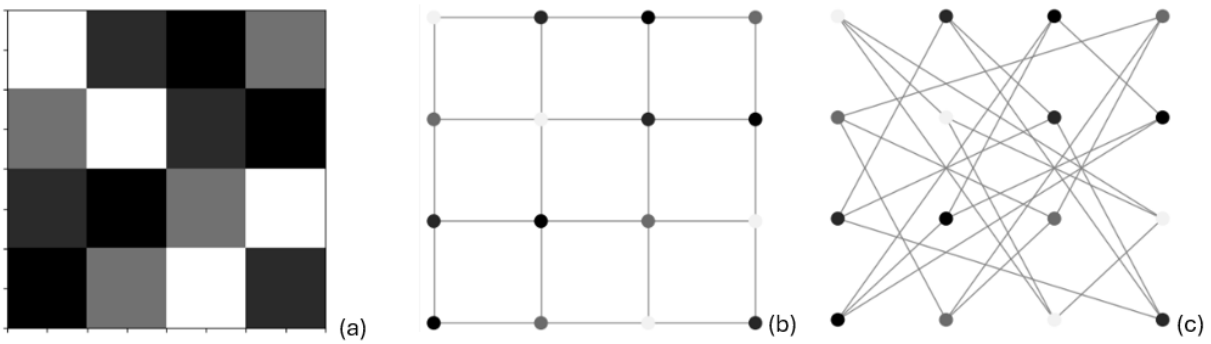
Onde $A_{i,j}$ representa os elementos da matriz de adjacência. A minimização dessa forma quadrática indica que o sinal é suave, ou seja, vértices fortemente conectados possuem valores de intensidade muito semelhantes. A importância prática da suavidade está no fato de que a GFT projeta o sinal nos autovetores de $\boldsymbol{L}$. Se um sinal manifesta alta suavidade sobre a topologia, a transformada consegue concentrar a maior parte da energia em poucos coeficientes associados às baixas frequências espectrais do grafo. Esse fenômeno gera uma representação esparsa, ideal para algoritmos de compressão de dados. (Shuman *et al.*, 2013)

2.4.1 Como a topologia influencia na distribuição dos coeficientes

Para demonstrar como a topologia usada influencia diretamente a distribuição dos coeficientes após o cálculo da GFT, vamos tomar como exemplo a imagem mostrada na Figura 18(a).

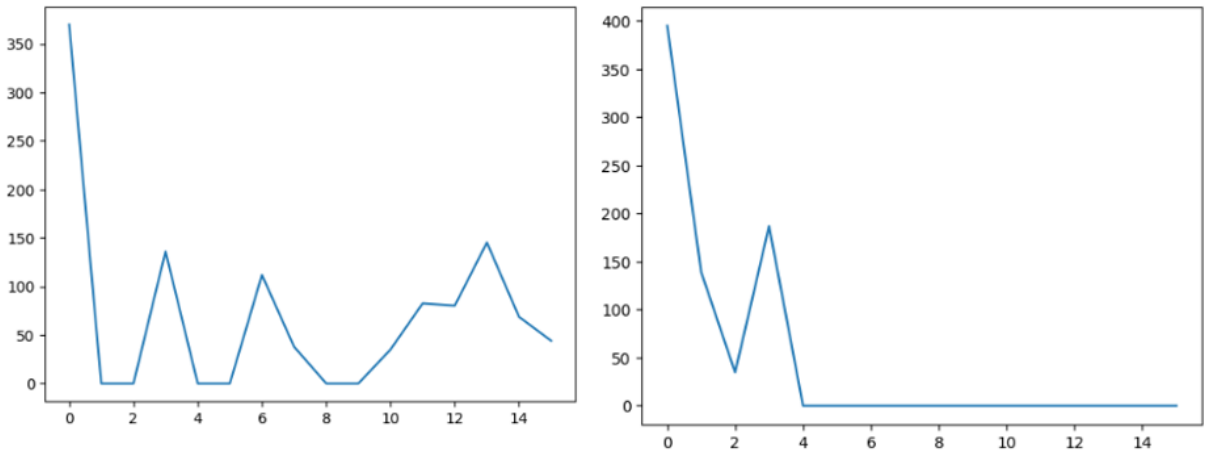
Ao realizar o cálculo da GFT para duas topologias diferentes, uma onde todos os pixels se ligam aos pixels ao redor (Figura 18(b)) e outra onde os pixels são ligados aos pixels iguais a eles (Figura 18(c)). É possível obter os gráficos mostrados na Figura 19 após a utilização da GFT.

Figura 18 - (a) imagem de referência, (b) topologia que liga pixels ao redor e (c) topologia que liga pixels de mesma intensidade.



Fonte: elaborado pelo próprio autor.

Figura 19 – Gráfico dos coeficientes da GFT da Figura 18 usando (a) a topologia que lida pixels ao redor e (b) a topologia que liga pixels de mesma intensidade.



Fonte: elaborado pelo próprio autor.

Na análise da Figura 19, pode-se notar que a topologia que se utiliza da semelhança entre os pixels consegue acumular mais coeficientes nas baixas frequências. Ao aplicar um filtro passa-baixas nos coeficientes, a topologia genérica perderia mais coeficientes e isso influenciaria a qualidade da imagem reconstruída. Neste trabalho, escolher a melhor topologia passa por entender a suavidade de sinal gerado e observar seu impacto prático por meio de métricas de qualidade de imagem digital.

2.5 MÉTRICAS DE QUALIDADE E COMPRESSÃO DE IMAGEM

O PSNR (*Peak Signal-to-noise Ratio* – Relação Sinal-Ruído de Pico) e o SSIM (*Structural Similarity Index* – Índice de Similaridade Estrutural) são as duas métricas mais utilizadas para medir a qualidade de uma imagem reconstruída ou comprimida em relação a uma imagem original.

O PSNR mede a relação entre a potência máxima do sinal e a potência do ruído e se baseia puramente na diferença pixel a pixel. A relação se utiliza da MSE (*Mean Squared Error* - Média dos quadrados dos erros) para fazer uma relação logarítmica dada na equação:

$$PSNR = 10 \times \log_{10}(\frac{MAX_I^2}{MSE}). \quad (9)$$

Na qual MAX_I é o valor máximo possível do pixel. Para as aplicações deste trabalho, $MAX_I = 2^8 = 255$. (Horé; Ziou, 2010)

Diferentemente do PSNR, focado na intensidade numérica dos pixels, o SSIM é projetado para ser mais próximo da visão humana, entendendo que o olho humano é sensível a mudanças na estrutura, brilho e contraste. (Horé; Ziou, 2010)

O SSIM compara duas janelas x e y de uma imagem e se baseia em três pilares: luminância (μ - média), contraste (σ^2- variância) e estrutura (σ_{xy} - covariância). A fórmula simplificada é dada por:

$$SSIM_{(x,y)} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2\mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{10}$$

sendo C_1 e C_2 constantes para evitar divisão por zero. O SSIM varia de 0 a 1, sendo valores próximos a 1 uma imagem fiel a original e índices próximos a 0 imagens pouco similares.

Além das métricas voltadas para a distorção visual, a avaliação completa de um sistema de compressão exige a análise de sua eficiência em reduzir o volume de dados. Neste trabalho, será utilizada a entropia de Shannon como uma métrica teórica para determinar o limite máximo de compressão alcançado pelos algoritmos.

Proposta por Claude Shannon, a entropia quantifica o nível de incerteza ou a quantidade média de informação contida em uma fonte de dados, sendo usualmente medida em bits por símbolo. A entropia $H(x)$ de uma matriz discreta de dados é dada conforme,

$$H(x) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i). \tag{11}$$

Nessa expressão, $P(x_i)$ representa a probabilidade de ocorrência de um determinado coeficiente x_i no conjunto de dados, e n é o número de valores únicos possíveis. (Shannon, 1948)

No contexto da GFT, a etapa de quantização aumenta drasticamente a repetição de valores (zeros) na matriz. Esse aumento da previsibilidade dos dados faz com que a entropia de Shannon decaia. Portanto, o valor da entropia atua como um indicativo matemático do quão pequeno será o arquivo se submetido a um codificador, permitindo traçar uma curva entre a taxa de compressão teórica (entropia) e a qualidade visual (PSNR ou SSIM).

3 METODOLOGIA DE COMPRESSÃO E TOPOLOGIAS PROPOSTAS

Esta seção discorre sobre o método de divisão estrutural dos blocos das imagens de *light field* e formaliza as topologias de grafos propostas para a aplicação da GFT.

3.1 DIVISÃO DE BLOCOS

Aplicar a GFT em imagens de alta resolução pode mostrar-se um desafio significativo, sendo, em muitos casos, impossível. Conforme discutido na seção 2.4, para aplicar a GFT em uma imagem é necessário definir uma matriz de adjacência, cujo número de elementos é definido pelo número de pixels ao quadrado.

Para uma imagem como a Figura 13 (que será usada nos testes desse trabalho) cujas dimensões são de $8 \times 8 \times 624 \times 432$, a matriz de adjacência resultante possui aproximadamente $2{,}98\text{x}10^{14}$ elementos. Considerando que cada elemento é um número inteiro, para armazenar essa matriz, seriam necessários 298 TB (terabytes) de memória RAM (*Random Access Memory* - Memória de Acesso Aleatório). Essa magnitude excede as capacidades de processamento convencionais, tornando isso impraticável.

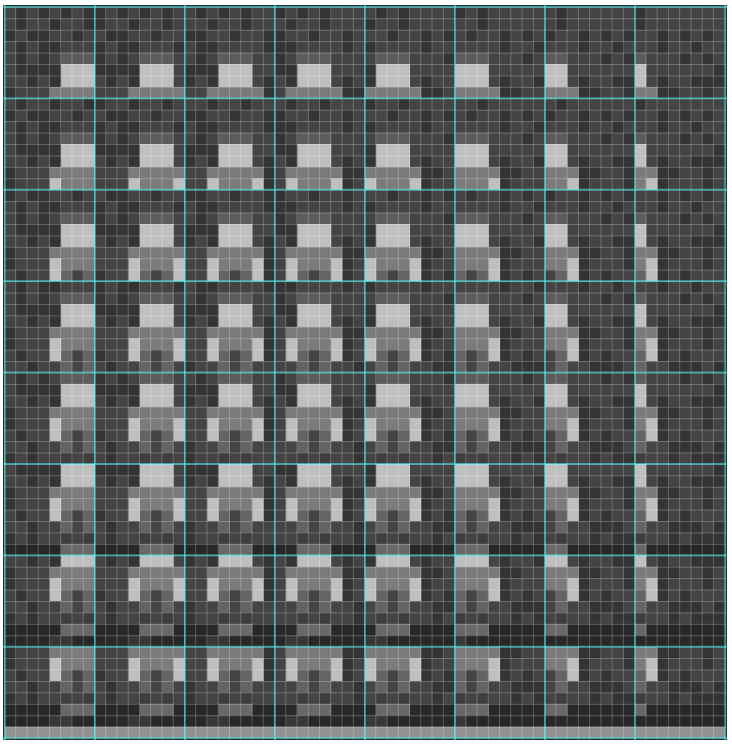
Esse problema pode ser contornado utilizando uma ideia parecida com o que é feito em compressões JPEG, dividindo as imagens em pedaços menores.

A Figura 20 é uma *pixel art* considerando variações sutis no ângulo para assemelhar-se a uma imagem de *light field*, possibilitando uma visualização mais simples das topologias que serão apresentadas.

Para este trabalho, as imagens serão capturadas de forma que se crie uma imagem de *light field* recortada de tamanho $8 \times 8 \times 8 \times 8$, ou seja, 64 subimagens, onde cada uma delas tem o tamanho 8×8. A Figura 21 mostra um exemplo dessa captura de área, onde em Figura 21(a) os pixels marcados em verde são capturados e juntos formam uma imagem alternativa mostrada em Figura 21(b).

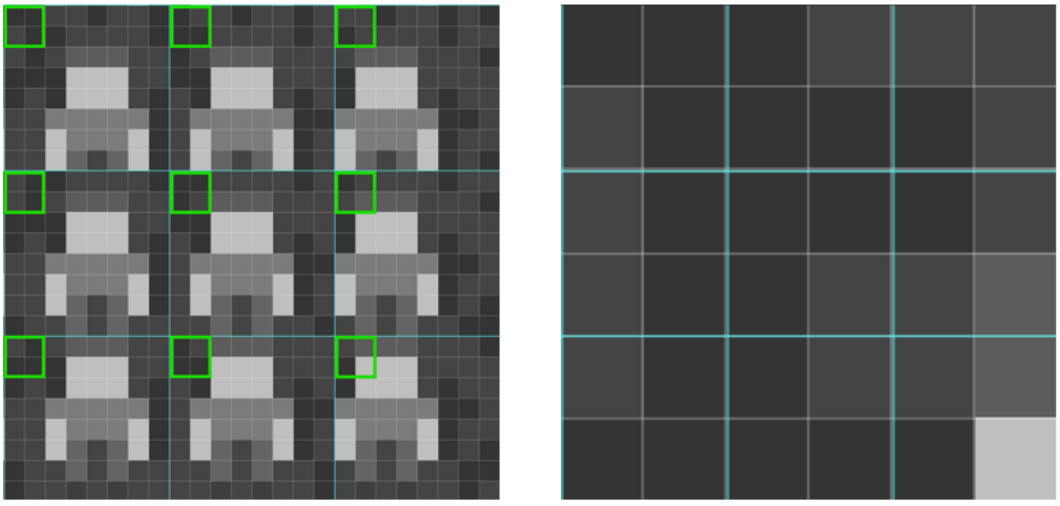
Em imagens JPEG, a compressão é feita a partir da 2D-DCT utilizando blocos 8x8.

Figura 20 – *Pixel art* criada para ilustrar uma imagem de *light field*.



Fonte: elaborado pelo próprio autor.

Figura 21 - (a) Mostrando a área capturada em cada subimagem e (b) a imagem resultante do processo de captura das áreas.



(a) (b)

Fonte: elaborado pelo próprio autor.

3.2 TOPOLOGIAS

Para a construção das topologias propostas, cada sinal contido no bloco quadridimensional de *light field* é mapeado como um vértice (ou pixel) p_i pertencente ao conjunto de vértices V do grafo $G(V, E)$. Cada vértice é formalmente indexado por suas

coordenadas espaciais (u, v) e angulares (s, t), sendo definido como o vetor de coordenadas $p_i = (u_i, v_i, s_i, t_i)$. A existência de uma aresta $e_{ij} \in E$ que conecta dois vértices distintos p_i e p_j é governada por regras estruturais específicas para cada topologia. As notações são feitas utilizando lógica proposicional[2].

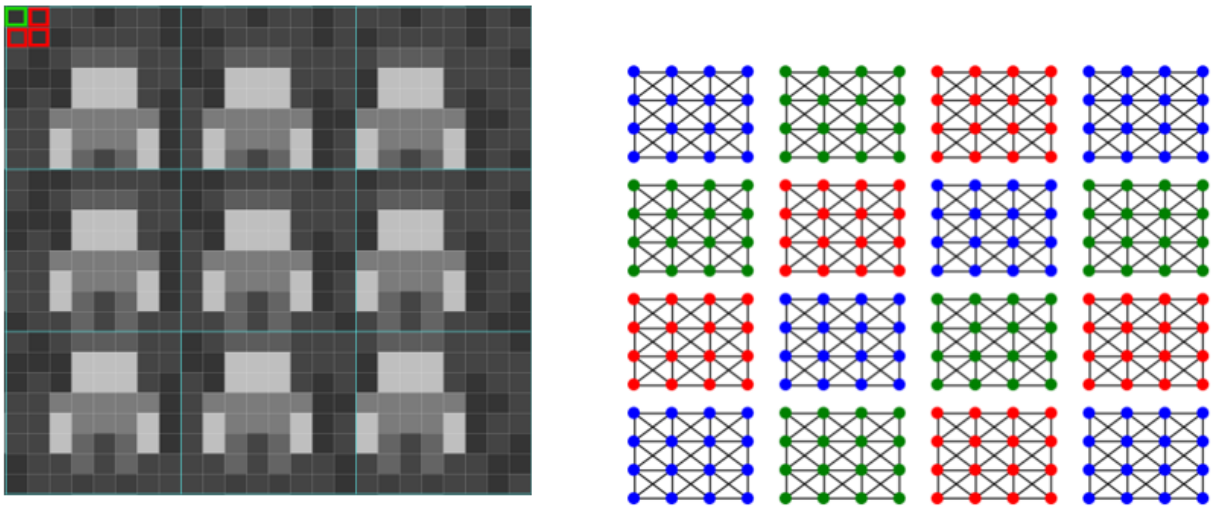
3.2.1 Pixel Vizinho

A topologia de Pixel Vizinho se utiliza da grande probabilidade dos pixels ao redor serem de intensidades semelhantes. Nela, um determinado pixel é considerado adjacente a todos os pixels que estão imediatamente próximos. Esse conceito também pode ser aplicado em áreas onde, ao invés de se ligar um pixel apenas aos seus vizinhos imediatos, liga-o a vários pixels em um raio.

Essa topologia pode ser usada de forma mais eficaz em imagens comuns (bidimensionais), onde semelhanças de intensidade de cor ocorrem nas proximidades de um pixel específico. Para imagens de *light field*, no entanto, podem ocorrer picos de energia na variação de uma subimagem para outra.

A Figura 22(a) é um exemplo da topologia de pixel vizinho. O pixel marcado em verde é ligado aos pixels em vermelho, os pixels ao seu redor. Isso se repete para todos os pixels em toda a imagem.

Figura 22 - (a) Exemplo discreto para a topologia do pixel vizinho e (b) exemplo da topologia usada em Python.



Fonte: elaborado pelo próprio autor.

[2] A formalização utiliza notação de lógica proposicional, na qual o operador ↔ denota equivalência lógica (se, e somente se), o operador ∧ denota a conjunção (e) e o operador ∨ denota a disjunção (ou).

Para este trabalho, as conexões são restritas ao domínio espacial interno de uma mesma subimagem, conforme mostrado na Figura 22(b). Isso faz com que a topologia seja mais otimizada para imagens LF. Dois vértices p_i e p_j estabelecem uma ligação se, e somente se, pertencerem à mesma coordenada angular e forem vizinhos espaciais diretos pela distância de Chebyshev, conforme,

$$e_{ij} \in E \leftrightarrow (s_i = s_j) \wedge (t_i = t_j) \wedge [\max(|u_i - u_j|, |v_i - v_j|) = 1]. \quad (12)$$

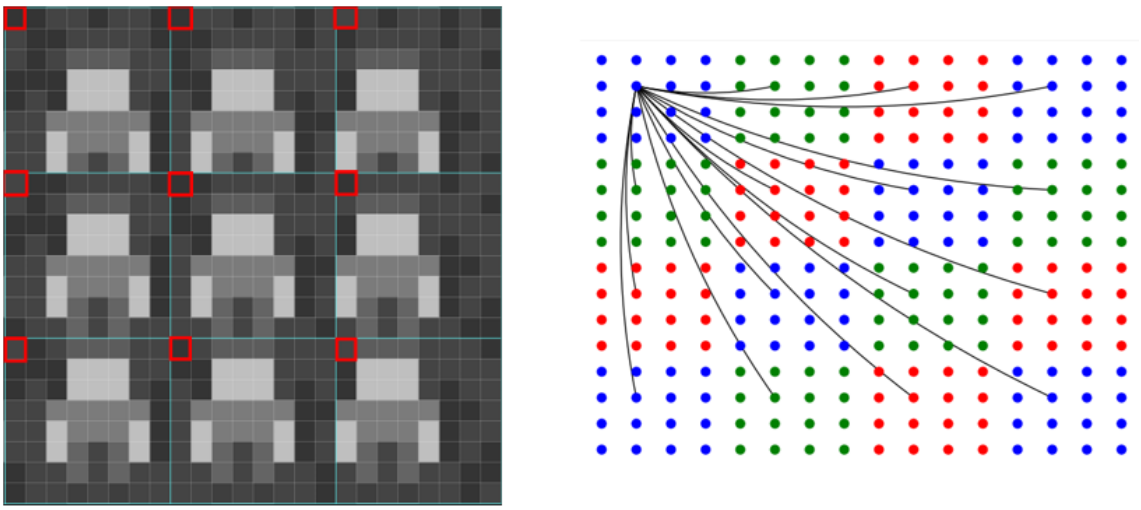
3.2.2 Pixel Gêmeo

Na topologia do Pixel Gêmeo, considera-se que, para cada pixel em uma subimagem, existe um de intensidade semelhante em outra.

A Figura 23(a) mostra um exemplo da topologia do pixel gêmeo, onde todos os pixels marcados em vermelho estão ligados entre si, e isso ocorre para todos os outros pixels da imagem LF.

Neste exemplo específico, a topologia não parece ligar pixels parecidos. Isso ocorre devido à baixa resolução e ao fato de ser uma imagem criada manualmente. Imagens de *light field* reais, normalmente, possuem uma resolução muito superior e variações mais sutis na cor dos pixels, o que pode tornar esta topologia eficaz.

Figura 23 - (a) Exemplo discreto para a topologia do pixel gêmeo e (b) exemplo da topologia usada em Python.



(a) (b)

Fonte: elaborado pelo próprio autor.

As conexões ignoram a vizinhança espacial local e focam exclusivamente na redundância entre perspectivas. Dois vértices p_i e p_j são conectados se ocuparem a mesma posição de pixel (u, v), mas estiverem localizados em outras subimagens no plano angular, seguindo:

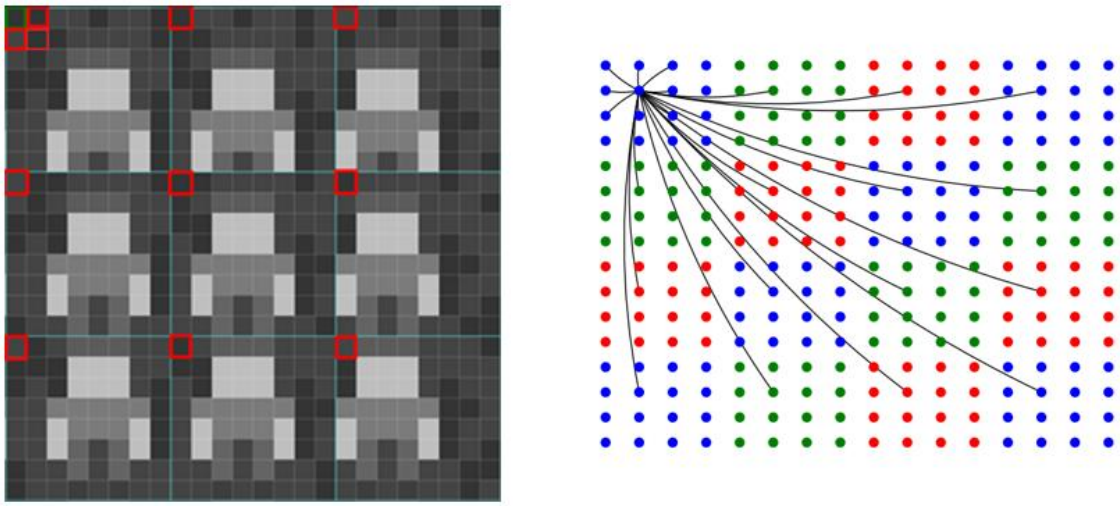
$$e_{ij} \in E \leftrightarrow [(s_i \neq s_j) \vee (t_i \neq t_j)] \wedge (u_i = u_j) \wedge (v_i = v_j). \quad (13)$$

3.2.3 Pixel vizinho + Pixel gêmeo

Esta topologia é importante para mostrar um conceito interessante. Juntando as duas topologias abordadas nas Seções 3.2.1 e 3.2.2, é possível observar uma topologia diferente que une as duas, então, além de se ligar um pixel ao seu pixel gêmeo, também o liga aos vizinhos imediatamente próximos a ele.

Na Figura 24, o pixel marcado em verde faz ligações com os pixels marcados em vermelho.

Figura 24 - (a) Exemplo discreto para a topologia do pixel vizinho + pixel gêmeo e (b) exemplo da topologia usada em Python.



Fonte: elaborado pelo próprio autor.

Esta topologia é regida pela lógica expressa em

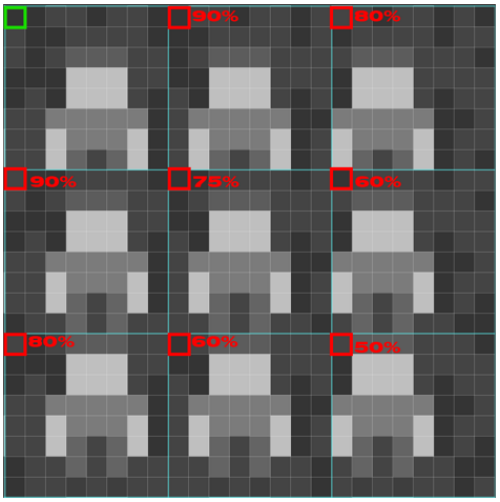
$$e_{ij} \in E \leftrightarrow \{(s_i = s_j) \wedge (t_i = t_j) \wedge [\max(|u_i - u_j|, |v_i - v_j|) = 1]\} \vee \{[(s_i \neq s_j) \vee (t_i \neq t_j)] \wedge (u_i = u_j) \wedge (v_i = v_j)\}. \quad (14)$$

3.2.4 Pixel gêmeo proporcional

Diferentemente da matriz de adjacência, na matriz de pesos é possível definir o quão forte é uma aresta. Assim, em vez de definir uma adjacência com peso “1”, pode-se fracionar este valor. Uma topologia baseada nesse conceito pode ser mais precisa, uma vez que é possível definir mais assertivamente a chance que dois pixels têm de ter intensidades de cor parecidas baseando-se na distância entre eles.

A Figura 25 mostra um exemplo para essa topologia. O pixel em verde está sendo ligado aos pixels em vermelho e a porcentagem ao lado de cada pixel é um valor definido considerando a distância das subimagens. É possível aumentar a precisão das porcentagens calculando a distância euclidiana das subimagens ou até entre os próprios pixels. Essa topologia gera uma matriz de pesos, como explicado na seção 1.1.

Figura 25 - Exemplo para a topologia do pixel gêmeo proporcional.



Fonte: elaborado pelo próprio autor.

Neste trabalho, o peso da aresta foi calculado para cada uma das ligações usando a Equação 15.

$$W = \exp\left(\frac{-dist(p_i, p_j)}{2\theta^2}\right) \tag{15}$$

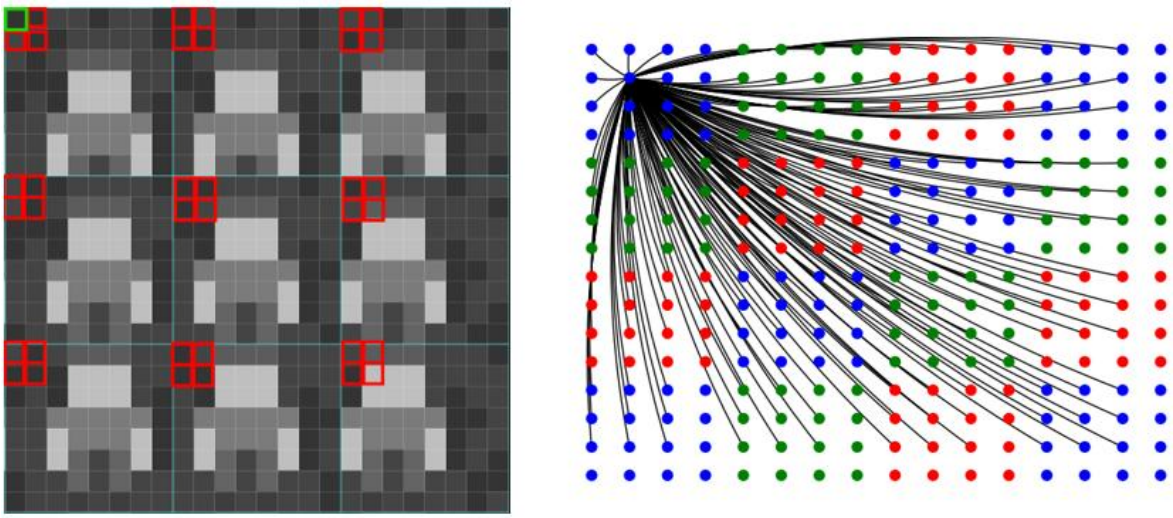
Onde, $dist(p_i, p_j)$ é a distância euclidiana entre os vértices p_i e p_j e θ é um parâmetro usado para controlar a distribuição. Neste trabalho, θ foi fixado em 5. Este valor foi definido de forma empírica e demonstrou ser estável para manter uma conectividade proporcional adequada em blocos de dimensões $8 \times 8 \times 8 \times 8$, já que valores de θ muito baixos fariam com que conexões perdessem força rapidamente, e valores excessivamente altos anulariam a proporcionalidade das distâncias.

3.2.5 Vizinhança gêmea

É possível ampliar a probabilidade de ligar pixels com a mesma intensidade de cor ligando toda uma vizinhança de pixels.

Na Figura 26(a), o pixel marcado em verde se liga a todos os pixels marcados em vermelho. Para uma imagem com resolução maior, é provável que essas semelhanças de intensidade de cor sejam maiores.

Figura 26 - (a) Exemplo discreto para a topologia da vizinhança gêmea e (b) exemplo da topologia usada em Python.



(a) (b)

Fonte: elaborado pelo próprio autor.

Dois pixels p_i e p_j estão conectados se, e somente se, possuírem uma distância espacial de Chebyshev menor ou igual a 1, independentemente do plano angular. Ou seja, a topologia conecta o pixel de origem aos seus vizinhos espaciais locais, aos seus pixels gêmeos e aos vizinhos dos pixels gêmeos em todas as perspectivas do LF, conforme:

$$e_{ij} \in E \leftrightarrow \left[\max\left(\left|u_i - u_j\right|, \left|v_i - v_j\right|\right) \leq 1\right] \wedge \left(p_i \neq p_j\right). \quad (16)$$

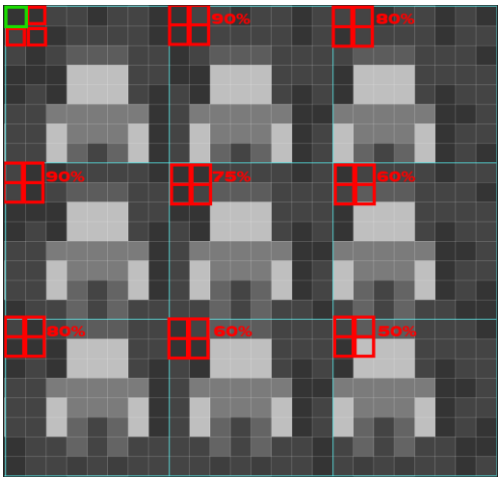
3.2.6 Vizinhança gêmea proporcional

Unindo as vantagens das topologias das seções 3.2.4 e 3.2.5, é possível desenvolver uma nova topologia. Esta também gera uma matriz de pesos, porém não liga cada pixel da imagem, e sim uma vizinhança de pixels de forma proporcional à distância das subimagens e dos pixels.

A Figura 27 mostra um exemplo para essa topologia usando valores arbitrários para as porcentagens. O pixel marcado em verde se liga a todos os pixels marcados em vermelho, onde seus vizinhos imediatos possuem porcentagem de 100%, enquanto as outras vizinhanças possuem porcentagens indicadas na imagem. Assim como a topologia do pixel gêmeo proporcional, é possível usar a distância euclidiana das subimagens para aumentar a assertividade da porcentagem.

Neste trabalho, essa topologia foi usada considerando a distância de todos os pixels, inclusive os vizinhos do pixel em questão. Para cada ligação, foi calculado o peso usando a Equação 15.

Figura 27 - Exemplo para a topologia da vizinhança gêmea proporcional.

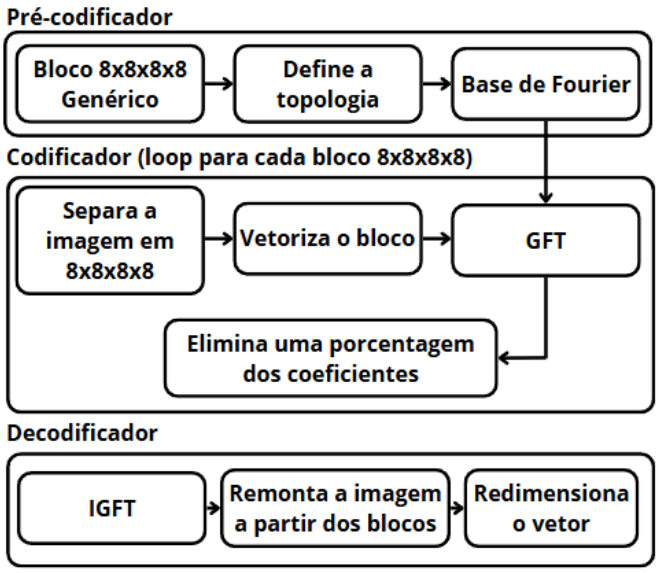


Fonte: elaborado pelo próprio autor.

3.3 FLUXO DE CODEC

O fluxo de operação do sistema, denominado *CODEC* (Codificador-Decodificador), foi estruturado para o processamento de imagens de *light field*. O processo completo é ilustrado na Figura 28, cujas etapas de codificação e decodificação são detalhadas a seguir.

Figura 28 - Fluxograma do CODEC elaborado.



Fonte: elaborado pelo próprio autor.

3.3.1 Codificação

Na etapa de codificação, a imagem de *light field* é interpretada como um grafo cujas arestas são definidas a partir da escolha de uma topologia entre as apresentadas.

A partir disso, o sistema computa a base de Fourier, onde são calculados os autovetores e autovalores da matriz laplaciana do grafo (definida, em partes, pela matriz de adjacência).

Os blocos 8x8x8x8 são separados dentro de um *loop* e para cada um deles é realizada a vetorização da imagem (transformando a imagem 4D em um vetor unidimensional) e a aplicação da GFT (com os autovalores e autovetores calculados fora do *loop*). A partir dessa etapa, os coeficientes do vetor unidimensional estão ordenados da forma desejada.

Para cada bloco, parte dos coeficientes de maior frequência são zerados, fazendo de fato a compressão dos dados, já que a eliminação dos coeficientes reduz a quantidade de informação a ser armazenada ou transmitida de acordo com a porcentagem de coeficientes eliminados.

3.3.2 Decodificação

Os dados de cada bloco $8 \times 8 \times 8 \times 8$, ao chegar ao destino final, passam pelo processo contrário. A transformada de Fourier inversa sobre grafos (IGFT) é aplicada em cada bloco; o vetor unidimensional é convertido de volta para o formato 4D e armazenado em uma matriz de reconstrução da imagem, aguardando o processamento dos demais blocos.

Quando todos os blocos são processados, é possível considerar toda a imagem já processada. A matriz de reconstrução é a imagem comprimida resultante do processo.

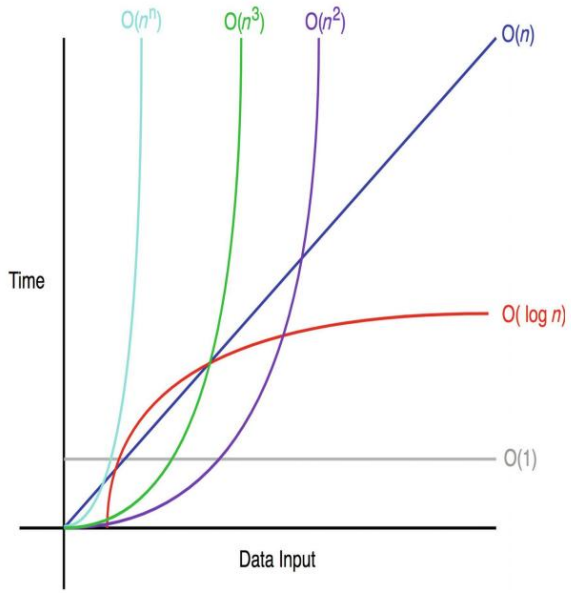
4 ANÁLISE DE COMPLEXIDADE E CUSTO COMPUTACIONAL

A avaliação do desempenho do sistema exige uma análise estruturada de suas etapas. Esta seção introduz a notação *Big-O* e detalha a complexidade algorítmica e o custo de processamento associados ao algoritmo de compressão utilizado.

4.1 ANÁLISE DE COMPLEXIDADE ALGORÍTMICA

Quando se trata de complexidade algorítmica, a notação *Big-O* é uma das principais métricas analisadas. Essa é a métrica fundamental para avaliar a escalabilidade de uma solução. O objetivo dessa análise é observar como a demanda por recursos computacionais cresce em função do volume de dados de entrada (*n).* Em outras palavras, a notação *Big-O* permite delimitar o comportamento do algoritmo no pior cenário possível, identificando o momento em que a implementação deixa de ser viável para aplicações práticas. A Figura 29 mostra algumas notações comuns de *Big-O.* (Sammie, 2019)

Figura 29 - Complexidades comuns de Big-O.



Fonte: Sammie, 2019.

Para definir a complexidade algorítmica do CODEC usado neste trabalho, é preciso definir algumas variáveis para representar os dados no algoritmo e separá-lo em duas fases: pré-processamento e processamento.

Será adotado *N* para definir o número total de pixels da imagem, *K* para a quantidade de pixels em um bloco (para os blocos $8 \times 8 \times 8 \times 8$, K=4096) e *B* para o número de blocos na imagem ($B = N/K$).

No pré-processamento, algumas etapas do algoritmo se destacam. Por exemplo, a definição da topologia é estabelecida a partir de um subalgoritmo que utiliza *loops* para conectar os vértices de um bloco proporcional a K e tem complexidade linear $O(K)$. A maior complexidade nessa etapa é o cálculo da base de Fourier. O algoritmo utiliza a matriz laplaciana de tamanho $K \times K$ e calcula os autovalores e autovetores. A decomposição espectral de uma matriz quadrada possui uma complexidade cúbica em relação ao seu tamanho, ou seja, $O(K^3)$.

Na fase de processamento, se destacam as etapas de vetorização, de custo linear $O(K)$; a aplicação da GFT, em que se multiplica a matriz de autovetores de tamanho $K \times K$ pelo vetor de tamanho K (complexidade $O(K^2)$); e a IGFT, a qual segue a mesma lógica matemática. Nessa fase, predomina o fator $O(K^2)$.

Portanto, é fundamental distinguir as características dessas duas fases. O pré-processamento possui uma taxa de crescimento elevadíssima, definida por $O(K^3)$, mas é executado apenas uma vez e independe do tamanho total da imagem (N). Já a fase de processamento possui uma complexidade menor por bloco ($O(K^2)$), porém possui um número de repetições definido por B (quantidade de blocos). O custo total dessa segunda fase é a multiplicação de B pelo custo do bloco: $O(B \cdot K^2)$, que pode ser reescrito matematicamente como $O(N \cdot K)$. Isso mostra que o tempo de codificação da imagem varia de forma linear em relação a resolução dela, mas pode ser limitado severamente de forma cúbica na primeira fase caso o tamanho do bloco aumente.

4.2 GARGALOS DE PROCESSAMENTO

Tendo em vista como funciona a complexidade algorítmica em cada etapa, é possível prever onde e por quê cada componente do *hardware* será sobrecarregado.

O processador (CPU) é o responsável por realizar as operações matemáticas, armazenando os valores temporariamente na memória RAM. Cada variável instanciada durante a execução do algoritmo ocupa um determinado espaço nessa memória volátil. O trabalho com operações de matrizes densas tende a ser custoso para a memória RAM, principalmente em se tratando de imagens com resoluções altas, como as de *light field*.

Embora a notação *Big-O* seja comumente utilizada para prever o tempo de execução (complexidade de tempo enfrentada pela CPU), ela também dita o crescimento do uso de memória (complexidade de espaço). Em aplicações práticas, as limitações físicas de memória RAM costumam ser a barreira que inviabiliza o algoritmo antes mesmo que o tempo de processamento se torne um problema crônico.

A principal operação limitante para a memória RAM neste CODEC é a etapa em que se calcula a base de Fourier. Para isso, o sistema precisa gerar e manter em memória uma matriz densa de autovetores de dimensão $K \times K$. Utilizando o formato *float64* para maior precisão, o processamento de blocos com dimensão $8 \times 8 \times 8 \times 8$ consome cerca de 134 MB (megabytes) de RAM apenas para esta matriz. Para se ter uma ideia da escalabilidade desse gargalo, se fossem utilizados blocos de dimensão 16x16x16x16, a mesma matriz passaria a exigir aproximadamente 34 GB (gigabytes) de memória, excedendo a capacidade da imensa maioria dos computadores domésticos.

Diante desse cenário, conclui-se que o processador atingirá picos de uso durante os laços de repetição que aplicam a transformada e sua inversa, enquanto a memória RAM sofrerá sua maior carga e alocação estática durante o cálculo da base de Fourier no pré-processamento.

Em termos práticos, a exigência de ordem cúbica $O(K^3)$ para a decomposição espectral é o maior problema do método proposto. Sem o uso de aproximações matemáticas, o cálculo exato da base de Fourier pode inviabilizar a aplicação da GFT em larga escala, impedindo o processamento em tempo real ou a utilização de blocos de maiores dimensões sem o esgotamento dos recursos da máquina. Para contornar essa barreira e tornar esse tipo de compressão comercialmente viável, a literatura recente tem focado no desenvolvimento de algoritmos de aproximação. Estratégias como as transformadas rápidas em grafos (do inglês, *Fast Graph Fourier Transforms* - FGFT) permitem uma aceleração significativa no tempo de execução, tornando a implementação em *hardware* eficiente, com menor consumo de RAM e CPU, sem comprometer severamente a qualidade da reconstrução visual. (Lu; Ortega, 2019)

5 RESULTADOS E DISCUSSÃO

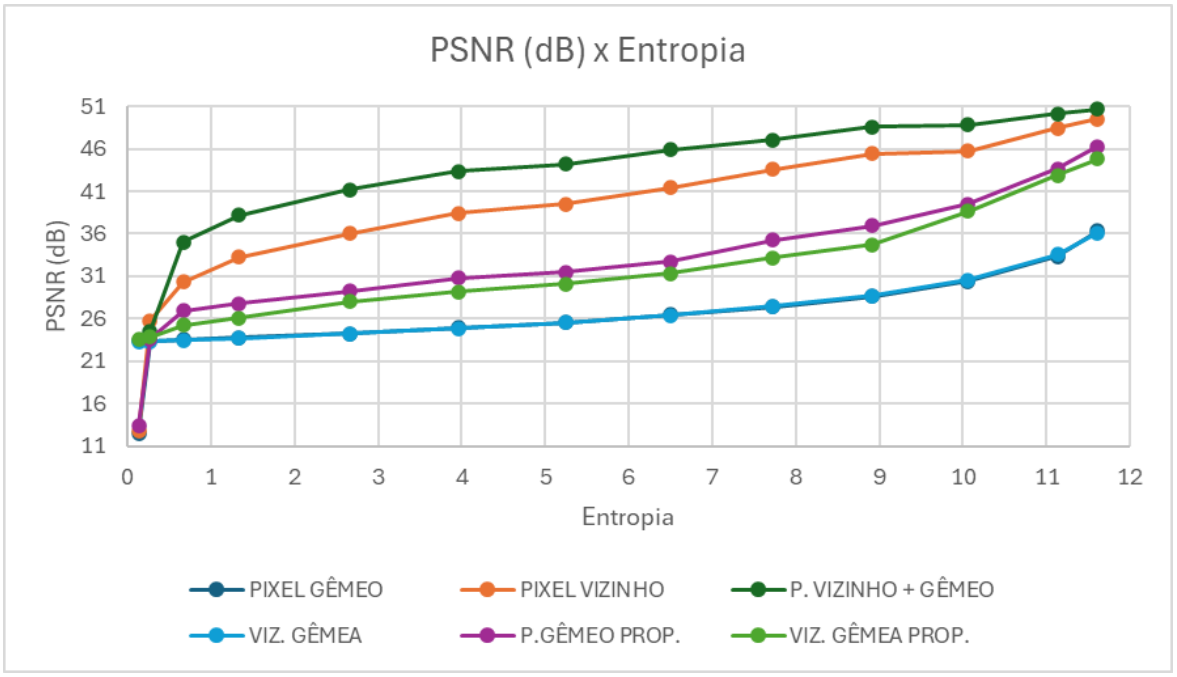
Os testes das topologias propostas foram realizados utilizando as imagens apresentadas nas Figuras 13, 14 e 15. Para cada topologia, executaram-se treze cenários de compressão, variando a porcentagem de coeficientes mantidos: 95%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%, 5%, 2% e 1%. Cabe ressaltar que, quanto maior a porcentagem de retenção, menos coeficientes representativos e, consequentemente, menor é a entropia de Shannon resultante, indicando uma representação com menos dados da imagem reconstruída.

5.1 DESEMPENHO DE COMPRESSÃO

Nesta seção, avalia-se o desempenho de compressão das topologias propostas. Inicialmente, os modelos baseados em grafos são comparados entre si para identificar o mais eficiente. Em seguida, a topologia que apresentar os melhores resultados será comparada diretamente com a 2D-DCT, garantindo que o cenário otimizado da GFT seja confrontado com o padrão consolidado de compressão de imagens.

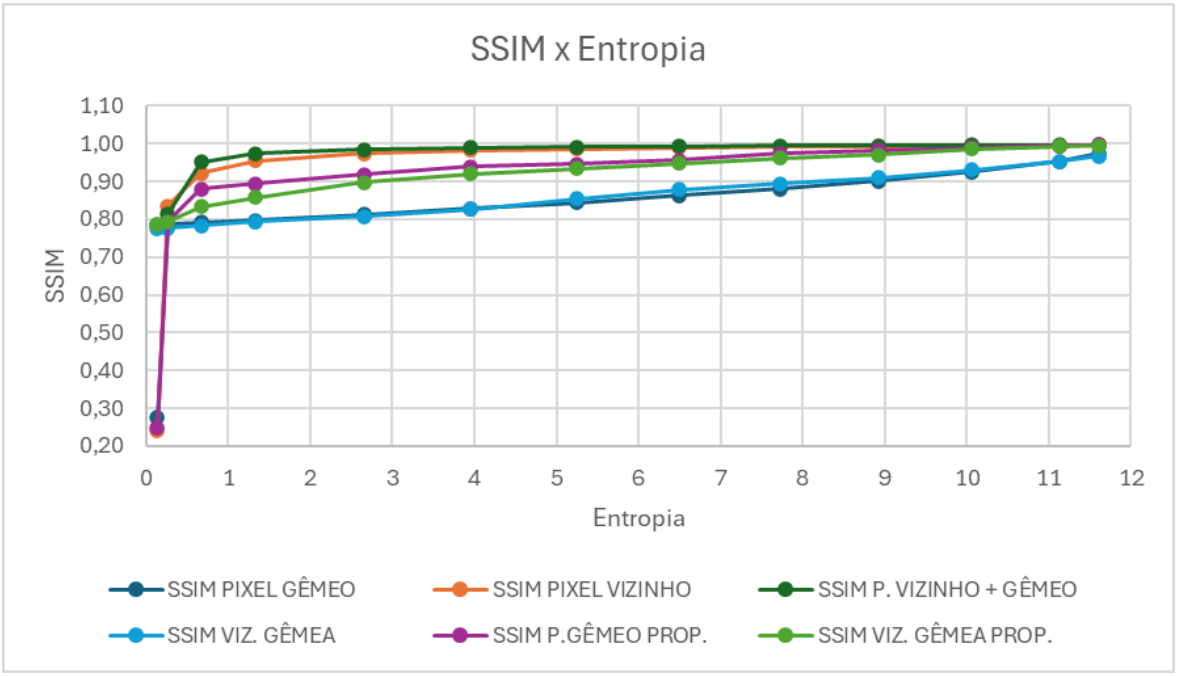
Embora cada imagem tenha sido testada separadamente, os dados do Apêndice A demonstram que os resultados de PSNR e SSIM são bastante próximos para um mesmo nível de entropia. Portanto, para facilitar a análise, optou-se por calcular a média desses valores. A partir desses dados (detalhados nas Tabelas 08 e 9), foram gerados os gráficos das Figuras 30 e 31, que ilustram, respectivamente, o comportamento médio do PSNR e do SSIM em função da entropia.

Figura 30 -Valores médios de PSNR (dB) obtidos pelas topologias propostas para diferentes níveis de entropia.



Fonte: elaborado pelo próprio autor.

Figura 31 - Valores médios de SSIM obtidos pelas topologias propostas para diferentes níveis de entropia.



Fonte: elaborado pelo próprio autor.

A análise dos resultados demonstra uma notável vantagem da topologia Pixel Vizinho + Gêmeo, representada pela curva verde-escura. Esse modelo apresenta um PSNR estável e um SSIM quase constante para entropias maiores que 1. Além disso, por se tratar da combinação de dois grafos, nota-se que sua curva acompanha de perto o comportamento do Pixel Vizinho e

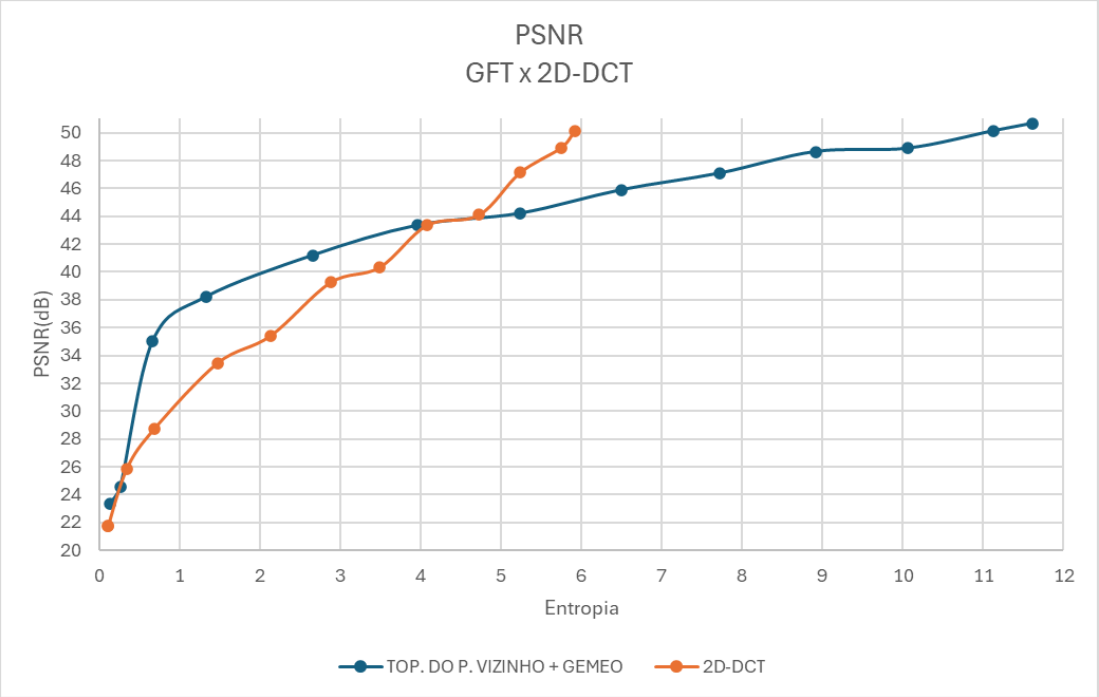
se afasta consideravelmente do Pixel Gêmeo. Isso evidencia que a maior contribuição para o sucesso do modelo híbrido vem das conexões espaciais (Pixel Vizinho).

O desempenho superior da topologia híbrida (Pixel Vizinho + Pixel Gêmeo) em cenários de alta compressão é explicado pelo conceito de suavidade do sinal fundamentado na Seção 2.4. Como as imagens de *light field* possuem dependências tanto no plano espacial quanto no angular, as abordagens isoladas não capturaram toda a similaridade, tratando as transições omitidas como ruído de alta frequência. Ao unificar as conexões de Pixel Vizinho e Pixel Gêmeo, a estrutura topológica proposta alinha-se perfeitamente à natureza quadridimensional dos dados, minimizando a variação do sinal entre os vértices conectados (arestas). Pela definição da forma quadrática da matriz laplaciana ($f^T \boldsymbol{L} f$), essa minimização resulta em um sinal de baixa frequência muito mais concentrado. Sob essa condição de alta suavidade, a GFT alcança um bom nível de esparsidade, concentrando a energia da cena em um número reduzido de coeficientes. Isso justifica o fato de a topologia híbrida operar com menor custo entrópico e preservar adequadamente a integridade estrutural (SSIM) e a relação sinal-ruído (PSNR).

Diante desse cenário, a topologia Pixel Vizinho + Gêmeo demonstra ser a mais eficiente entre os modelos de grafos propostos, sendo, portanto, a escolhida para a comparação final diante da 2D-DCT. As Tabelas 10 e 11, que dão origem às Figuras 32 e 33, apresentam a comparação direta dos resultados entre essa topologia e o padrão de compressão.

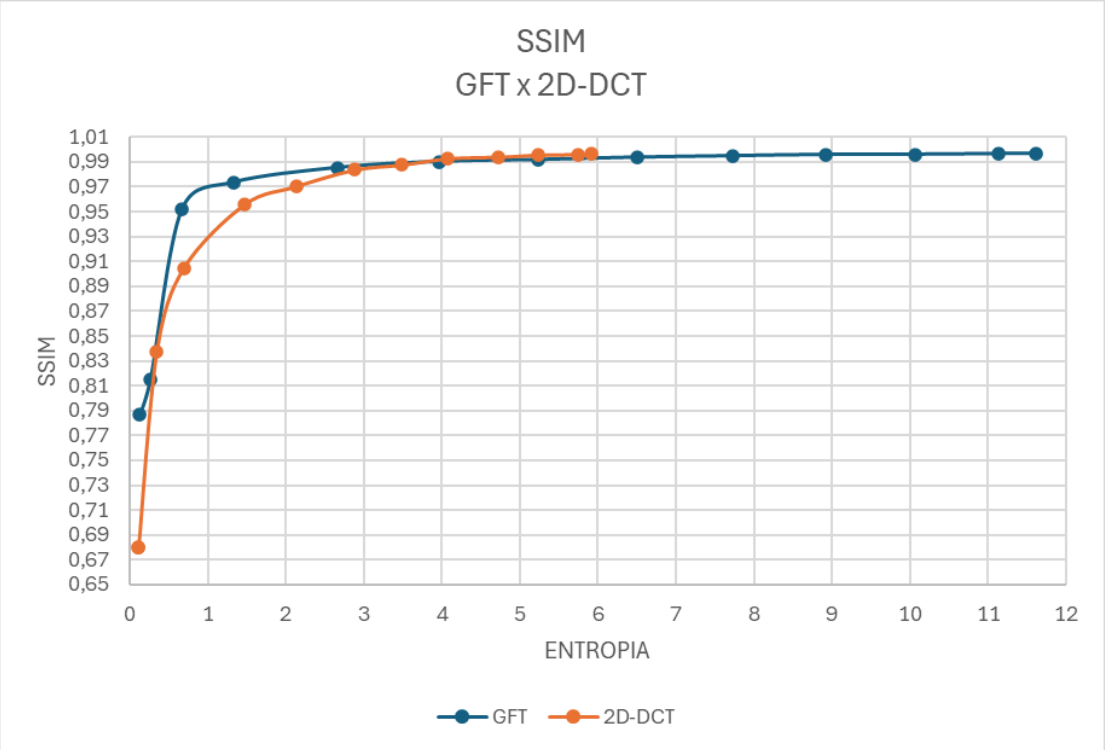
Para esta análise, há um fator crucial a ser considerado: o valor da entropia para uma mesma porcentagem de coeficientes mantidos difere significativamente entre os métodos. Isso ocorre porque a base de processamento de cada transformada possui dimensões distintas: enquanto a GFT opera em blocos de $8 \times 8 \times 8 \times 8$ (processando 4096 coeficientes simultâneos), a 2D-DCT atua em blocos bidimensionais de 8×8 (processando apenas 64 coeficientes). Como consequência, o limite matemático de entropia para a GFT é de 12 bits ($\log_2 4096$), ao passo que para a 2D-DCT é de apenas 6 bits ($\log_2 64$). O resultado disso é evidente nas Figuras 32 e 33, em que a curva correspondente à 2D-DCT inicia seu decaimento em valores próximos a 6 bits e a GFT em valores próximos a 12 bits, que estão próximos seus valores máximos teóricos de entropia.

Figura 32 - Comparativo do PSNR e da entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT.



Fonte: elaborado pelo próprio autor.

Figura 33 - Comparativo do SSIM e da entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT.



Fonte: elaborado pelo próprio autor.

Para a correta interpretação destes resultados, é necessário ter em mente que a entropia atua como a métrica absoluta na compressão de dados. Portanto, a comparação justa entre os métodos exige a análise do PSNR e do SSIM sob uma mesma taxa informacional. A análise da

Figura 32 revela com clareza o ponto de inflexão a favor do modelo proposto: para níveis de entropia inferiores a 4 bits, a GFT entrega valores de PSNR consideravelmente superiores aos da 2D-DCT. Um comportamento análogo é observado na Figura 33, onde a GFT comprova sua resiliência na manutenção da qualidade estrutural (SSIM) em cenários de alta compressão.

Outra forma de avaliar esses dados é estabelecer limites de qualidade visual aceitáveis para o observador humano. Na literatura, considera-se frequentemente um PSNR de 30 dB e um SSIM de 0,95 como patamares de conformidade visual. Ao analisar a Figura 32 com o auxílio da Tabela A. 10, nota-se que a GFT atinge 35,02 dB consumindo apenas 0,668 bits de entropia. Em contrapartida, a 2D-DCT necessita de 2,135 bits para entregar uma qualidade equivalente (35,41 dB). Um cenário análogo ocorre com o SSIM: para alcançar a marca de 0,9517, a GFT utiliza os mesmos 0,668 bits, enquanto a 2D-DCT exige 1,4782 bits para chegar a 0,9559, conforme Figura 33 e Tabela A. 11.

5.2 ANÁLISE DE HARDWARE

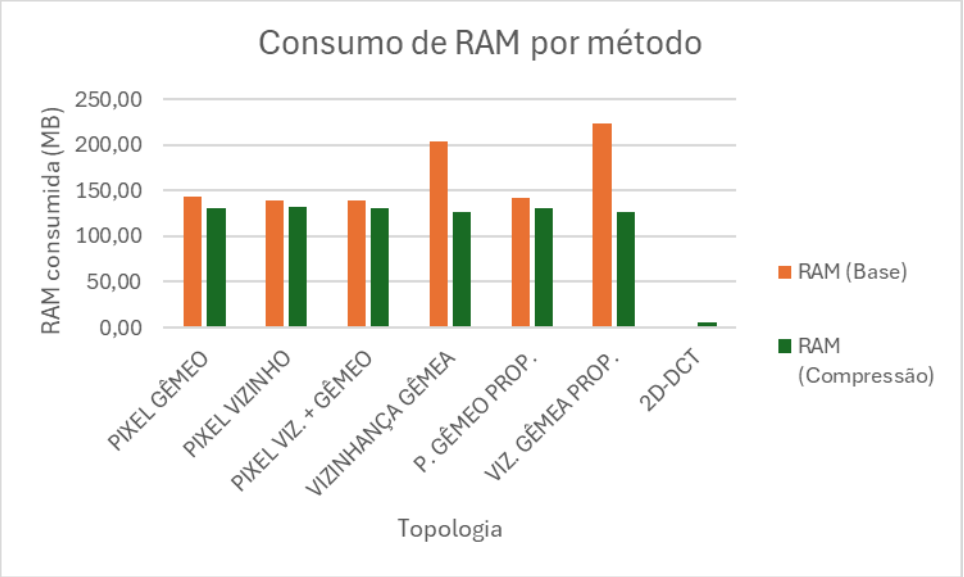
Conforme previamente discutido na Seção 4.2, os principais impactos do uso da GFT decorrem da sua exigência de memória RAM. Esta seção avalia as topologias em termos do seu consumo de memória RAM e uso de CPU no ambiente de testes.

Na seção 4.1, o algoritmo foi analisado em duas etapas distintas: o pré-processamento e a compressão. Nesta seção, de forma similar, para cada topologia, o consumo de RAM e CPU será analisado nas etapas do cálculo da base de Fourier (pré-processamento) e durante as iterações nos blocos $8 \times 8 \times 8 \times 8$ de cálculo da GFT (compressão).

Para garantir a confiabilidade das métricas, o monitoramento foi restrito ao processo do Python (linguagem de programação usada neste trabalho), isolando-o do restante dos processos secundários do sistema operacional. Adicionalmente, implementaram-se rotinas de medição para que a captação dos dados ocorresse estritamente durante os intervalos exatos de execução de cada fase computacional, garantindo a máxima precisão na obtenção dos resultados.

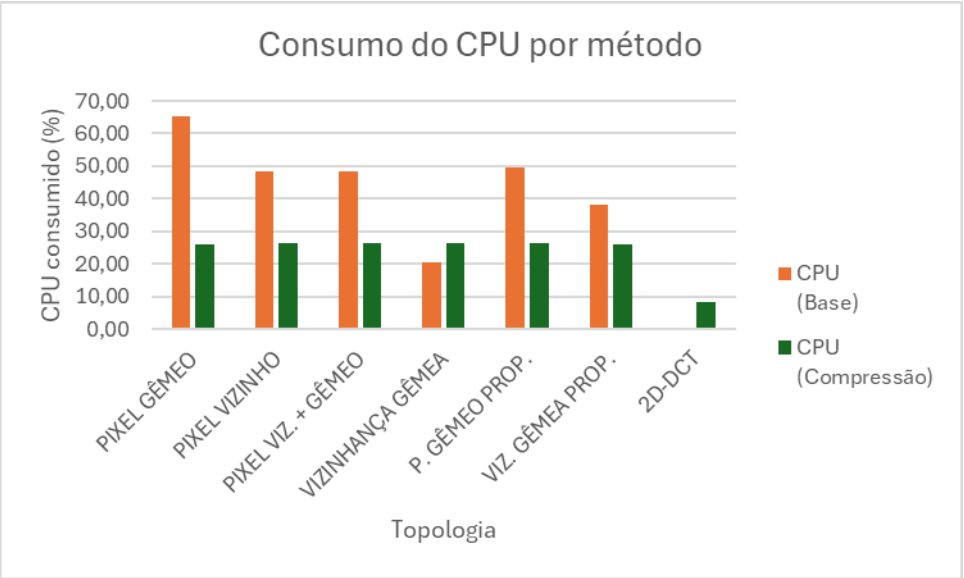
Na seção 5.1, considerou-se a média dos índices de qualidade de imagem para as três imagens diferentes. Durante os testes, foi observada uma constância nos valores de uso de CPU e memória RAM independentemente das imagens e das diferentes taxas de compressão. A consolidação desses dados encontra-se detalhada nas Tabelas 12 a 18, que dão origem à Tabela A. 19, atuando como base estrutural para os gráficos projetados nas Figuras 34 e 35.

Figura 34 - Consumo de RAM por método utilizado.



Fonte: elaborado pelo próprio autor.

Figura 35 - Uso de CPU por método utilizado.



Fonte: elaborado pelo próprio autor.

A Figura 34 mostra um comportamento similar para a maioria das topologias da GFT; no entanto, destacam-se picos nas topologias que incorporam o conceito de Vizinhança Gêmea. Esse aumento de demanda pode ser explicado devido ao alto número de arestas presentes no grafo, já que essas topologias utilizam mais ligações de vértices comparadas às demais.

Além disso, é possível notar que o uso da 2D-DCT aparece apenas no momento da compressão, uma vez que não existem etapas de pré-processamento no cálculo da 2D-DCT. O baixo consumo de memória no padrão JPEG justifica-se principalmente devido ao tamanho dos

blocos. Como necessita de blocos de apenas 8x8, o consumo é baixíssimo. Isso seria diferente se o padrão não utilizasse o conceito de divisão de blocos similar ao detalhado na Seção 3.1.

A Figura 35 ilustra o consumo da CPU para os diferentes métodos durante as duas etapas do algoritmo. É possível observar que a demanda computacional ocorre de forma mais acentuada durante o cálculo da base de Fourier. A 2D-DCT, conforme fundamentado anteriormente, não requer uma etapa de pré-processamento, restringindo o uso da CPU exclusivamente à fase de compressão.

De forma contraintuitiva, as topologias com maior número de arestas são as que menos exigem do processador. Por exemplo, o modelo com menos ligações (Pixel Gêmeo) é o que mais consome CPU na geração da base, enquanto a topologia com mais arestas (Vizinhança Gêmea) mostra-se consideravelmente mais eficiente. Esse fenômeno é explicado pela autodecomposição da matriz laplaciana: ao conectar pixels sem adjacência espacial, o grafo fragmenta-se em múltiplos componentes desconexos. A introdução das conexões espaciais unifica a estrutura em um único componente fortemente conectado, o que acelera a convergência do solucionador matemático e reduz o consumo da CPU.

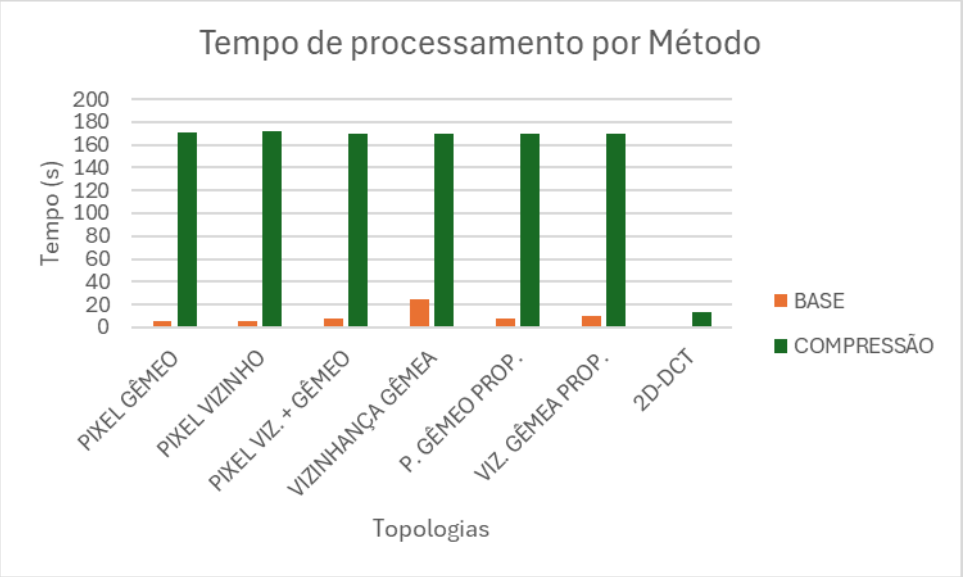
5.3 ANÁLISE DE TEMPO

A análise de processamento temporal conduzida nesta seção segue uma metodologia análoga à da Seção 5.2, considerando as duas etapas do algoritmo e as médias de tempo para as três imagens em todas as taxas de compressão. Os dados brutos consolidados encontram-se nas Tabelas 20 a 26, culminando nos resultados da Tabela A. 27.

A Figura 36 apresenta o gráfico originado desses dados. Na etapa de pré-processamento, todas as topologias demandam de 5 a 25 segundos para calcular os autovalores e autovetores da matriz laplaciana. O grafo da Vizinhança Gêmea exige o maior tempo de execução, apesar de a Seção 5.2 ter demonstrado seu baixo impacto na CPU. Esse fenômeno evidencia um gargalo de memória: devido à alta densidade de arestas da matriz, a CPU esgota rapidamente seu espaço em memória *cache* e entra em tempo de espera para a transferência de dados da RAM.

A maior parcela do tempo total, contudo, reside na etapa de compressão e descompressão. O algoritmo processa individualmente cada bloco $8 \times 8 \times 8 \times 8$, aplicando a transformada (GFT), eliminando componentes de alta frequência e executando a transformada inversa (IGFT). Ao todo, esse ciclo matricial é repetido 4212 vezes para cada imagem.

Figura 36 - Média do tempo de processamento por método utilizado.



Fonte: elaborado pelo próprio autor.

A superioridade computacional da 2D-DCT em termos de tempo de execução e consumo de memória justifica-se por sua natureza matemática analítica e separável. Como a base de cossenos da DCT é universal e estática, ela dispensa qualquer etapa de processamento prévio, permitindo ainda que a transformação de blocos bidimensionais seja dividida em operações unidimensionais sucessivas de baixo custo. Em contrapartida, a aplicação da GFT sobre partições quadridimensionais de $8 \times 8 \times 8 \times 8$ impõe um custo de pré-processamento substancial. Embora esse cálculo da base espectral ocorra somente uma vez na etapa de inicialização da topologia, a necessidade de diagonalização de uma matriz laplaciana densa de dimensões 4096×4096 representa um esforço computacional significativamente superior ao padrão de mercado.

6 CONCLUSÃO

Nesta seção, são consolidadas as descobertas finais da pesquisa. Apresentam-se as considerações sobre o desempenho das topologias desenvolvidas frente ao padrão tradicional, bem como as limitações da abordagem e as sugestões para o direcionamento de trabalhos futuros.

6.1 CONSIDERAÇÕES GERAIS

As imagens de *light field* possuem um vasto potencial para se tornarem uma tecnologia de adoção em massa e uso prático. No entanto, o elevado volume de dados gerado por essas capturas configura-se como um gargalo, impedindo que a tecnologia alcance seu pleno protagonismo.

Para mitigar esse problema, as topologias desenvolvidas neste trabalho para a Transformada de Fourier em Grafos (GFT) foram avaliadas sob a ótica de métricas de qualidade visual, entropia de Shannon, consumo de memória RAM, uso de CPU e tempo de processamento. Após uma análise minuciosa dos dados obtidos na Seção 0, constata-se que a topologia que combina os conceitos de Pixel Vizinho e Pixel Gêmeo obteve o maior destaque.

O grande mérito desta pesquisa reside na comprovação de que, em cenários de compressão severa, essa estrutura híbrida superou a 2D-DCT (padrão consolidado no formato JPEG). Para ilustrar o patamar de desempenho alcançado, em um cenário de corte retendo apenas 5% dos coeficientes (que equivale a um limite teórico de entropia próximo a 0,6 bit/símbolo), a topologia proposta garantiu a reconstrução da imagem com excelentes 35,02 dB de PSNR e 0,95 de SSIM. Sob a mesma equivalência de entropia, o método da 2D-DCT garante apenas 28,75 dB de PSNR e 0,90 de SSIM. Esse ganho de mais de 6 dB evidencia que a GFT, ao preservar a correspondência angular da matriz 4D, reduz drasticamente as perdas de compressão.

Evidencia-se, porém, que essa superioridade na preservação visual traz consigo um custo associado ao consumo de *hardware* e ao tempo de processamento. A complexidade cúbica atrelada ao cálculo da base Laplaciana na GFT exige um esforço computacional expressivamente mais elevado em comparação à 2D-DCT. Dessa forma, conclui-se que, dependendo dos requisitos do sistema, a aplicação da GFT utilizando as topologias espaciais e angulares propostas apresenta-se como uma alternativa substancialmente mais eficaz para a compressão de imagens de *light field* do que a adoção do padrão JPEG.

6.2 LIMITAÇÕES E TRABALHOS FUTUROS

Apesar dos resultados promissores alcançados na compressão das imagens *light field*, o presente estudo possui limitações inerentes ao escopo e à infraestrutura de testes, o que abre um vasto leque para investigações futuras. Dentre as principais propostas de continuidade para o aprimoramento da tecnologia, destacam-se:

- Expansão para múltiplos canais de cores: O algoritmo atual foi estruturado e validado para atuar sobre a luminância ou em canais únicos, simplificando a modelagem dos grafos. Como trabalho futuro, é fundamental adaptar as topologias propostas para processar imagens coloridas completas, explorando as redundâncias cromáticas em espaços de cor como RGB ou YCbCr, avaliando o impacto dessas novas dimensões na entropia final.
- Implementação da Transformada Rápida em Grafos (FGFT): Como o cálculo da GFT tradicional exige a autodecomposição completa da Matriz Laplaciana, o custo computacional impõe uma barreira de tempo significativa. Uma solução direta para essa limitação é o estudo e a aplicação de algoritmos de aproximação, como a *Fast Graph Fourier Transform* (FGFT) proposta por Keng Shih Lu e Antonio Ortega (2019). A adoção dessa transformada rápida tem o potencial de reduzir drasticamente a complexidade matemática, aproximando a eficiência da GFT à dos algoritmos rápidos da DCT.
- Aceleração via processamento paralelo em GPU: Todo o fluxo de dados deste trabalho foi executado sob a arquitetura de processamento central (CPU). Uma vez que as transformadas em grafos baseiam-se em multiplicações e fatorações de matrizes gigantescas, a migração do código para ambientes de processamento paralelo em placas de vídeo (GPU) — utilizando arquiteturas como CUDA — permitiria a execução simultânea de milhares de blocos, mitigando o elevado tempo de compressão documentado na Seção 0.
- Redução da dimensionalidade dos blocos: Com o intuito de solucionar o gargalo de memória identificado durante a geração da base de Fourier, propõe-se a otimização do particionamento do *light field*. A atual abordagem em blocos de $8 \times 8 \times 8 \times 8$ gera matrizes que excedem a capacidade da memória *cache* dos processadores comerciais. Uma via de investigação promissora consiste em reduzir os recortes espaciais para arquiteturas como $8 \times 8 \times 4 \times 4$. Essa reestruturação reduziria o tamanho dos dados

para limites compatíveis com a memória de altíssima velocidade da CPU, acelerando o pré-processamento matricial sem comprometer a integridade topológica do modelo.

- Estudo de sensibilidade de θ: Neste trabalho, o parâmetro θ, responsável por controlar o decaimento dos pesos nas topologias proporcionais, foi fixado empiricamente em 5 para viabilizar a análise comparativa entre as estruturas espaciais, não havendo tempo hábil para uma otimização. Propõe-se, para investigações futuras, a condução de um estudo de sensibilidade para determinar o valor ótimo de θ. Avaliar o impacto da variação deste parâmetro sobre a esparsidade do grafo e, consequentemente, nas métricas de distorção (PSNR e SSIM), possui o potencial de maximizar a eficiência da topologia.

REFERÊNCIAS

ELL, T. A.; LE BIHAN, N.; SANGWINE, S. J. **Quaternion Fourier Transforms for Signal and Image Processing**. 1. ed. Londres: ISTE Ltd, 2014. v. 1.

FANG, L. **Plenoptic Imaging and Processing**. Beijing: Springer, 2025.

FERREIRA, F. A. B. S. **Técnicas de marca d'água no domínio da transformada para imagens plenópticas**. 2020. Tese (Doutorado em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, 2020.

GONZALEZ, R. C.; WOODS, R. C. **Processamento Digital de Imagens**. 3. ed. São Paulo: Pearson Prentice Hall, 2010.

HORÉ, A.; ZIOU, D. Image Quality Metrics: PSNR vs. SSIM. *In*: INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION, 20., 2010, Istambul. **Anais** [...]. Istambul: IEEE, 2010. p. 2366-2369. Disponível em: https://doi.org/10.1109/ICPR.2010.579. Acesso em: 15 abr. 2026.

KHAYAM, S. A. The Discrete Cosine Transform (DCT): Theory and Application. **Michigan State University Technical Report**, East Lansing, v. 1, n. 1, p. 1-32, mar. 2003. Disponível em: https://www.cse.iitd.ac.in/~pkalra/col783-2017/DCT-TR802.pdf. Acesso em: 18 abr. 2026.

LATHI, B. P. **Sinais e sistemas lineares**. 2. ed. Porto Alegre: Bookman, 2014.

LU, K.-S.; ORTEGA, A. Fast Graph Fourier Transforms Based on Graph Symmetry and Bipartition. **IEEE Transactions on Signal Processing**, [S. l.], v. 67, n. 18, p. 4855-4869, set. 2019. Disponível em: https://arxiv.org/pdf/1907.07875. Acesso em: 20 abr. 2026.

NEELAMANI, R. *et al*. JPEG compression history estimation for color images. **IEEE Transactions on Image Processing**, [S. l.], v. 15, n. 6, p. 1365-1378, jun. 2006. Disponível em: https://doi.org/10.1109/ICIP.2003.1247227. Acesso em: 22 abr. 2026.

PEREIRA, F.; SILVA, E. A. B. da; LAFRUIT, G. Light field imaging and processing. **Academic Press Library in Signal Processing**, [S. l.], v. 6, p. 75-111, 2018. Disponível em: https://www.sciencedirect.com/science/chapter/edited-volume/abs/pii/B9780128118894000026. Acesso em: 23 abr. 2026.

SAMMIE, B. Big-O Notation. *In*: SAMMIE, B. **JavaScript Data Structures and Algorithm**. 1. ed. Hamilton: Apress, 2019. cap. 1, p. 1-15. Disponível em: https://doi.org/10.1007/978-1-4842-3988-9. Acesso em: 25 abr. 2026.

SHANNON, C. E. A Mathematical Theory of Communication. **The Bell System Technical Journal**, [S. l.], v. 27, n. 3, p. 379-423, jul. 1948. Disponível em: https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf. Acesso em: 26 abr. 2026.

SHUMAN, D. I. *et al*. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. **IEEE Signal Processing**

Magazine, [S. l.], v. 30, n. 3, p. 83-98, maio 2013. Disponível em: https://doi.org/10.48550/arXiv.1211.0053. Acesso em: 28 abr. 2026.

SINGH, K.; RANADE, S. K.; SINGH, C. Comparative performance analysis of various wavelet and nonlocal means based approaches for image denoising. **Optik - International Journal for Light and Electron Optics**, [S. l.], v. 127, n. 24, p. 1-15, nov. 2016. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0030402616313729. Acesso em: 29 abr. 2026.

ZHANG, C.; CHEN, T. A self-reconfigurable camera array. *In*: EUROGRAPHICS SYMPOSIUM ON RENDERING, 2004, Norrköping. **Proceedings** [...]. Norrköping: Eurographics Association, 2004. p. 243-254. Disponível em: http://chenlab.ece.cornell.edu/Publication/Cha/eurosym04_Cha.pdf. Acesso em: 30 abr. 2026.

APÊNDICE A – Tabelas

Tabela A. 1 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Gêmeo.

TOPOLOGIA DO PIXEL GÊMEO							
		FIGURA 14		FIGURA 15		FIGURA 13	
COEFS. MANTIDOS	ENTROPIA	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
95%	11,616	34,5015	0,9639	36,8831	0,9771	37,7526	0,9789
90%	11,131	31,3894	0,9416	33,7977	0,9597	34,7643	0,9612
80%	10,062	28,4241	0,9129	30,9927	0,9318	31,8114	0,9309
70%	8,921	26,6197	0,8927	29,2448	0,9075	30,0438	0,9041
60%	7,727	25,3834	0,8767	28,0222	0,8857	28,8077	0,8799
50%	6,500	24,4317	0,8642	27,0581	0,8650	27,8653	0,8579
40%	5,241	23,5524	0,8522	26,1507	0,8445	27,0228	0,8355
30%	3,958	22,8752	0,8424	25,4787	0,8267	26,3551	0,8151
20%	2,657	22,2379	0,8334	24,8089	0,8083	25,7280	0,7944
10%	1,335	21,7469	0,8262	24,2992	0,7921	25,2301	0,7751
5%	0,668	21,5144	0,8239	24,0344	0,7855	25,0014	0,7662
2%	0,266	21,3944	0,8234	23,9056	0,7826	24,8807	0,7612
1%	0,131	13,2104	0,2936	10,8405	0,2967	13,4746	0,2352

Fonte: elaborado pelo próprio autor.

Tabela A. 2 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Vizinho.

TOPOLOGIA DO PIXEL VIZINHO							
		FIGURA 14		FIGURA 15		FIGURA 13	
COEFS. MANTIDOS	ENTROPIA	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
95%	11,616	49,7498	0,9945	49,3936	0,9966	49,5035	0,9971
90%	11,131	48,7198	0,9939	48,2861	0,9962	48,3787	0,9968
80%	10,062	46,1756	0,9920	45,5413	0,9947	45,4337	0,9954
70%	8,921	45,9627	0,9917	45,2985	0,9945	45,1819	0,9951
60%	7,727	44,3638	0,9901	43,3718	0,9930	43,0152	0,9933
50%	6,500	42,2618	0,9872	41,1882	0,9905	40,8945	0,9908
40%	5,241	40,3604	0,9841	39,2012	0,9876	38,9738	0,9874
30%	3,958	39,3590	0,9811	38,0856	0,9845	37,8325	0,9839
20%	2,657	37,1531	0,9736	35,6825	0,9754	35,2484	0,9735
10%	1,335	34,3362	0,9571	33,1042	0,9572	32,3825	0,9521
5%	0,668	31,5682	0,9323	30,0230	0,9199	29,4035	0,9129
2%	0,266	26,6135	0,8680	25,0381	0,8046	25,7363	0,8321
1%	0,131	13,6335	0,2740	11,0480	0,2574	13,6028	0,1919

Fonte: elaborado pelo próprio autor.

Tabela A. 3 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Vizinho + Gêmeo.

TOPOLOGIA DO PIXEL VIZINHO + GÊMEO							
		FIGURA 14		FIGURA 15		FIGURA 13	
COEFS. MANTIDOS	ENTROPIA	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
95%	11,616	50,9030	0,9955	50,4428	0,9972	50,6413	0,9975
90%	11,131	50,5520	0,9953	49,7694	0,9969	50,0448	0,9973
80%	10,062	49,8744	0,9950	48,1766	0,9963	48,5687	0,9967
70%	8,921	49,6994	0,9948	47,9304	0,9961	48,2733	0,9965
60%	7,727	48,7946	0,9940	46,0468	0,9949	46,4164	0,9953
50%	6,500	47,9740	0,9933	44,6890	0,9936	45,0013	0,9939
40%	5,241	46,7023	0,9921	42,6411	0,9913	43,2550	0,9918
30%	3,958	46,0108	0,9912	41,7317	0,9896	42,2767	0,9899
20%	2,657	44,6391	0,9889	39,2821	0,9839	39,6261	0,9831
10%	1,335	42,3138	0,9838	35,9843	0,9698	36,3427	0,9673
5%	0,668	39,5634	0,9780	32,4148	0,9394	33,1039	0,9378
2%	0,266	22,7499	0,8468	25,0393	0,8070	25,9131	0,7919
1%	0,131	21,3737	0,8199	23,8725	0,7805	24,8434	0,7591

Fonte: elaborado pelo próprio autor.

Tabela A. 4 - Valores de PSNR e SSIM das três imagens de teste com a topologia Vizinhança Gêmea.

TOPOLOGIA DA VIZINHANÇA GÊMEA							
		FIGURA 14		FIGURA 15		FIGURA 13	
COEFS. MANTIDOS	ENTROPIA	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
95%	11,616	34,2940	0,9553	36,8849	0,9736	37,2125	0,9743
90%	11,131	31,7283	0,9413	34,1576	0,9594	34,7600	0,9595
80%	10,062	28,6379	0,9217	31,1496	0,9358	31,8641	0,9331
70%	8,921	26,7721	0,9072	29,3355	0,9150	30,1183	0,9101
60%	7,727	25,4486	0,8963	28,0219	0,8967	28,8799	0,8897
50%	6,500	24,4285	0,8873	26,9457	0,8785	27,8585	0,8703
40%	5,241	23,6336	0,8634	26,0744	0,8551	26,9383	0,8440
30%	3,958	22,9415	0,8395	25,4116	0,8276	26,2667	0,8142
20%	2,657	22,3053	0,8254	24,7986	0,8066	25,6796	0,7909
10%	1,335	21,7706	0,8187	24,2436	0,7904	25,1674	0,7725
5%	0,668	21,5133	0,8105	23,9418	0,7793	24,8237	0,7577
2%	0,266	21,3666	0,8072	23,8021	0,7729	24,6879	0,7510
1%	0,131	21,3209	0,8072	23,7574	0,7716	24,6461	0,7494

Fonte: elaborado pelo próprio autor.

Tabela A. 5 - Valores de PSNR e SSIM das três imagens de teste com a topologia Pixel Gêmeo Proporcional.

TOPOLOGIA DO PIXEL GÊMEO PROPORCIONAL							
		FIGURA 14		FIGURA 15		FIGURA 13	
COEFS. MANTIDOS	ENTROPIA	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
95%	11,616	45,1241	0,9946	47,0229	0,9967	46,6993	0,9963
90%	11,131	42,1475	0,9929	44,6892	0,9954	44,1254	0,9944
80%	10,062	37,2192	0,9878	41,3047	0,9915	40,0678	0,9877
70%	8,921	34,5352	0,9810	38,7246	0,9863	37,6134	0,9797
60%	7,727	32,7505	0,9729	37,1174	0,9796	35,8483	0,9696
50%	6,500	30,1458	0,9548	34,3974	0,9659	33,5867	0,9510
40%	5,241	28,9834	0,9438	32,8097	0,9572	32,6676	0,9399
30%	3,958	28,3299	0,9376	32,0161	0,9507	31,9809	0,9299
20%	2,657	26,8733	0,9195	30,2978	0,9314	30,5671	0,9045
10%	1,335	25,4835	0,8991	28,5499	0,9094	29,3129	0,8765
5%	0,668	24,6577	0,8883	27,6795	0,8945	28,4718	0,8577
2%	0,266	21,6296	0,8250	24,1540	0,7890	25,1315	0,7675
1%	0,131	14,0533	0,2618	11,9318	0,2785	14,4227	0,2023

Fonte: elaborado pelo próprio autor.

Tabela A. 6 - Valores de PSNR e SSIM das três imagens de teste com a topologia Vizinhança Gêmea Proporcional.

TOPOLOGIA DA VIZINHANÇA GÊMEA PROPORCIONAL							
		FIGURA 14		FIGURA 15		FIGURA 13	
COEFS. MANTIDOS	ENTROPIA	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
95%	11,616	43,5139	0,9927	45,8100	0,9954	45,0604	0,9952
90%	11,131	41,4404	0,9912	44,0943	0,9940	43,1448	0,9930
80%	10,062	36,5085	0,9839	40,2594	0,9885	39,1335	0,9850
70%	8,921	32,4100	0,9675	36,2907	0,9761	35,4849	0,9686
60%	7,727	30,9562	0,9579	34,3720	0,9680	34,2120	0,9583
50%	6,500	29,2342	0,9453	32,3051	0,9549	32,5507	0,9416
40%	5,241	28,0094	0,9324	30,8473	0,9429	31,3756	0,9248
30%	3,958	27,1233	0,9216	29,9957	0,9295	30,4204	0,9089
20%	2,657	26,0781	0,9022	28,7505	0,9091	29,2766	0,8830
10%	1,335	24,1713	0,8682	26,6668	0,8662	27,4179	0,8374
5%	0,668	23,2769	0,8498	25,7816	0,8402	26,5417	0,8104
2%	0,266	21,9382	0,8195	24,3472	0,7940	25,1909	0,7678
1%	0,131	21,6997	0,8144	24,1210	0,7857	24,9807	0,7600

Fonte: elaborado pelo próprio autor.

Tabela A. 7 - Valores de PSNR e SSIM das três imagens de teste com aplicação da 2D-DCT.

TRANSFORMADA DISCRETA DO COSSENO BIDIMENSIONAL							
		FIGURA 14		FIGURA 15		FIGURA 13	
COEFS. MANTIDOS	ENTROPIA	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
95%	5,907	50,3560	0,9951	50,1005	0,9970	49,8878	0,9973
90%	5,737	49,2778	0,9948	48,7511	0,9966	48,7800	0,9971
80%	5,227	47,8201	0,9943	46,5249	0,9959	47,0772	0,9965
70%	4,718	45,2195	0,9933	42,8765	0,9941	44,2314	0,9947
60%	4,076	44,9654	0,9928	41,9084	0,9924	43,2535	0,9931
50%	3,486	41,5599	0,9881	39,0781	0,9874	40,3993	0,9877
40%	2,871	40,5332	0,9837	38,0737	0,9831	39,1178	0,9840
30%	2,129	35,3949	0,9713	34,8441	0,9715	36,0142	0,9683
20%	1,474	34,1274	0,9612	32,8279	0,9555	33,4459	0,9512
10%	0,690	28,2580	0,9208	28,7082	0,9063	29,2912	0,8860
5%	0,347	24,9390	0,8656	26,1148	0,8394	26,6006	0,8063
2%	0,116	20,5454	0,7193	21,9360	0,6577	22,8672	0,6640
1%	0,116	20,5454	0,7193	21,9360	0,6577	22,8672	0,6640

Fonte: elaborado pelo próprio autor.

Tabela A. 8 - Valores médios de PSNR (dB) obtidos pelas topologias propostas para diferentes níveis de entropia.

	PSNR (dB)					
ENTROPIA	PIXEL GÊMEO	PIXEL VIZINHO	P. VIZINHO + GÊMEO	VIZ. GÊMEA	P. GÊMEO PROP.	VIZ. GÊMEA PROP.
11,616	36,3791	49,5490	**50,6624**	36,1305	46,2821	44,7948
11,131	33,3171	48,4615	**50,1221**	33,5486	43,6540	42,8932
10,062	30,4094	45,7169	**48,8732**	30,5505	39,5306	38,6338
8,921	28,6361	45,4810	**48,6344**	28,7420	36,9577	34,7285
7,727	27,4044	43,5836	**47,0859**	27,4501	35,2387	33,1801
6,500	26,4517	41,4482	**45,8881**	26,4109	32,7100	31,3633
5,241	25,5753	39,5118	**44,1995**	25,5488	31,4869	30,0774
3,958	24,9030	38,4257	**43,3397**	24,8733	30,7756	29,1798
2,657	24,2583	36,0280	**41,1824**	24,2612	29,2461	28,0351
1,335	23,7587	33,2743	**38,2136**	23,7272	27,7821	26,0853
0,668	23,5167	30,3316	**35,0274**	23,4263	26,9363	25,2001
0,266	23,3936	**25,7960**	24,5674	23,2855	23,6384	23,8254
0,131	12,5085	12,7614	23,3632	23,2415	13,4693	**23,6005**

Fonte: elaborado pelo próprio autor.

Tabela A. 9 - Valores médios de SSIM obtidos pelas topologias propostas para diferentes níveis de entropia.

ENTROPIA	SSIM					
	PIXEL GÊMEO	PIXEL VIZINHO	P. VIZINHO + GÊMEO	VIZ. GÊMEA	P.GÊMEO PROP.	VIZ. GÊMEA PROP.
11,616	0,9733	0,9961	**0,9967**	0,9677	0,9959	0,9944
11,131	0,9542	0,9956	**0,9965**	0,9534	0,9942	0,9927
10,062	0,9252	0,9940	**0,9960**	0,9302	0,9890	0,9858
8,921	0,9014	0,9938	**0,9958**	0,9108	0,9823	0,9707
7,727	0,8808	0,9921	**0,9947**	0,8942	0,9740	0,9614
6,500	0,8624	0,9895	**0,9936**	0,8787	0,9572	0,9473
5,241	0,8441	0,9864	**0,9917**	0,8542	0,9470	0,9334
3,958	0,8281	0,9832	**0,9902**	0,8271	0,9394	0,9200
2,657	0,8120	0,9742	**0,9853**	0,8076	0,9185	0,8981
1,335	0,7978	0,9555	**0,9736**	0,7939	0,8950	0,8573
0,668	0,7919	0,9217	**0,9517**	0,7825	0,8802	0,8335
0,266	0,7891	**0,8349**	0,8152	0,7770	0,7938	0,7938
0,131	0,2752	0,2411	0,7865	0,7761	0,2475	**0,7867**

Fonte: elaborado pelo próprio autor.

Tabela A. 10 - Comparativo do PSNR e entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT.

COEFS. MANTIDOS	TOP. DO P. VIZINHO + GEMEO		2D-DCT	
	ENTROPIA	PSNR (dB)	ENTROPIA	PSNR (dB)
95%	11,616	**50,6624**	5,920	50,115
90%	11,131	**50,1221**	5,751	48,936
80%	10,062	**48,8732**	5,241	47,141
70%	8,921	**48,6344**	4,732	44,109
60%	7,727	**47,0859**	4,086	43,376
50%	6,500	**45,8881**	3,496	40,346
40%	5,241	**44,1995**	2,880	39,242
30%	3,958	**43,3397**	2,135	35,418
20%	2,657	**41,1824**	1,478	33,467
10%	1,335	**38,2136**	0,691	28,752
5%	0,668	**35,0274**	0,347	25,885
2%	0,266	**24,5674**	0,116	21,783
1%	0,131	**23,3632**	0,116	21,783

Fonte: elaborado pelo próprio autor.

Tabela A. 11 - Comparativo do SSIM e entropia da Topologia do Pixel Vizinho + Gêmeo com a 2D-DCT.

COEFS. MANTIDOS	TOP. DO P. VIZINHO + GEMEO		2D-DCT	
	ENTROPIA	SSIM	ENTROPIA	SSIM
95%	11,616	**0,9967**	5,920	0,9965
90%	11,131	**0,9965**	5,751	0,9962
80%	10,062	**0,9960**	5,241	0,9956
70%	8,921	**0,9958**	4,732	0,9940
60%	7,727	**0,9947**	4,086	0,9928
50%	6,500	**0,9936**	3,496	0,9877
40%	5,241	**0,9917**	2,880	0,9836
30%	3,958	**0,9902**	2,135	0,9704
20%	2,657	**0,9853**	1,478	0,9560
10%	1,335	**0,9736**	0,691	0,9044
5%	0,668	**0,9517**	0,347	0,8371
2%	0,266	**0,8152**	0,116	0,6803
1%	0,131	**0,7865**	0,116	0,6803

Fonte: elaborado pelo próprio autor.

Tabela A. 12 - Medições de consumo de RAM e CPU na topologia do Pixel Gêmeo.

PIXEL GÊMEO												
COEFS. MANTIDOS	FIGURA 13				FIGURA 14				FIGURA 15			
	BASE		COMPRESSÃO		BASE		COMPRESSÃO		BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
95%	143,64	63,59	131,63	26,11	144,66	65,56	131,77	25,97	142,17	66,89	130,86	25,93
90%	143,64	63,59	131,84	26,18	144,66	65,56	131,88	25,89	142,17	66,89	131,79	26,22
80%	143,64	63,59	131,74	26,11	144,66	65,56	131,85	24,22	142,17	66,89	130,77	25,93
70%	143,64	63,59	131,81	25,93	144,66	65,56	130,84	25,30	142,17	66,89	131,77	26,12
60%	143,64	63,59	131,77	26,03	144,66	65,56	131,77	26,16	142,17	66,89	130,81	25,81
50%	143,64	63,59	131,75	26,43	144,66	65,56	131,76	25,82	142,17	66,89	130,81	26,33
40%	143,64	63,59	131,74	26,28	144,66	65,56	131,77	26,34	142,17	66,89	131,70	26,12
30%	143,64	63,59	131,83	26,18	144,66	65,56	131,90	25,86	142,17	66,89	131,85	26,17
20%	143,64	63,59	131,82	26,34	144,66	65,56	129,59	26,21	142,17	66,89	131,78	26,07
10%	143,64	63,59	131,73	26,28	144,66	65,56	131,70	26,10	142,17	66,89	130,78	26,30
5%	143,64	63,59	131,68	26,37	144,66	65,56	130,77	26,43	142,17	66,89	131,68	26,28
2%	143,64	63,59	131,17	26,27	144,66	65,56	131,12	26,25	142,17	66,89	131,19	26,37
1%	143,64	63,59	124,56	26,22	144,66	65,56	125,89	26,16	142,17	66,89	124,51	26,29
MÉDIA	143,64	63,59	131,16	26,21	144,66	65,56	130,97	25,90	142,17	66,89	130,79	26,15

Fonte: elaborado pelo próprio autor.

Tabela A. 13 - Medições de consumo de RAM e CPU na topologia do Pixel Vizinho.

PIXEL VIZINHO												
COEFS. MANTIDOS	FIGURA 13				FIGURA 14				FIGURA 15			
	BASE		COMPRESSÃO		BASE		COMPRESSÃO		BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
95%	91,42	67,08	131,80	26,29	122,78	60,40	131,74	26,16	94,25	66,09	131,82	25,97
90%	91,42	67,08	131,80	26,17	122,78	60,40	131,80	26,31	94,25	66,09	131,79	25,42
80%	91,42	67,08	131,78	26,14	122,78	60,40	131,79	26,03	94,25	66,09	131,78	26,19
70%	91,42	67,08	131,78	26,43	122,78	60,40	131,73	26,18	94,25	66,09	131,81	25,57
60%	91,42	67,08	131,79	26,21	122,78	60,40	131,75	26,12	94,25	66,09	131,75	25,77
50%	91,42	67,08	131,71	25,97	122,78	60,40	131,72	26,13	94,25	66,09	131,75	25,12
40%	91,42	67,08	131,72	25,77	122,78	60,40	131,68	26,09	94,25	66,09	131,72	25,93
30%	91,42	67,08	131,85	26,00	122,78	60,40	131,90	26,28	94,25	66,09	131,84	25,64
20%	91,42	67,08	131,79	26,17	122,78	60,40	131,79	25,83	94,25	66,09	131,77	26,10
10%	91,42	67,08	131,74	26,30	122,78	60,40	131,71	26,18	94,25	66,09	127,11	25,56
5%	91,42	67,08	126,59	26,03	122,78	60,40	123,05	25,44	94,25	66,09	131,70	26,43
2%	91,42	67,08	131,63	26,59	122,78	60,40	132,01	26,32	94,25	66,09	131,64	26,29
1%	91,42	67,08	129,86	26,17	122,78	60,40	129,64	25,80	94,25	66,09	129,87	26,21
MÉDIA	91,42	67,08	131,22	26,17	122,78	60,40	130,95	26,07	94,25	66,09	131,26	25,86

Fonte: elaborado pelo próprio autor.

Tabela A. 14 - Medições de consumo de RAM e CPU na topologia do Pixel Vizinho + Gêmeo.

PIXEL VIZINHO + GÊMEO												
COEFS. MANTIDOS	FIGURA 13				FIGURA 14				FIGURA 15			
	BASE		COMPRESSÃO		BASE		COMPRESSÃO		BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
95%	134,96	47,47	130,85	26,04	139,95	48,48	130,86	26,11	141,24	49,53	131,80	26,12
90%	134,96	47,47	131,77	26,32	139,95	48,48	130,90	26,00	141,24	49,53	131,79	26,09
80%	134,96	47,47	131,78	25,93	139,95	48,48	131,78	25,91	141,24	49,53	130,85	26,24
70%	134,96	47,47	131,77	26,37	139,95	48,48	131,77	26,14	141,24	49,53	131,79	26,17
60%	134,96	47,47	131,81	26,32	139,95	48,48	130,73	26,03	141,24	49,53	131,75	26,24
50%	134,96	47,47	131,82	26,27	139,95	48,48	130,79	26,03	141,24	49,53	131,73	26,18
40%	134,96	47,47	131,77	26,28	139,95	48,48	131,75	26,43	141,24	49,53	131,76	26,33
30%	134,96	47,47	131,88	26,34	139,95	48,48	132,03	26,25	141,24	49,53	130,95	26,32
20%	134,96	47,47	131,82	26,28	139,95	48,48	131,76	26,22	141,24	49,53	131,80	26,31
10%	134,96	47,47	131,82	26,28	139,95	48,48	130,59	26,12	141,24	49,53	130,81	26,39
5%	134,96	47,47	131,48	26,26	139,95	48,48	130,74	26,13	141,24	49,53	131,52	26,13
2%	134,96	47,47	131,13	26,41	139,95	48,48	131,09	25,87	141,24	49,53	131,10	26,20
1%	134,96	47,47	115,61	25,99	139,95	48,48	115,65	26,41	141,24	49,53	115,43	26,38
MÉDIA	134,96	47,47	130,41	26,24	139,95	48,48	130,03	26,13	141,24	49,53	130,24	26,24

Fonte: elaborado pelo próprio autor.

Tabela A. 15 - Medições de consumo de RAM e CPU na topologia da Vizinhança Gêmea.

VIZINHANÇA GÊMEA												
COEFS. MANTIDOS	FIGURA 13				FIGURA 14				FIGURA 15			
	BASE		COMPRESSÃO		BASE		COMPRESSÃO		BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
95%	198,73	20,43	131,33	25,97	222,30	20,48	131,32	26,12	192,10	20,46	131,30	26,09
90%	198,73	20,43	131,36	26,17	222,30	20,48	131,30	25,81	192,10	20,46	131,30	26,21
80%	198,73	20,43	131,29	26,37	222,30	20,48	131,29	26,11	192,10	20,46	131,30	25,77
70%	198,73	20,43	131,32	26,28	222,30	20,48	131,31	26,32	192,10	20,46	131,29	26,19
60%	198,73	20,43	131,26	26,35	222,30	20,48	131,28	25,88	192,10	20,46	131,29	26,26
50%	198,73	20,43	131,32	26,21	222,30	20,48	131,27	26,12	192,10	20,46	131,32	26,38
40%	198,73	20,43	131,29	26,32	222,30	20,48	131,29	25,89	192,10	20,46	131,26	26,39
30%	198,73	20,43	131,39	26,27	222,30	20,48	131,38	26,10	192,10	20,46	131,38	25,91
20%	198,73	20,43	131,30	26,19	222,30	20,48	131,32	25,85	192,10	20,46	131,33	26,36
10%	198,73	20,43	131,28	26,33	222,30	20,48	131,17	26,27	192,10	20,46	131,30	26,36
5%	198,73	20,43	131,23	26,54	222,30	20,48	131,22	26,47	192,10	20,46	131,25	26,43
2%	198,73	20,43	130,70	26,44	222,30	20,48	130,63	26,12	192,10	20,46	130,69	26,34
1%	198,73	20,43	65,34	26,37	222,30	20,48	65,32	26,10	192,10	20,46	65,27	26,38
MÉDIA	198,73	20,43	126,19	26,29	222,30	20,48	126,16	26,09	192,10	20,46	126,18	26,24

Fonte: elaborado pelo próprio autor.

Tabela A. 16 - Medições de consumo de RAM e CPU na topologia do Pixel Gêmeo Proporcional.

PIXEL GÊMEO PROPORCIONAL												
COEFS. MANTIDOS	FIGURA 13				FIGURA 14				FIGURA 15			
	BASE		COMPRESSÃO		BASE		COMPRESSÃO		BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
95%	138,19	50,72	131,84	25,93	146,20	49,73	130,88	26,18	140,54	47,97	130,88	26,28
90%	138,19	50,72	130,78	25,98	146,20	49,73	131,50	26,28	140,54	47,97	131,79	25,89
80%	138,19	50,72	131,77	26,25	146,20	49,73	131,78	26,12	140,54	47,97	130,84	26,12
70%	138,19	50,72	130,84	26,03	146,20	49,73	131,76	26,38	140,54	47,97	131,77	25,72
60%	138,19	50,72	131,75	26,41	146,20	49,73	130,81	26,19	140,54	47,97	131,76	26,14
50%	138,19	50,72	131,75	26,12	146,20	49,73	130,80	26,25	140,54	47,97	131,76	26,15
40%	138,19	50,72	131,72	26,04	146,20	49,73	131,76	26,54	140,54	47,97	130,82	26,26
30%	138,19	50,72	131,78	26,07	146,20	49,73	131,85	26,53	140,54	47,97	130,91	25,77
20%	138,19	50,72	131,84	26,21	146,20	49,73	131,89	26,28	140,54	47,97	131,78	26,15
10%	138,19	50,72	115,57	26,07	146,20	49,73	131,78	26,48	140,54	47,97	131,53	26,06
5%	138,19	50,72	130,80	26,22	146,20	49,73	131,61	26,55	140,54	47,97	131,73	26,28
2%	138,19	50,72	131,63	26,18	146,20	49,73	128,12	26,64	140,54	47,97	130,68	26,01
1%	138,19	50,72	126,02	26,21	146,20	49,73	126,03	26,47	140,54	47,97	126,10	26,16
MÉDIA	138,19	50,72	129,85	26,13	146,20	49,73	130,81	26,38	140,54	47,97	130,95	26,08

Fonte: elaborado pelo próprio autor.

Tabela A. 17 - Medições de consumo de RAM e CPU na topologia da Vizinhança Gêmea Proporcional.

VIZINHANÇA GÊMEA PROPORCIONAL												
COEFS. MANTIDOS	FIGURA 13				FIGURA 14				FIGURA 15			
	BASE		COMPRESSÃO		BASE		COMPRESSÃO		BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
95%	222,85	38,19	131,33	26,09	223,68	38,69	131,32	26,12	222,89	37,93	131,33	25,74
90%	222,85	38,19	131,34	26,26	223,68	38,69	131,33	25,88	222,89	37,93	130,20	26,12
80%	222,85	38,19	131,34	26,10	223,68	38,69	131,33	26,01	222,89	37,93	131,31	26,18
70%	222,85	38,19	131,01	25,91	223,68	38,69	131,30	26,02	222,89	37,93	131,29	25,76
60%	222,85	38,19	131,30	25,97	223,68	38,69	131,30	26,03	222,89	37,93	131,28	25,82
50%	222,85	38,19	131,28	26,24	223,68	38,69	131,29	26,25	222,89	37,93	131,29	26,03
40%	222,85	38,19	131,27	26,18	223,68	38,69	131,27	26,40	222,89	37,93	131,28	26,00
30%	222,85	38,19	131,41	26,03	223,68	38,69	131,36	26,42	222,89	37,93	131,39	26,12
20%	222,85	38,19	124,47	26,17	223,68	38,69	124,29	26,30	222,89	37,93	131,32	25,99
10%	222,85	38,19	131,27	26,18	223,68	38,69	131,29	26,18	222,89	37,93	131,31	26,38
5%	222,85	38,19	131,25	26,34	223,68	38,69	131,26	26,21	222,89	37,93	131,27	26,27
2%	222,85	38,19	130,54	26,29	223,68	38,69	130,62	26,33	222,89	37,93	130,59	26,03
1%	222,85	38,19	71,76	26,04	223,68	38,69	71,84	26,09	222,89	37,93	64,67	26,28
MÉDIA	222,85	38,19	126,12	26,14	223,68	38,69	126,14	26,17	222,89	37,93	126,04	26,06

Fonte: elaborado pelo próprio autor.

Tabela A. 18 - Medições de consumo de RAM e CPU na 2D-DCT.

TRANSFORMADA DISCRETA DO COSSENO BIDIMENSIONAL												
COEFS. MANTIDOS	FIGURA 13				FIGURA 14				FIGURA 15			
	BASE		COMPRESSÃO		BASE		COMPRESSÃO		BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
95%	0,00	0,00	4,58	8,33	0,00	0,00	8,03	8,24	0,00	0,00	7,02	8,34
90%	0,00	0,00	4,02	8,28	0,00	0,00	7,03	8,33	0,00	0,00	7,02	8,33
80%	0,00	0,00	4,45	8,25	0,00	0,00	8,04	8,33	0,00	0,00	5,90	8,33
70%	0,00	0,00	4,58	8,32	0,00	0,00	6,03	8,31	0,00	0,00	7,02	8,33
60%	0,00	0,00	4,02	8,28	0,00	0,00	6,04	8,34	0,00	0,00	7,02	8,33
50%	0,00	0,00	4,45	8,32	0,00	0,00	5,91	8,33	0,00	0,00	8,02	8,33
40%	0,00	0,00	4,58	8,30	0,00	0,00	8,02	8,25	0,00	0,00	8,15	8,38
30%	0,00	0,00	4,02	8,28	0,00	0,00	8,02	8,15	0,00	0,00	8,02	8,34
20%	0,00	0,00	4,46	8,37	0,00	0,00	8,02	8,22	0,00	0,00	7,02	8,29
10%	0,00	0,00	5,02	8,33	0,00	0,00	7,98	8,22	0,00	0,00	7,02	8,33
5%	0,00	0,00	5,02	8,33	0,00	0,00	10,02	8,25	0,00	0,00	7,02	8,26
2%	0,00	0,00	4,57	8,32	0,00	0,00	11,04	8,28	0,00	0,00	7,01	8,22
1%	0,00	0,00	0,00	8,34	0,00	0,00	2,77	8,28	0,00	0,00	0,00	8,15
MÉDIA	0,00	0,00	4,14	8,31	0,00	0,00	7,46	8,27	0,00	0,00	6,63	8,30

Fonte: elaborado pelo próprio autor.

Tabela A. 19 - Média das medições de RAM e CPU das três imagens de teste para diferentes métodos.

MÉDIA DAS IMAGENS				
TOPOLOGIA	BASE		COMPRESSÃO	
	RAM (MB)	CPU (%)	RAM (MB)	CPU (%)
PIXEL GÊMEO	143,49	65,35	130,97	26,09
PIXEL VIZINHO	138,72	48,49	131,43	26,20
PIXEL VIZ. + GÊMEO	138,72	48,49	130,23	26,20
VIZINHANÇA GÊMEA	204,38	20,46	126,17	26,21
P. GÊMEO PROP.	141,64	49,47	130,54	26,19
VIZ. GÊMEA PROP.	223,14	38,27	126,10	26,12
2D-DCT	0,00	0,00	6,08	8,30

Fonte: elaborado pelo próprio autor.

Tabela A. 20 - Medições do tempo usado em segundos para a topologia do Pixel Gêmeo nas diferentes imagens.

PIXEL GÊMEO						
COEFS. MANTIDOS	FIGURA 13		FIGURA 14		FIGURA 15	
	BASE	COMPRESSÃO	BASE	COMPRESSÃO	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)
95%	5,18	170,58	5,16	170,09	5,15	172,64
90%	5,18	170,44	5,16	172,62	5,15	170,78
80%	5,18	170,51	5,16	177,67	5,15	172,23
70%	5,18	170,44	5,16	172,96	5,15	170,05
60%	5,18	170,30	5,16	169,21	5,15	171,72
50%	5,18	170,03	5,16	171,03	5,15	169,90
40%	5,18	170,16	5,16	168,88	5,15	171,66
30%	5,18	169,92	5,16	171,08	5,15	169,77
20%	5,18	169,71	5,16	168,39	5,15	170,46
10%	5,18	169,14	5,16	168,08	5,15	169,20
5%	5,18	169,24	5,16	167,63	5,15	170,22
2%	5,18	169,00	5,16	168,24	5,15	169,14
1%	5,18	169,19	5,16	167,70	5,15	169,86
MÉDIA	5,18	169,90	5,16	170,28	5,15	170,59

Fonte: elaborado pelo próprio autor.

Tabela A. 21 - Medições do tempo usado em segundos para a topologia do Pixel Vizinho nas diferentes imagens.

PIXEL VIZINHO						
COEFS. MANTIDOS	FIGURA 13		FIGURA 14		FIGURA 15	
	BASE	COMPRESSÃO	BASE	COMPRESSÃO	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)
95%	5,53	172,45	5,74	173,08	5,56	172,70
90%	5,53	172,60	5,74	172,92	5,56	175,10
80%	5,53	172,40	5,74	172,75	5,56	172,06
70%	5,53	172,58	5,74	172,32	5,56	175,10
60%	5,53	172,30	5,74	172,86	5,56	172,41
50%	5,53	172,54	5,74	172,83	5,56	175,64
40%	5,53	173,10	5,74	172,45	5,56	172,53
30%	5,53	172,36	5,74	172,63	5,56	174,80
20%	5,53	172,28	5,74	172,75	5,56	172,14
10%	5,53	172,20	5,74	172,76	5,56	175,28
5%	5,53	169,89	5,74	174,44	5,56	168,05
2%	5,53	167,18	5,74	171,95	5,56	167,18
1%	5,53	167,52	5,74	173,28	5,56	167,65
MÉDIA	5,53	171,49	5,74	172,85	5,56	172,36

Fonte: elaborado pelo próprio autor.

Tabela A. 22 - Medições do tempo usado em segundos para a topologia do Pixel Vizinho + Gêmeo nas diferentes imagens.

PIXEL VIZINHO + GÊMEO						
COEFS. MANTIDOS	FIGURA 13		FIGURA 14		FIGURA 15	
	BASE	COMPRESSÃO	BASE	COMPRESSÃO	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)
95%	7,52	170,18	7,51	170,42	7,52	170,55
90%	7,52	170,13	7,51	170,27	7,52	170,52
80%	7,52	170,12	7,51	170,33	7,52	170,33
70%	7,52	170,07	7,51	170,55	7,52	170,43
60%	7,52	169,96	7,51	170,64	7,52	170,38
50%	7,52	169,80	7,51	169,59	7,52	169,74
40%	7,52	169,32	7,51	169,70	7,52	169,33
30%	7,52	169,24	7,51	169,72	7,52	169,32
20%	7,52	169,24	7,51	169,12	7,52	169,54
10%	7,52	169,15	7,51	169,24	7,52	169,39
5%	7,52	167,93	7,51	167,65	7,52	167,93
2%	7,52	168,04	7,51	168,73	7,52	167,49
1%	7,52	167,64	7,51	167,29	7,52	167,44
MÉDIA	7,52	169,29	7,51	169,48	7,52	169,41

Fonte: elaborado pelo próprio autor.

Tabela A. 23 - Medições do tempo usado em segundos para a topologia da Vizinhança Gêmea nas diferentes imagens.

VIZINHANÇA GÊMEA						
COEFS. MANTIDOS	FIGURA 13		FIGURA 14		FIGURA 15	
	BASE	COMPRESSÃO	BASE	COMPRESSÃO	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)
95%	24,65	170,84	24,79	170,87	24,77	170,64
90%	24,65	170,67	24,79	170,77	24,77	170,63
80%	24,65	170,76	24,79	170,98	24,77	170,65
70%	24,65	170,34	24,79	170,60	24,77	170,25
60%	24,65	170,70	24,79	171,03	24,77	170,26
50%	24,65	170,44	24,79	170,68	24,77	170,07
40%	24,65	170,47	24,79	170,77	24,77	170,16
30%	24,65	169,89	24,79	170,45	24,77	170,00
20%	24,65	169,59	24,79	170,52	24,77	169,21
10%	24,65	169,19	24,79	169,53	24,77	169,09
5%	24,65	169,15	24,79	169,60	24,77	169,27
2%	24,65	169,20	24,79	169,60	24,77	169,02
1%	24,65	169,06	24,79	169,44	24,77	169,11
MÉDIA	24,65	170,02	24,79	170,37	24,77	169,87

Fonte: elaborado pelo próprio autor.

Tabela A. 24 - Medições do tempo usado em segundos para a topologia do Pixel Gêmeo Proporcional nas diferentes imagens.

PIXEL GÊMEO PROPORCIONAL						
COEFS. MANTIDOS	FIGURA 13		FIGURA 14		FIGURA 15	
	BASE	COMPRESSÃO	BASE	COMPRESSÃO	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)
95%	7,31	171,05	7,51	170,86	7,29	170,28
90%	7,31	170,03	7,51	169,97	7,29	172,04
80%	7,31	170,02	7,51	170,32	7,29	170,41
70%	7,31	169,96	7,51	169,98	7,29	171,92
60%	7,31	170,06	7,51	169,93	7,29	169,75
50%	7,31	170,04	7,51	170,18	7,29	170,46
40%	7,31	170,45	7,51	169,14	7,29	169,41
30%	7,31	169,86	7,51	168,72	7,29	170,35
20%	7,31	169,84	7,51	168,67	7,29	169,35
10%	7,31	169,11	7,51	168,49	7,29	169,34
5%	7,31	168,12	7,51	168,63	7,29	167,76
2%	7,31	168,19	7,51	167,82	7,29	169,10
1%	7,31	168,25	7,51	167,37	7,29	167,92
MÉDIA	7,31	169,61	7,51	169,24	7,29	169,85

Fonte: elaborado pelo próprio autor.

Tabela A. 25 - Medições do tempo usado em segundos para a topologia da Vizinhança Gêmea Proporcional nas diferentes imagens.

VIZINHANÇA GÊMEA PROPORCIONAL						
COEFS. MANTIDOS	FIGURA 13		FIGURA 14		FIGURA 15	
	BASE	COMPRESSÃO	BASE	COMPRESSÃO	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)
95%	10,02	170,44	10,02	170,38	10,05	171,58
90%	10,02	170,32	10,02	170,59	10,05	171,57
80%	10,02	170,12	10,02	170,46	10,05	171,07
70%	10,02	170,00	10,02	170,08	10,05	171,59
60%	10,02	168,56	10,02	169,54	10,05	171,21
50%	10,02	168,31	10,02	169,42	10,05	170,86
40%	10,02	168,31	10,02	169,45	10,05	170,98
30%	10,02	168,43	10,02	169,39	10,05	169,98
20%	10,02	168,32	10,02	169,51	10,05	169,97
10%	10,02	168,07	10,02	167,91	10,05	169,95
5%	10,02	167,81	10,02	167,57	10,05	169,91
2%	10,02	167,88	10,02	167,55	10,05	169,84
1%	10,02	167,87	10,02	167,54	10,05	168,28
MÉDIA	10,02	168,80	10,02	169,18	10,05	170,52

Fonte: elaborado pelo próprio autor.

Tabela A. 26 - Medições do tempo usado em segundos para o método da 2D-DCT nas diferentes imagens.

2D-DCT						
COEFS. MANTIDOS	FIGURA 13		FIGURA 14		FIGURA 15	
	BASE	COMPRESSÃO	BASE	COMPRESSÃO	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)	TEMPO (s)
95%	0,00	13,57	0,00	13,90	0,00	13,27
90%	0,00	13,92	0,00	13,58	0,00	13,29
80%	0,00	14,98	0,00	13,49	0,00	13,37
70%	0,00	14,30	0,00	13,64	0,00	13,25
60%	0,00	14,84	0,00	13,29	0,00	13,35
50%	0,00	15,45	0,00	13,37	0,00	13,63
40%	0,00	15,56	0,00	13,60	0,00	13,40
30%	0,00	13,55	0,00	13,83	0,00	13,43
20%	0,00	13,50	0,00	13,46	0,00	13,30
10%	0,00	13,24	0,00	13,43	0,00	13,61
5%	0,00	13,40	0,00	13,23	0,00	14,22
2%	0,00	13,30	0,00	13,38	0,00	14,42
1%	0,00	13,52	0,00	13,49	0,00	14,29
MÉDIA	0,00	14,09	0,00	13,51	0,00	13,60

Fonte: elaborado pelo próprio autor.

Tabela A. 27 - Média das medições de tempo das três imagens de teste para diferentes métodos.

MÉDIA DAS IMAGENS		
TOPOLOGIA	BASE	COMPRESSÃO
	TEMPO (s)	TEMPO (s)
PIXEL GÊMEO	5,16	170,25
PIXEL VIZINHO	5,61	172,23
PIXEL VIZ. + GÊMEO	7,52	169,40
VIZINHANÇA GÊMEA	24,74	170,09
P. GÊMEO PROP.	7,37	169,57
VIZ. GÊMEA PROP.	10,03	169,50
2D-DCT	0,00	13,73

Fonte: elaborado pelo próprio autor.