



UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE MATEMÁTICA



FERNANDO VENÂNCIO AIRES JÚNIOR

APLICAÇÃO DA ÁLGEBRA LINEAR NO TRATAMENTO DE IMAGENS

SÃO CARLOS
2024

Junior, Fernando Venancio Aires

Aplicação da álgebra linear no tratamento de imagens /
Fernando Venancio Aires Junior -- 2024.
65f.

TCC (Graduação) - Universidade Federal de São Carlos,
campus São Carlos, São Carlos
Orientador (a): Humberto Luiz Talpo
Banca Examinadora: Yuriko Baldin, José Nazareno Vieira
Gomes
Bibliografia

1. Matemática. I. Junior, Fernando Venancio Aires. II.
Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática
(SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Arildo Martins - CRB/8 7180

FERNANDO VENÂNCIO AIRES JÚNIOR

APLICAÇÃO DA ÁLGEBRA LINEAR NO TRATAMENTO DE IMAGENS

Monografia apresentada ao Curso de Licenciatura em Matemática da Universidade Federal de São Carlos.

Orientador: Prof. Dr. Humberto Luiz Talpo

SÃO CARLOS
2024

*Dedico este trabalho à minha família, aos meus amigos,
e ao meu orientador Talpo.*

AGRADECIMENTOS

Agradeço profundamente à minha mãe, Maria Eva, por tudo o que fez por mim desde o momento em que nasci. Agradeço pelo cuidado constante, pelo apoio incondicional e por sempre estar ao meu lado em cada momento da minha vida.

Às minhas irmãs, que são pilares essenciais na minha trajetória, agradeço pela confiança imensurável que depositam em mim e nas minhas capacidades, tornando minha caminhada ainda mais segura e significativa.

Aos meus amigos, agradeço de coração, pois são indispensáveis na minha vida. A vida se torna ainda mais rica e valiosa na companhia de pessoas tão especiais. Em particular, agradeço ao meu amigo Gabriel Malaquias, cuja ajuda com a linguagem de programação Python foi crucial para a realização deste trabalho.

Por fim, gostaria de expressar minha sincera gratidão ao meu orientador, Talpo. Agradeço não apenas pela contribuição fundamental no desenvolvimento do texto e pela sua imensa expertise, mas também pela compreensão ao longo de toda a jornada. Sua paciência foram inspirações constantes durante esse processo.

E, por fim, a todos que, de alguma forma, contribuíram para a minha formação e para o desenvolvimento da pessoa que me tornei ao longo dessa jornada.

RESUMO

Este Trabalho de Conclusão de Curso explora a aplicação da Álgebra Linear no processamento de imagens digitais, mostrando como operações matriciais, como soma, multiplicação e convolução, podem ser usadas para aprimorar a análise visual e a qualidade de imagens. O trabalho destaca a importância das operações de convolução nos domínios espacial e da frequência para realçar bordas, texturas e reduzir ruídos. Abordando tanto o domínio espacial quanto o da frequência, o TCC ilustra a relevância dos conceitos matemáticos no processamento de imagens e sua aplicabilidade na interpretação e extração de informações visuais significativas.

Palavras-chave: Tratamento de imagens. Domínio espacial. Domínio da frequência. Representação de imagens digitais. Álgebra linear.

SUMÁRIO

1	INTRODUÇÃO	7
2	EXPLORAÇÃO E INTERPRETAÇÃO DE IMAGENS DIGITAIS: ABORDAGENS DE REPRESENTAÇÃO	8
2.1	INTRODUÇÃO À IMAGEM DIGITAL	8
2.2	PIXELS E MATRIZ DE IMAGEM: A CONSTRUÇÃO DA REALIDADE DIGITAL	8
2.3	PROFUNDIDADE DE COR E ESCALA DE CINZA: EXPLORANDO A PALETA VISUAL	9
2.4	OUTROS SISTEMAS DE COR: ALÉM DO RGB	12
2.4.1	Sistema CMYK: Ciano, Magenta, Amarelo, Key (Black)	12
2.4.2	Sistema HSL/HSV: Matiz, Saturação e Luminosidade/Valor	12
2.4.3	Sistema de Cor de Luminância e Crominância (YUV/YCbCr)	12
2.4.4	Comparando Sistemas de Cor	13
2.5	FORMATOS DE IMAGEM: IMPACTOS NO ARMAZENAMENTO E COMPATIBILIDADE	13
2.5.1	Formato JPEG (Joint Photographic Experts Group)	13
2.5.2	Formato PNG (Portable Network Graphics)	13
2.5.3	Formato GIF (Graphics Interchange Format)	14
2.5.4	Formato TIFF (Tagged Image File Format)	14
2.5.5	Formato BMP (Bitmap)	14
3	PASSOS ESSENCIAIS PARA O PROCESSAMENTO DA IMAGEM DIGITAL	15
3.1	AQUISIÇÃO	15
3.2	ARMAZENAMENTO	15
3.3	PROCESSAMENTO	16
3.4	COMUNICAÇÃO	16
3.5	EXIBIÇÃO	17
4	MÉTODOS DE PROCESSAMENTO DE IMAGENS	18
4.1	SOMA DE MATRIZES	20
4.2	SUBTRAÇÃO DE MATRIZES	23
4.3	MULTIPLICAÇÃO DE MATRIZES	23
4.4	CONVOLUÇÃO	26
4.4.1	Filtro de Média	28
4.4.2	Filtro de Realce de Bordas	29

4.4.3	Filtro de Detecção de Bordas	31
5	DOMÍNIO DA FREQUÊNCIA	34
5.1	TRANSFORMADA DE FOURIER	34
5.2	A TRANSFORMADA DISCRETA DE FOURIER (TDF)	37
5.3	PROPRIEDADES DA TRANSFORMADA BIDIMENSIONAL DE FOURIER	43
5.3.1	Separabilidade	45
5.3.2	Translação	46
5.3.3	Deslocamento e Translação na Transformada de Fourier	46
5.3.4	Simetria Conjugada	48
5.3.5	Rotação	48
5.3.6	Linearidade	49
5.3.7	Convolução	49
5.4	TRANSFORMADA RÁPIDA DE FOURIER (FFT)	50
5.5	TRATAMENTOS DE IMAGENS USANDO FFT	50
5.5.1	Filtro Passa-Baixa	53
5.5.2	Filtro Passa-Alta	53
5.5.3	Filtro Detector de Bordas (Filtro de Sobel)	55
5.5.4	Outros Exemplos de Tratamentos Bem-Sucedidos	56
6	CONSIDERAÇÕES FINAIS	58
	REFERÊNCIAS	59
	APÊNDICE A CÓDIGOS UTILIZADOS NO TRATAMENTO DE IMAGENS NO DOMÍNIO DA FREQUÊNCIA	60

1 INTRODUÇÃO

Ao investigar a representação de imagens digitais, elucidamos as nuances do sistema RGB e a relevância dos pixels como blocos fundamentais de informação visual. Demonstramos a correspondência direta entre as imagens digitais e as matrizes, onde cada pixel é codificado como um elemento dessa matriz. Esse entendimento prévio foi crucial para nos lançarmos aos desafios subsequentes.

No segundo capítulo, abordamos os cinco passos centrais do processamento de imagens: aquisição, armazenamento, processamento, comunicação e exibição. Exploramos cada etapa, desde a coleta de dados visuais até a apresentação final, destacando tecnologias e conceitos envolvidos. Aprofundamos o entendimento das complexidades inerentes a cada fase, incluindo captura, conversão digital, armazenamento eficiente, técnicas de processamento, transferência e visualização. Esse capítulo busca oferecer uma compreensão abrangente do ciclo completo de vida das imagens digitais, delineando sua importância em diversos contextos e destacando as interconexões essenciais entre as etapas.

Em seguida, no terceiro capítulo, abordamos a fundamentação matemática essencial que sustenta os métodos de processamento de imagens. Estabelecemos um alicerce sólido ao explorar conceitos-chave da Álgebra Linear, que se mostraram pré-requisitos indispensáveis para a compreensão das técnicas subsequentes. Essa introdução teórica foi crucial para apreciar a manipulação matricial que subjaz a métodos como soma, subtração, multiplicação de matrizes e o fundamental método de convolução.

A convolução emergiu como um elemento central do nosso estudo, revelando-se uma técnica muito importante no processamento de imagens. Investigamos a aplicação dessa operação, considerando máscaras específicas, como a máscara de Sobel, que desempenham um papel crucial na detecção de bordas e características em imagens. Ao experimentar diversas combinações de máscaras, compreendemos como os coeficientes podem ser ajustados para alterar drasticamente os resultados visuais.

Portanto, chegamos ao cerne de nossa pesquisa, onde as operações matriciais se harmonizam com o processamento de imagens. Cada capítulo contribuiu para essa narrativa, destacando a importância da base matemática e a aplicação prática das técnicas no aprimoramento e análise de imagens.

Durante o curso deste estudo, o último estágio trouxe à tona uma realização crucial: a compreensão da relevância da definição do produto usual de matrizes em sua forma atual. Descobrimos que essa definição permite expressar composições como produtos de matrizes, uma propriedade que não se estende à convolução. Esse discernimento surgiu da necessidade do método de Sobel exigir duas convoluções consecutivas. Essa percepção destacou ainda mais a natureza intrincada da matemática subjacente ao processamento de imagens, fortalecendo nossa compreensão do campo e sua interconexão com a Álgebra Linear.

2 EXPLORAÇÃO E INTERPRETAÇÃO DE IMAGENS DIGITAIS: ABORDAGENS DE REPRESENTAÇÃO

2.1 INTRODUÇÃO À IMAGEM DIGITAL

Para iniciar nossa jornada no mundo da representação de imagens digitais, é crucial compreender os fundamentos dessa forma de expressão visual. Em um mundo cada vez mais digitalizado, a imagem digital se tornou uma parte inseparável das nossas vidas. Mas o que exatamente é uma imagem digital?

Uma imagem digital, em contraste com formas físicas como pinturas e fotografias, é uma expressão visual codificada e transmitida em formato digital. Ela é construída por pixels, que são pontos minúsculos que se unem para criar a imagem que percebemos. A essência da representação digital está na quantização dos atributos visuais em valores numéricos discretos, permitindo flexibilidade na edição e processamento. A representação ocorre em matrizes de pixels, onde cada a_{ij} carrega informações de cor, intensidade e transparência. Ao explorar isso, entramos no mundo complexo do processamento de imagens e suas infinitas possibilidades de manipulação.

2.2 PIXELS E MATRIZ DE IMAGEM: A CONSTRUÇÃO DA REALIDADE DIGITAL

Nessa seção vamos nos aprofundar na essência dos elementos que compõem essa forma única de expressão visual: os pixels e a matriz de imagem. Esses fundamentos são os blocos de construção que dão vida ao mundo visual que experimentamos através de nossos dispositivos eletrônicos.

No cerne dessa representação está a matriz de imagem. Ela é a estrutura organizacional que contém todos os pixels e seus respectivos valores. Cada pixel é atribuído a uma posição específica na matriz, correspondendo às coordenadas espaciais da imagem. As linhas e colunas da matriz representam a grade que compõe a imagem, e a interseção entre elas é o lugar onde cada pixel reside.

A resolução da imagem, frequentemente expressa em termos de largura e altura da matriz, determina a quantidade de detalhes visuais que podem ser capturados e exibidos. Uma matriz maior, com mais pixels, resulta em imagens mais detalhadas, enquanto uma matriz menor produz imagens mais suavizadas.

Vejamos um exemplo. A seguir, uma imagem representada por uma matriz de dimensão 220 linhas e 220 colunas seguida da mesma imagem representada por uma matriz quadrada 50×50 .

Figura 2.1 – Capa do álbum Theatre of Fate



Fonte: Site oficial¹

2.3 PROFUNDIDADE DE COR E ESCALA DE CINZA: EXPLORANDO A PALETA VISUAL

Nossa exploração da representação de imagens digitais nos leva a um reino de cores, intensidades e matrizes. A profundidade de cor e a escala de cinza são as ferramentas que moldam a riqueza visual das imagens digitais, adicionando dimensão e nuances à nossa experiência visual.

Em uma imagem digital colorida, a profundidade de cor determina quantas cores únicas podem ser representadas. Essa profundidade é expressa em bits e define a quantidade de informações disponíveis para cada canal de cor.

Vamos entender melhor o que é o bit. Os computadores utilizam o sistema binário, que é um sistema numérico baseado em dois dígitos: 0 e 1. Cada dígito binário é chamado de “bit” (contração de “binary digit” em inglês). Portanto, o sistema binário é um sistema de base 2.

Agora, vamos pensar no processo de cálculo das combinações possíveis.

Para entender quantas combinações de números diferentes podemos representar usando 8 bits, vamos considerar que cada bit é uma posição onde podemos colocar um 0 ou um 1. Para cada posição, temos duas opções possíveis (0 ou 1). Portanto, para calcular o número total de combinações possíveis, multiplicamos o número de opções em cada posição (2) pelo número de posições (8 bits):

¹ Fonte: Site oficial. Disponível em: <<https://viperbrazil.com.br/site/discografia-completa/>>. Acesso em: 17 de ago. 2023.

$$2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256.$$

Normalmente para representar uma imagem colorida é usado o sistema RGB que possui três canais de cores, disso, temos então, um total de $256 \times 256 \times 256 = 256^3 = 16.777.216$ cores diferentes possíveis para uma imagem colorida de de 8 bits.

As siglas RGB significam, respectivamente: vermelho, verde e azul. Logo, podemos pensar na imagem colorida como sendo uma matriz tridimensional. Segue alguns exemplos das cores mais comuns:

Figura 2.2 – Branco - (255, 255, 255), Azul - (0, 0, 255) e Vermelho - (255, 0, 0).



Fonte: Autoria própria.

Figura 2.3 – Verde - (0, 255, 0), Amarelo - (255, 255, 0) e Magenta - (255, 0, 255).



Fonte: Autoria própria.

Figura 2.4 – Ciano - (0, 255, 255) e Preto (0, 0, 0).



Fonte: Autoria própria

Se o que possibilita toda essa gama de cores é dado pela quantidade de bits que é usado para determinar uma cor (8 bits), então por qual motivo não usamos simplesmente, por exemplo, 10, 12 ou mesmo 16 bits? Um dos motivos é a memória.

Uma imagem em Full HD de $1920p \times 1080p$ no sistema RGB, onde cada pixel é composto por três canais, ocupa aproximadamente 6MB de memória, mas não é um valor preciso, já que esse valor depende de vários fatores, incluindo a qualidade da extensão e o formato da imagem.

O cálculo é o seguinte: cada canal possui a profundidade de 8 bits, o que nos dá então 24 bits totais por pixel ou $\frac{24}{8} = 3$ bytes por pixel, dado que um byte, segundo (FEOFILOFF, 2019), é dado por 8 bits; agora basta multiplicar o número total de pixels da imagem pelos bytes. Segue o cálculo:

$$1920 \times 1080 \times 3 = 6.220.800 \text{ bytes.}$$

O que resulta em aproximadamente 6MB. Numa imagem com profundidade de 16 bits teríamos

$$\frac{16 \times 3 \times 1920 \times 1080}{8} = 12.441.600 \text{ bytes.}$$

O que resulta em aproximadamente 12MB.

É interessante ressaltarmos que, por mais que não seja utilizado de forma usual as profundidades que não seja o de 8 bits, na medicina se utiliza 10, 12 e até 16 bits para ter um detalhamento e precisão maior das imagens e as suas nuances. Para efeito de comparação, uma imagem de profundidade de 16 bits possui uma gama de $2^{16} = 65.536$ para cada cor, o que dá impressionantes $65.536^3 = 281.474.976.710.656$ cores diferentes.

Outros fatores de ser usado comumente a profundidade de 8 bits são: processamento mais rápido e eficiente, suficiência para a maioria das aplicações e a ampla compatibilidade destes arquivos com softwares e hardwares.

Por outro lado, nas imagens em escala de cinza, a profundidade de cor define quantos tons de cinza distintos podem ser representados. Utilizando o mesmo princípio de 8 bits, obtemos 256 variações de intensidade luminosa, variando de preto absoluto a branco puro, com toda a gama de tons intermediários. Isso permite a criação de imagens ricas em detalhes, sombras e destaques.

Figura 2.5 – Tons de cinza variando de 20 em 20 tons.



Fonte: Criação do Autor.

Para efeito de comparação, uma imagem Full HD de 1920p por 1080p em escala de cinza que possui um canal, ocupa de memória cerca de somente 2MB.

$$\frac{8 \times 1920 \times 1080}{8} = 2.073.600 \text{ bytes.}$$

2.4 OUTROS SISTEMAS DE COR: ALÉM DO RGB

Além do popular sistema de cor RGB, que combina as cores primárias vermelha, verde e azul, existem outros sistemas de cor que oferecem abordagens únicas para representar cores. Cada sistema foi projetado para atender a diferentes necessidades e contextos, desde a impressão até a especificação intuitiva de cores com base em suas características visuais. Vamos explorar os sistemas CMYK, HSL/HSV e o sistema de cor de luminância e crominância (YUV/YCbCr).

2.4.1 Sistema CMYK: Ciano, Magenta, Amarelo, Key (Black)

O sistema CMYK é amplamente utilizado na indústria gráfica e de impressão. Ele emprega quatro cores primárias: ciano, magenta, amarelo e preto (key). Esse sistema é baseado na subtração de cores, ao contrário do RGB, que se baseia na adição. O ciano absorve vermelho, o magenta absorve verde e o amarelo absorve azul. A combinação dessas cores permite criar uma ampla gama de cores, sendo o preto (key) utilizado para garantir um contraste mais profundo em impressões.

2.4.2 Sistema HSL/HSV: Matiz, Saturação e Luminosidade/Valor

O sistema HSL (Matiz, Saturação e Luminosidade) e o sistema HSV (Matiz, Saturação e Valor) fornecem uma abordagem mais intuitiva para especificar cores com base em suas características visuais.

Matiz (Hue): Representa a cor propriamente dita, com valores variando de 0 a 360 graus. Ele define onde a cor está localizada no círculo cromático, como vermelho, verde, azul, etc.

Saturação (Saturation): Indica a pureza ou intensidade da cor. Valores mais altos resultam em cores mais vívidas, enquanto valores mais baixos tendem a ser mais acinzentados ou pastéis.

Luminosidade (Luminance) ou Valor (Value): Representa a luminosidade da cor. No sistema HSL, a luminosidade controla o brilho da cor, enquanto no sistema HSV, o valor é mais relacionado à intensidade.

2.4.3 Sistema de Cor de Luminância e Crominância (YUV/YCbCr)

O sistema YUV (Luminance e Chrominance) ou YCbCr é comumente usado na transmissão de sinais de vídeo e na compressão de imagens. Ele separa a informação de luminância (Y) da crominância (U e V). A luminância representa o brilho, enquanto a crominância lida com a informação de cor. Esse sistema é vantajoso para a compressão, pois a perda de detalhes de cor (crominância) é menos perceptível aos olhos humanos do que a perda de detalhes de luminância.

2.4.4 Comparando Sistemas de Cor

Cada sistema de cor tem suas vantagens e aplicações específicas. O RGB é amplamente usado em dispositivos eletrônicos, o CMYK na impressão, o HSL/HSV para especificação visual intuitiva (se refere a um método de definir cores com base em características visuais percebidas, em oposição a abordagens puramente numéricas, muito útil para designers gráficos) e o sistema YUV/YCbCr na transmissão de vídeo e compressão de imagem. A escolha do sistema de cor depende do contexto e do objetivo da representação de cor, garantindo que as cores sejam fielmente reproduzidas de acordo com as necessidades específicas de cada campo.

2.5 FORMATOS DE IMAGEM: IMPACTOS NO ARMAZENAMENTO E COMPATIBILIDADE

Os formatos de imagem desempenham um papel crucial na eficiência de armazenamento, compartilhamento e exibição de imagens digitais. Cada formato possui suas próprias características, níveis de compressão e recursos, afetando o tamanho do arquivo, a qualidade da imagem e a compatibilidade com diferentes dispositivos e plataformas. Vamos explorar os principais formatos de imagem e suas características.

2.5.1 Formato JPEG (Joint Photographic Experts Group)

O formato JPEG é um dos formatos de imagem mais amplamente utilizados. Ele oferece uma ótima relação entre qualidade de imagem e tamanho do arquivo, o que o torna ideal para armazenamento e compartilhamento em plataformas online e redes sociais. O JPEG utiliza compressão com perdas, o que significa que ele remove detalhes redundantes para reduzir o tamanho do arquivo. No entanto, a qualidade da imagem pode ser comprometida em níveis de compressão mais altos.

2.5.2 Formato PNG (Portable Network Graphics)

O formato PNG é uma escolha popular para imagens que precisam de alta qualidade e compressão sem perdas. Ele suporta transparência e exibe imagens nítidas e detalhadas. Embora os arquivos PNG sejam geralmente maiores do que os JPEG, eles mantêm a qualidade da imagem intacta. O PNG é amplamente usado para gráficos, logotipos e elementos com bordas nítidas, bem como para imagens com áreas transparentes.

2.5.3 Formato GIF (Graphics Interchange Format)

O formato GIF é conhecido por suportar animações e é amplamente usado para criar imagens animadas e *memes*. Ele utiliza uma paleta de cores limitada, o que resulta em tamanhos de arquivo relativamente pequenos, mas também pode levar a uma qualidade de imagem reduzida em comparação com outros formatos. Devido à sua capacidade de suportar animações, o GIF é usado para criar pequenos vídeos em *loop* para compartilhamento online.

2.5.4 Formato TIFF (Tagged Image File Format)

O formato TIFF é amplamente utilizado para aplicações de alta qualidade, como impressão e edição de imagens profissionais. Ele suporta compressão com e sem perdas, bem como profundidade de cor e resolução variáveis. Isso o torna ideal para imagens de alta resolução e para preservar todos os detalhes de uma imagem sem comprometer a qualidade. No entanto, os arquivos TIFF tendem a ser significativamente maiores em comparação com outros formatos.

2.5.5 Formato BMP (Bitmap)

O formato BMP é um dos formatos mais simples e diretos. Ele não utiliza compressão, o que resulta em arquivos grandes e não otimizados para armazenamento online ou compartilhamento. Devido à sua simplicidade, o BMP é frequentemente usado em aplicativos que exigem manipulação direta de bits de imagem.

A escolha do formato de imagem depende das necessidades específicas. O JPEG é ótimo para fotografias online, enquanto o PNG é preferível para imagens nítidas com transparência. O GIF é ideal para animações simples, e o TIFF é a escolha para imagens de alta qualidade.

3 PASSOS ESSENCIAIS PARA O PROCESSAMENTO DA IMAGEM DIGITAL

No âmbito do processamento de imagem, existem cinco etapas essenciais que devem ser consideradas: aquisição, armazenamento, processamento, comunicação e exibição de imagens. Vamos ver cada uma dessas etapas em seguida.

3.1 AQUISIÇÃO

A aquisição de imagens digitais começa com a captura da informação visual do mundo real e sua conversão em formato digital. Dois elementos principais são essenciais para essa etapa. O primeiro é um dispositivo sensível a uma faixa específica do espectro eletromagnético, como raios X ou luz visível, que gera um sinal elétrico proporcional à energia captada. O segundo é o digitalizador, que transforma esse sinal em formato digital, permitindo o processamento e armazenamento eletrônico das imagens.

Diferentes tipos de dispositivos são empregados para a detecção de luz visível e infravermelha. Os microdensitrômetros utilizam feixes de luz para registrar níveis de cinza em imagens de transparências ou fotografias. Câmeras Vidicon operam com base na fotocondutividade, criando padrões de condutividade variável em resposta à imagem ótica. Já as matrizes de estado sólido (CCDs) se destacam por registrar imagens em alta velocidade, sendo usadas em áreas como medicina, astronomia e monitoramento ambiental. Enquanto isso, tubos de raios catódicos (câmeras Vidicon) foram usados em dispositivos mais antigos, e os microdensitômetros produzem imagens precisas, adequadas para campos como radiologia e arqueologia.

3.2 ARMAZENAMENTO

No cenário atual, o armazenamento para processamento de imagens avançou consideravelmente, adotando tecnologias modernas além das abordagens tradicionais. A demanda por capacidade de armazenamento para imagens digitais é suprida por soluções inovadoras. Destacamos algumas tecnologias atuais de armazenamento e suas aplicações.

O armazenamento em nuvem revolucionou o cenário, permitindo acesso e armazenamento remoto de dados. No processamento de imagens, isso proporciona flexibilidade e escalabilidade, com serviços como Dropbox e Google Drive.

As unidades de estado sólido (SSDs) substituíram os discos rígidos tradicionais, sendo ideais para armazenar projetos e software de processamento, graças à leitura e gravação rápidas.

Dispositivos externos como discos rígidos e unidades flash USB continuam populares para armazenamento temporário e transferência de dados.

Armazenamento em disco ainda é utilizado em servidores, especialmente com discos de alta capacidade e unidades híbridas.

Mídias ópticas e magnéticas, como CDs e DVDs, têm lugar em algumas aplicações, embora com capacidades limitadas.

A evolução tecnológica ofereceu diversas opções para o armazenamento no processamento de imagens.

3.3 PROCESSAMENTO

O processamento de imagens digitais implica a aplicação de técnicas específicas para aprimorar, modificar ou extrair informações de uma imagem. Similar a ajustar uma fotografia para aumentar sua nitidez ou destaque, essas tarefas podem ser realizadas por meio de programas computacionais. Por exemplo, quando uma imagem apresenta granulação devido a iluminação fraca, várias fotos do mesmo objeto podem ser combinadas, calculando a média dos valores de cor dos pixels para suavizar a granulação.

Em situações que demandam processamento veloz, como a melhoria de imagens capturadas sob iluminação fraca, é possível combinar várias imagens rapidamente, assemelhando-se a um vídeo. Entretanto, essa abordagem requer significativa potência de processamento, o que motiva o desenvolvimento de componentes dedicados para essa finalidade. A evolução tecnológica permitiu reduzir o tamanho e o custo das máquinas de processamento de imagens, tornando-as mais acessíveis. Hoje, muitos computadores incluem componentes especializados, como digitalizadores, unidades lógicas e aritméticas (ULAs) e *frame buffers*, para otimizar o processamento de imagens e melhorar a qualidade das imagens de diversas maneiras. Embora a tecnologia moderna facilite o processo, abordagens criativas e pesquisas ainda são necessárias para resolver problemas específicos.

3.4 COMUNICAÇÃO

A comunicação no processamento digital de imagens está principalmente relacionada à troca de informações entre sistemas de processamento de imagem e à comunicação remota ponto a ponto durante a transmissão de dados de imagem.

No contexto atual de processamento de imagens, a comunicação desempenha um papel fundamental para a troca de dados entre sistemas e a transmissão de informações visuais. As redes de alta velocidade, como 5G e fibra óptica, possibilitam transferências rápidas, enquanto tecnologias sem fio, como Wi-Fi e NFC, permitem conectividade instantânea entre aparelhos. Outras tecnologias como o armazenamento em nuvem, a transmissão de vídeo em tempo real e tecnologias de realidade virtual também remodelaram a maneira como as imagens são comunicadas. Em resumo, a comunicação moderna impulsiona a eficiência e a colaboração no processamento de imagens, conectando sistemas de maneira ágil e segura.

3.5 EXIBIÇÃO

A fase de exibição é crucial em sistemas de processamento de imagens digitais, uma vez que é nesse ponto que as imagens processadas são apresentadas aos usuários para análise e interpretação. Nessa etapa, vários dispositivos de exibição são empregados, cada um com suas próprias características e capacidades.

Os monitores de televisão, tanto em preto e branco quanto coloridos, têm uma presença proeminente nesse cenário. Eles funcionam como interfaces visuais entre os sistemas de processamento de imagens e os usuários finais. Esses monitores são alimentados por saídas de módulos de hardware conectados aos computadores principais. Isso permite que as imagens sejam mostradas com precisão e em alta resolução, proporcionando uma representação visual vívida das imagens processadas.

Além dos monitores, existem outras formas de exibição que também desempenham papéis importantes. Um exemplo são os tubos de raios catódicos de acesso aleatório (CRTs). Nesses sistemas, o feixe de elétrons é controlado por um computador para posicionar o feixe em coordenadas verticais e horizontais, resultando na formação da imagem de saída. A intensidade do feixe é ajustada de acordo com os valores numéricos correspondentes na matriz de imagem, o que gera um padrão de luminosidade variável. Esse padrão é então registrado por uma câmera.

4 MÉTODOS DE PROCESSAMENTO DE IMAGENS

Existem diferentes abordagens para o tratamento de imagens, neste projeto vamos explorar a abordagem do domínio espacial, que consiste na modificação de imagens através da manipulação direta dos valores de seus pixels em uma vizinhança. Todas as imagens aqui tratadas foram processadas usando o Python com códigos elaborados em conjunto com o estudante do departamento de computação Gabriel Rodrigues Malaquias.

Neste capítulo será apresentado alguns preceitos essenciais para estudarmos os métodos de processamento de imagens e, logo em seguida, definiremos as operações e as suas respectivas consequências nas imagens.

Definição 4.1. Um espaço vetorial sobre um corpo \mathbb{K} é um conjunto V que satisfaz as seguintes propriedades:

Adição de vetores. Para todos os vetores $u, v \in V$, $u + v \in V$ possuindo as seguintes propriedades:

- (1) $u + v = v + u$, $\forall u, v \in V$ (comutatividade).
- (2) $(u + v) + w = u + (v + w)$, $\forall u, v, w \in V$ (associatividade).
- (3) Existe 0 em V , chamado vetor nulo, tal que $0 + v = v$, $\forall v \in V$.
- (4) Para todo vetor v em V , existe um vetor $-v \in V$ tal que $v + (-v) = 0$.

Multiplicação por Escalar. Para cada $\alpha \in \mathbb{K}$ e $v \in V$, $\alpha \cdot v \in V$, denominado multiplicação por escalar, com as seguintes propriedades:

- (1) $(\alpha\beta) \cdot v = \alpha(\beta \cdot v)$, $\forall \alpha, \beta \in \mathbb{K}$ e $v \in V$ (associatividade).
- (2) Existe $1 \in \mathbb{K}$ tal que $1 \cdot v = v$, para todo $v \in V$ (dizemos que 1 é o elemento identidade de \mathbb{K}).
- (3) $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$, para todo $\alpha \in \mathbb{K}$ e para todos $u, v \in V$.
- (4) $(\alpha + \beta) \cdot v = \alpha \cdot v + \beta \cdot v$ $\forall \alpha, \beta \in \mathbb{K}$ e $v \in V$.

Definição 4.2. Um conjunto de vetores v_1, v_2, \dots, v_n em um espaço vetorial é dito ser linearmente dependente se existem escalares c_1, c_2, \dots, c_n , não todos nulos, tais que a combinação linear

$$c_1 v_1 + c_2 v_2 + \dots + c_n v_n = 0.$$

Caso contrário, se a única combinação linear que iguala o vetor nulo é a combinação trivial, ou seja, $c_1 = c_2 = \dots = c_n = 0$, então o conjunto de vetores é dito ser linearmente independente.

Definição 4.3. Seja V um espaço vetorial sobre um corpo \mathbb{K} . Um conjunto $B = \{v_1, v_2, \dots, v_n\}$ de vetores em V é uma base de V se:

- a) B é linearmente independente, ou seja, a única combinação linear de vetores em B que iguala o vetor nulo é a combinação trivial, ou seja, $c_1 v_1 + c_2 v_2 + \dots + c_n v_n = 0$ implica que $c_1 = c_2 = \dots = c_n = 0$.
- b) B gera V , ou seja, para cada vetor v em V , existe uma combinação linear de vetores em B que iguala v , ou seja, para todo v em V , existem escalares c_1, c_2, \dots, c_n em \mathbb{K} tais que $v = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$.

Definição 4.4. Chamamos de dimensão de um espaço vetorial V o número de elementos de uma base para V .

Definição 4.5. Sejam V e W espaços vetoriais sobre um mesmo corpo \mathbb{K} . Uma função $T : V \rightarrow W$ é chamada de transformação linear se satisfaz a seguinte propriedade:

$$T(\alpha u + v) = \alpha T(u) + T(v), \text{ onde } u, v \in V \text{ e } \alpha \in \mathbb{K}.$$

Teorema 4.1. A multiplicação por matriz $m \times n$ é uma transformação linear de \mathbb{R}^n em \mathbb{R}^m .

Demonstração. Consultar (PELLEGRINI, 2015), página 111. □

Exemplo 4.1. A transformação com a matriz

$$T = \begin{bmatrix} 1 & 2 & 4 \\ 0 & 3 & 2 \end{bmatrix}$$

é dada por:

$$T(x, y, z) = (x + 2y + 4z, 3y + 2z).$$

Agora, compreendemos que a multiplicação por uma matriz representa uma transformação linear. No entanto, o que acontece quando multiplicamos duas matrizes? Logo adiante exploraremos essa situação.

Teorema 4.2. Sejam $T : U \rightarrow V$ e $S : V \rightarrow W$ transformações lineares entre espaços vetoriais, todos sobre um mesmo corpo \mathbb{K} . A composta $S \circ T : U \rightarrow W$ é uma transformação linear.

Demonstração. Sejam $u, v \in U$ e $\alpha, \beta \in \mathbb{K}$. Desse modo,

$$\begin{aligned}
(S \circ T)(\alpha u + \beta v) &= S(T(\alpha u + \beta v)) \\
&= S(T(\alpha u) + T(\beta v)) \\
&= S(\alpha T(u)) + S(\beta T(v)) \\
&= \alpha S(T(u)) + \beta S(T(v)) \\
&= \alpha(S \circ T)(u) + \beta(S \circ T)(v).
\end{aligned}$$

□

Teorema 4.3. *Seja V e U espaços vetoriais n e m -dimensional, respectivamente, ambos sobre o corpo \mathbb{K} . Seja R uma base ordenada de V e R' uma base ordenada de U . Para cada transformação linear T de V em U , existe uma $m \times n$ matriz A sobre o corpo \mathbb{K} , a matriz de T em relação a R, R' , tal que*

$$[T(u)]_{R'} = A[u]_R,$$

para qualquer $u \in V$.

Demonstração. Consultar (HOFFMAN, 1971), página 87. □

Teorema 4.4. *Sejam $T : V \rightarrow W$ e $S : W \rightarrow Z$ transformações lineares onde todos os espaços vetoriais estão sobre o mesmo corpo \mathbb{K} . Se R, R' e R'' são bases ordenadas dos espaços V, W e Z , respectivamente, se A é matriz de T em relação ao par R, R' e B é a matriz de S em relação ao par R', R'' , então a matriz da composta $S \circ T$ em relação ao par R, R'' é a matriz produto $C = BA$.*

Demonstração. Consultar (HOFFMAN, 1971), página 90. □

A seguir, examinaremos o efeito de cada operação matricial nas imagens, que, por sua vez, também são representadas como matrizes.

4.1 SOMA DE MATRIZES

Definição 4.6. Dadas duas matrizes A e B de mesma ordem $m \times n$, a soma de A e B , denotada por $A + B$, é uma matriz $m \times n$ em que cada elemento é obtido pela adição dos elementos correspondentes de A e B :

$$(A + B) = a_{ij} + b_{ij}$$

Onde a_{ij} é o elemento da matriz A na i -ésima linha e j -ésima coluna, e b_{ij} é o elemento da matriz B na i -ésima linha e j -ésima coluna.

Quando aplicado em imagens, obtemos uma sobreposição das mesmas, ilustrado nas figuras abaixo:

Figura 4.1 – Capa do álbum Arise e Paranoid.



Fonte: Site oficial¹

Figura 4.2 – Imagens sobrepostas.



Fonte: Autoria própria

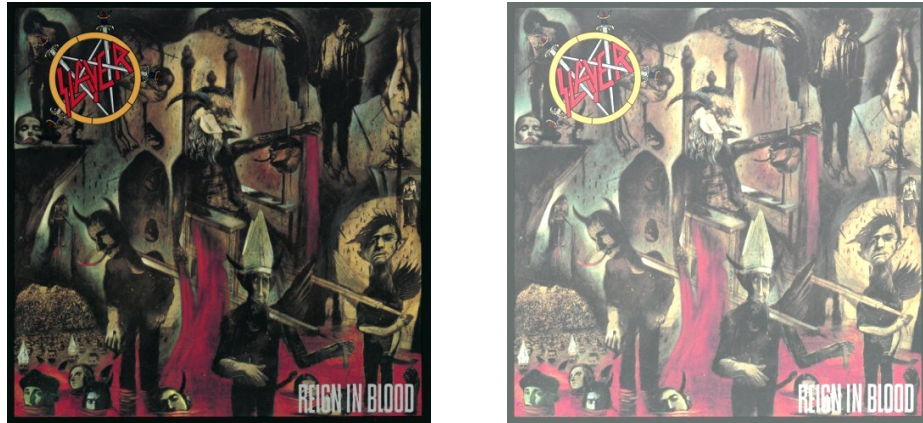
É observável que a imagem resultante experimenta uma alteração em sua intensidade de cor. Esse fenômeno ocorre devido à soma dos pixels, visto que é bem sabido que à medida que nos aproximamos do valor 255, a imagem tende a tornar-se mais clara.

Se acrescentarmos uma constante a todos os elementos da matriz, isso resultará em um aumento geral no brilho da imagem. Vamos examinar a seguir como essa imagem se configuraria ao adicionar 100 a todas as três matrizes que a compõem.

¹ Disponível em: <<https://www.sepultura.com.br/pt/albums>>. Acesso em: 17 ago. 2023.

¹ Disponível em: <<https://www.blacksabbath.com/discography.html>>. Acesso em: 17 ago. 2023.

Figura 4.3 – Capa do álbum Reign in Blood.



Fonte: Site oficial²

Um outro método de editar uma imagem é adicionar uma matriz constante em somente alguns dos canais. Para ilustrar, vamos fazer

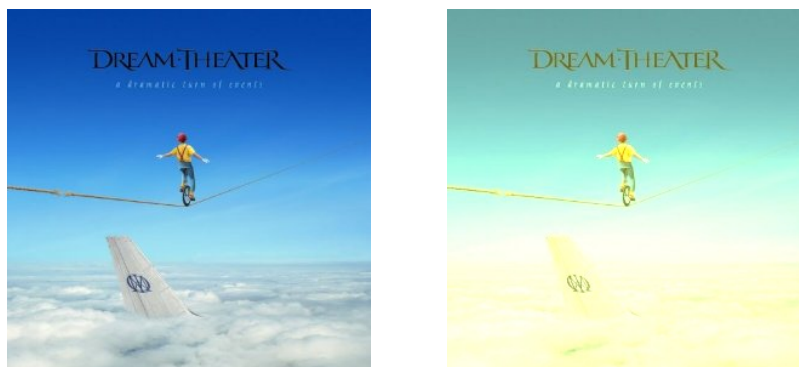
$$100 + R = 100 + r_{ij},$$

para todo r_{ij} de R , sendo R a matriz que representa o canal vermelho da imagem - como descrito na subseção 2.3. De modo análogo, façamos também

$$100 + G = 100 + g_{ij},$$

para todo r_{ij} em G , sendo esta a matriz que representa o canal verde da imagem.

Figura 4.4 – Capa do álbum A Dramatic Turn of Events.



Fonte: Site oficial³

² Disponível em: <<https://www.slayer.net/discography>>. Acesso em: 17 ago. 2023.

² Disponível em; <<https://dreamtheater.net/discography/>>. Acesso em: 17 ago. 2023.

4.2 SUBTRAÇÃO DE MATRIZES

A seguir vamos ver o que acontece quando subtraímos uma imagem por outra (ou então por um valor específico)

Definição 4.7. Dadas duas matrizes A e B de mesma ordem $m \times n$, a subtração de A e B , denotada por $A - B$, é uma matriz $m \times n$ em que cada elemento é obtido pela subtração dos elementos correspondentes de A e B :

$$(A - B) = a_{ij} - b_{ij}$$

Onde a_{ij} é o elemento da matriz A na i -ésima linha e j -ésima coluna, e b_{ij} é o elemento da matriz B na i -ésima linha e j -ésima coluna.

Aplicando esse conceito, podemos subtrair o fundo de uma imagem, assim como conseguimos aplicar o efeito negativo, como segue abaixo:

Figura 4.5 – Capa do álbum Holy Diver



Fonte: Site oficial⁴

Para obter a Figura 4.5 fizemos o seguinte cálculo:

$$255 - A = 255 - a_{ij}.$$

4.3 MULTIPLICAÇÃO DE MATRIZES

Nessa seção estudaremos a aplicação de multiplicação de matrizes no tratamento de imagens.

⁴ Disponível em: <<https://ronniejamesdio.com/discography/>>. Acesso em: 17 de ago. 2023.

Definição 4.8. Dadas duas matrizes A de dimensão $m \times n$ e B de dimensão $n \times p$, o produto $C = A \cdot B$ é uma matriz $m \times p$ onde cada elemento c_{ij} da matriz C é calculado pela seguinte expressão:

$$c_{ij} = \sum_{k=1}^n a_{ik}b_{kj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj}$$

Nesta expressão, a_{ik} é o elemento da matriz A na linha i e coluna k , b_{kj} é o elemento da matriz B na linha k e coluna j , e o somatório é realizado para todos os valores de k de 1 a n . O resultado deste cálculo é atribuído ao elemento c_{ij} da matriz C .

Quando multiplicamos uma imagem pela matriz

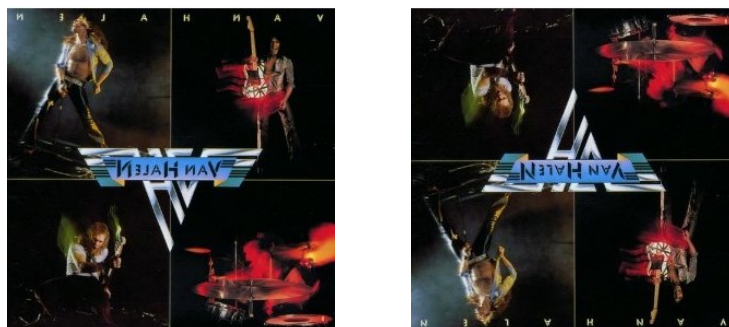
$$I' = \begin{bmatrix} 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & \dots & 0 \end{bmatrix} =$$

pela direita e pela esquerda, obtemos, respectivamente

Figura 4.6 – Capa do álbum Van Halen.



Fonte: Site oficial⁵



⁴ Disponível em: <<https://www.vanhalenstore.com/page/VH/music-video/LP01>>. Acesso em: 12 de nov. 2024.

Podemos também aplicar a transformação linear de rotação às imagens. Para tal, consideremos a matriz

$$\begin{bmatrix} \cos(\theta) & -\text{sen}(\theta) \\ \text{sen}(\theta) & \cos(\theta) \end{bmatrix},$$

onde θ é a medida do ângulo que desejamos rotacionar uma imagem, no sentido anti-horário. A rotação se dá multiplicando à direita toda coordenada (x, y) dos pixels de uma imagem pela matriz de rotação. Vejamos abaixo um exemplo onde usamos $\theta = 45^\circ$ na Figura 4.7 e $\theta = -60^\circ$ na Figura 4.17.

Figura 4.7 – Capa do álbum ...And Justice for All.



Fonte: Site oficial⁶

Figura 4.8 – Capa do álbum Fear of the Dark.



Fonte: Site oficial⁷

⁶ Disponível em: <<https://www.metallica.com/releases/albums/and-justice-for-all-album.html>>. Acesso em: 17 de ago. 2023.

⁶ Disponível em: <<https://www.ironmaiden.com/studio-album/fear-of-the-dark/>>. Acesso em: 17 de ago. 2023.

4.4 CONVOLUÇÃO

Chegamos numa das partes mais interessantes desse nosso projeto: o processamento de imagens através da convolução de uma imagem por uma máscara espacial. Uma máscara espacial é uma matriz bidimensional que é utilizada para realizar operações de filtragem espacial em uma imagem. Essa máscara é uma matriz de coeficientes que é aplicada a cada pixel da imagem para modificar sua intensidade ou cor de acordo com um determinado padrão. A máscara espacial é uma das principais ferramentas utilizadas no processamento de imagens para uma ampla gama de aplicações, como suavização, nitidez, detecção de bordas, entre outras. A máscara espacial é uma matriz que pode ser definida manualmente ou gerada automaticamente por meio de algoritmos de processamento de imagens. Agora que sabemos o que é uma máscara espacial, vamos entender o que é uma convolução.

Definição 4.9. Sejam $h(x, y)$ e $f(x, y)$ matrizes discretas de dimensões $m \times n$ e $p \times q$, respectivamente. A convolução de f por h é dada por:

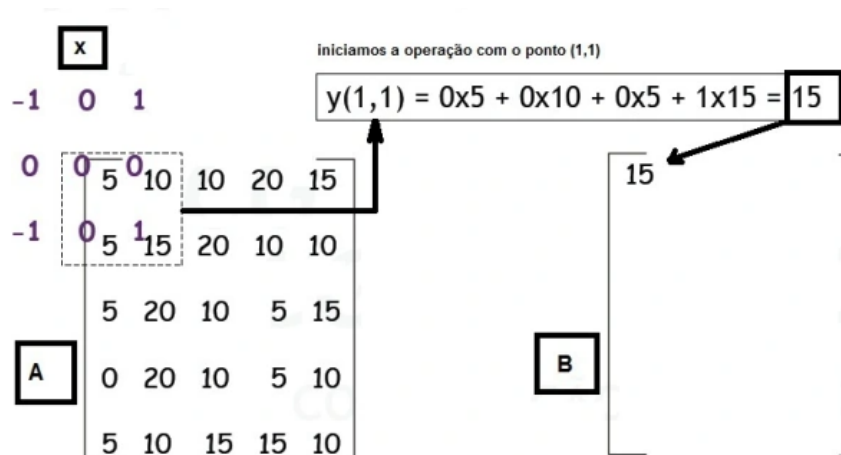
$$g(x, y) = \sum_{i=0}^{N-1} \sum_{k=0}^{N-1} h(x-i, y-k)f(x, y)$$

com $x, y = 0, 1, 2, \dots, N-1$ e o expoente $N-1$ representa o tamanho da janela de convolução.

No tratamento de imagens no domínio espacial, h é o nosso filtro espacial (ou máscara espacial), f a nossa imagem a ser tratada e g a imagem filtrada.

Para ilustrar (e entendermos melhor), segue abaixo como é a operação de convolução:

Figura 4.9 – Cálculo da convolução.



Fonte: Blog do autor Marciano Moura⁸

⁸ Disponível em: <<https://marcianomoura.wordpress.com/2012/02/15/convolucao/>>. Acesso em: 24 ago. 2023.

Nesse exemplo a imagem a ser tratada é a matriz A , a máscara é dada por x e B é a imagem de saída (ou processada).

Continuando o calculo para as demais posições, temos:

$$y(1,2) = 0 \times 5 + 0 \times 10 + 0 \times 10 - 1 \times 5 + 0 \times 15 + 1 \times 20 = 15$$

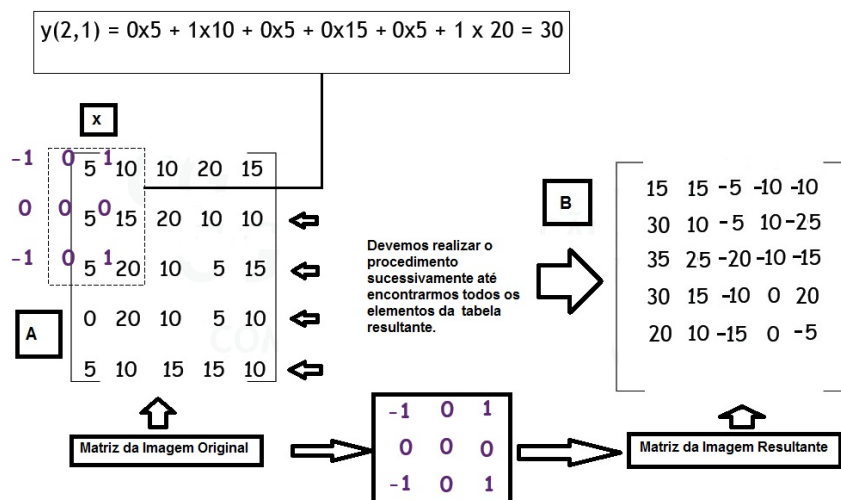
$$y(1,3) = 0 \times 10 + 0 \times 10 + 0 \times 20 - 1 \times 15 + 0 \times 20 + 1 \times 10 = 5$$

$$y(1,4) = 0 \times 10 + 0 \times 20 + 0 \times 15 - 1 \times 20 + 0 \times 10 + 1 \times 10 = -10$$

$$y(1,5) = 0 \times 20 + 0 \times 15 - 1 \times 10 + 0 \times 10 = -10$$

Para a segunda linha o processo é o mesmo, o que diferencia é que, desta vez, a máscara vai estar posicionado uma casa abaixo, e prosseguimos assim sucessivamente, como ilustra a Figura 4.10 abaixo.

Figura 4.10 – Cálculo da convolução 2ª linha.



Fonte: Blog do autor Marciano Moura⁹

Com base na máscara selecionada, somos capazes de gerar efeitos como suavização, realce de bordas, detecção de objetos e muito mais. Essa variabilidade é alcançada apenas por meio da alteração dos coeficientes na máscara espacial. Iniciaremos nossa exploração com máscaras que têm o propósito de suavizar a imagem.

Para notarmos as diferenças e para efeito de comparação vamos usar a Figura 4.11 para aplicar os métodos de nitidez e suavização.

⁹ Disponível em: <<https://marcianomoura.wordpress.com/2012/02/15/convolucao/>>. Acesso em: 24 ago. 2023.

Figura 4.11 – Capa do álbum Pendulum.



Fonte: Site oficial¹⁰

4.4.1 Filtro de Média

Iniciaremos a nossa experimentação com um filtro conhecido como filtro de média. O filtro de média é uma técnica de processamento de imagens que é utilizada para suavizar a imagem, removendo o ruído e as irregularidades. O filtro de média é um filtro que envolve a substituição do valor de cada pixel pela média dos valores dos pixels em sua vizinhança. O filtro de média é simples e eficiente, mas pode resultar em uma perda de detalhes e bordas na imagem. O filtro de média é amplamente utilizado em diversas áreas, como processamento de imagens médicas, visão computacional e reconhecimento de padrões.

Utilizaremos, respectivamente, nas imagens abaixo, os seguintes filtros de média:

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

Figura 4.12 – Filtro de média 3 × 3.



Fonte: Autoria própria

¹⁰ Disponível em: <<https://johnfogerty.com/discography/pendulum-1970/>>. Acesso em: 20 ago. 2023.

$$\begin{bmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{bmatrix}$$

Figura 4.13 – Filtro de média 5 × 5.



Fonte: Autoria própria

$$\frac{1}{49} \times \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Figura 4.14 – Filtro de média 7 × 7.



Fonte: Autoria própria

Como podemos ver, o tamanho da vizinhança utilizada no filtro de média afeta o grau de suavização da imagem e a perda de detalhes. Vizinhanças maiores resultam em uma suavização mais forte, mas também em uma perda maior de detalhes.

4.4.2 Filtro de Realce de Bordas

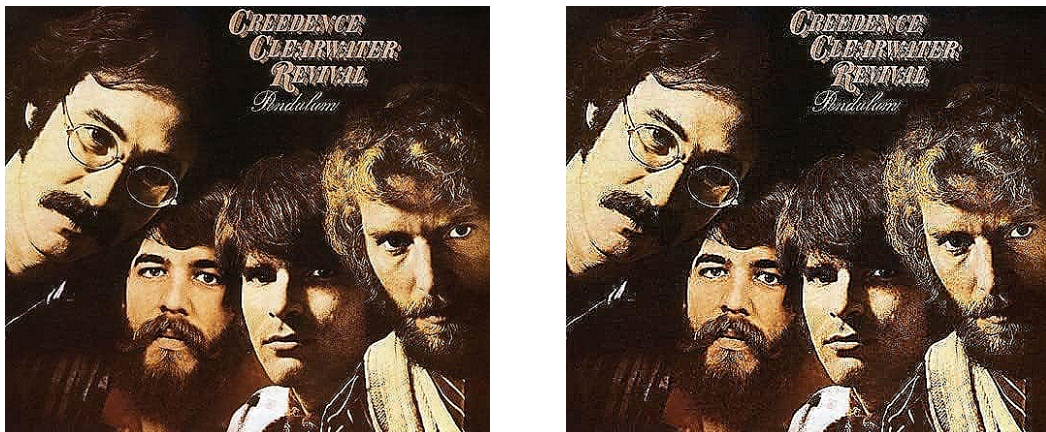
O filtro de realce de bordas é um tipo de filtro que é utilizado para realçar as bordas e detalhes presentes em uma imagem. Ele funciona destacando as transições de intensidade na imagem, aumentando a nitidez de detalhes abruptos e provocando uma redução no contraste da imagem. O filtro de realce de bordas pode ser usado para realçar certas características presentes na imagem, como bordas, linhas curvas ou manchas, no entanto, é importante lembrar que o filtro de realce de bordas também realça o ruído presente na imagem, o que pode prejudicar a interpretação. Diferente dos filtros de suavização, os filtros de realce têm um efeito de aumentar a nitidez de detalhes abruptos das imagens.

Usaremos as máscaras espaciais abaixo na Figura 4.11 e analisaremos o resultado.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & -2 & 0 \\ -2 & 9 & -2 \\ 0 & -2 & 0 \end{bmatrix}$$

A imagem à esquerda é o resultado da primeira máscara espacial e à direita o resultado da segunda.

Figura 4.15 – Filtros de nitidez.



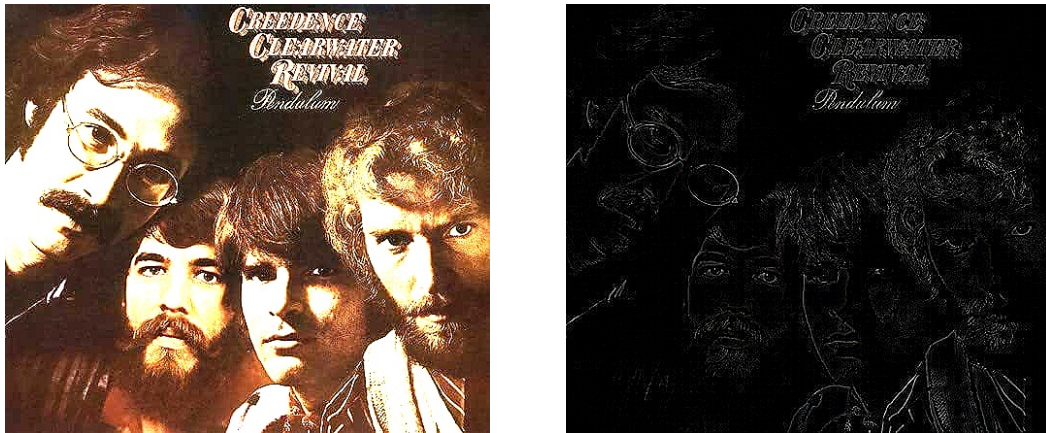
Fonte: Autoria própria.

Nota-se claramente que a segunda máscara espacial torna a imagem mais nítida. Isso se deve ao valor do peso do pixel central a ser substituído ter sido maior que na primeira máscara. É importante destacar que a soma das constantes do filtro espacial de nitidez pode variar a depender do objetivo do tratamento da imagem em questão.

Não podemos afirmar que vale para toda a imagem (não foi encontrado uma fonte que elucide esse ponto), mas foi notado através de experimentação que se a soma das entradas do filtro espacial for igual a 1, então o brilho da imagem não será mudado; se a soma for menor que 1 a imagem será escurecida e, se maior que 1, a imagem ficará mais brilhante. Apenas para curiosidade, segue abaixo o que obtemos ao aumentar para 2 a soma e diminuir para -1 , respectivamente, usando as máscaras abaixo com os respectivos resultados.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Figura 4.16 – Filtros de nitidez de soma 2 e 0, respectivamente.



Fonte: Autoria própria.

4.4.3 Filtro de Detecção de Bordas

Além dos filtros já mencionados, temos também um filtro chamado filtro de Sobel. O operador de Sobel é um filtro que é utilizado para detectar bordas em uma imagem e é aplicado para detectar objetos em imagens, como por exemplo os caracteres de uma placa de um veículo. Ao aplicar o filtro de Sobel em uma imagem colorida, as informações de cor podem interferir na detecção de bordas e transições de intensidade, resultando em um resultado impreciso. Por isso, é importante aplicar o filtro de Sobel em imagens em escala de cinza, onde cada pixel possui apenas um valor de intensidade luminosa. Veremos a seguir um exemplo da aplicação deste filtro.

O método envolve a convolução da imagem que desejamos processar usando as seguintes matrizes:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

A primeira máscara é projetada para identificar bordas na direção horizontal, enquanto a segunda é destinada a detectar bordas na direção vertical. Para ilustrar a aplicação deste filtro, tomemos como exemplo a imagem abaixo:

Figura 4.17 – Freddie Mercury.



Fonte: Página de notícias sobre Curitiba¹¹

Analisando o resultado acima obtido com o resultado da Figura 4.16, pode parecer que um filtro de nitidez serve também para realçar bordas. Vamos investigar aplicando na Figura 4.11 o operador de Sobel.

Figura 4.18 – Filtro de Sobel.



Fonte: Autoria própria.

Podemos notar que há uma grande perda de detalhes e que, portanto, filtros de nitidez não são os mais indicados quando o objetivo é detectar bordas.

Conforme estabelecemos nos resultados iniciais deste capítulo, sabemos que toda matriz possui a capacidade de representar uma transformação linear, e, por sua vez, toda transformação linear é suscetível de ser expressa por uma matriz. Além disso, cientes de que a composição de duas transformações lineares se traduz em um processo de multiplicação das matrizes que representam tais transformações, fica claro que esta operação não encontra equivalência no contexto da convolução. A razão é simples: se fosse o caso, a obtenção do filtro de Sobel

¹¹ Disponível em: <<https://curitibacult.com.br/cevejarria-de-curitiba-promove-festival-com-tributo-a-grandes-nomes-do-rock/>>. Acesso em: 17 de ago. 2023.

não demandaria uma dupla filtragem da imagem, bastaria a multiplicação direta das matrizes para alcançar a composição desejada. No entanto, a ausência destas propriedades no caso da convolução é perceptível, levando em consideração a necessidade de dupla filtragem para gerar o filtro de Sobel. Essa consideração ganha relevância, especialmente na esfera computacional, onde otimizar recursos de memória e acelerar processos é uma meta muito relevante.

Aqui emerge a importância do produto usual, um conceito não imediatamente óbvio, presente no conjunto de matrizes. Trata-se de uma ferramenta de incrível poder, que nos capacita a realizar composições em série com notável facilidade e celeridade. Este produto contrasta nitidamente com as propriedades ausentes na convolução, reforçando sua relevância.

5 DOMÍNIO DA FREQUÊNCIA

No processamento de imagens digitais, trabalhar no domínio da frequência oferece uma perspectiva diferente e muitas vezes mais poderosa do que trabalhar no domínio espacial. No domínio da frequência, em vez de lidar diretamente com os valores de intensidade dos pixels, lidamos com as variações de intensidade representadas por diferentes frequências. Isso nos permite analisar e manipular as características globais de uma imagem, como bordas, texturas e padrões, de maneira mais controlada e eficaz.

Um dos métodos mais utilizados para transformar uma imagem do domínio espacial para o domínio da frequência é a transformada de Fourier. Essa ferramenta matemática decompõe uma imagem em suas componentes básicas de seno e cosseno, representando-a como uma soma de ondas de diferentes frequências. O resultado dessa transformação é uma imagem no domínio de Fourier, onde cada ponto indica a amplitude e a fase de uma frequência específica presente na imagem original.

Trabalhar no domínio da frequência com a transformada de Fourier permite uma série de aplicações avançadas, como a filtragem de ruídos, a compressão de dados, e a reconstrução de imagens. Ao manipular as frequências da imagem, podemos destacar ou atenuar certos detalhes, melhorar a qualidade visual ou reduzir o espaço necessário para armazenar a imagem. O entendimento e o uso da transformada de Fourier são, portanto, essenciais para explorar a fundo as técnicas de tratamento de imagens digitais.

Dessa forma, o capítulo abordará em detalhes a transformada de Fourier, suas propriedades, e suas principais aplicações no tratamento de imagens digitais.

5.1 TRANSFORMADA DE FOURIER

Definição 5.1. Seja $f(x)$ uma função contínua, integrável com $x \in \mathbb{R}$ e $F(u) \in \mathbb{C}$ integrável. A equação $\mathcal{F}\{f(x)\}$ é dita transformada de Fourier de $f(x)$, definida por

$$\mathcal{F}\{f(x)\} = F(u) = \int_{-\infty}^{\infty} f(x) \exp[-i2\pi ux] dx,$$

onde $i = \sqrt{-1}$.

Definição 5.2. Dado $F(u)$ e $f(x)$ como na definição anterior, $f(x)$ pode ser obtida através da transformação inversa de Fourier, dada por

$$\mathcal{F}^{-1}\{F(u)\} = f(x) = \int_{-\infty}^{\infty} F(u) \exp[i2\pi ux] du.$$

Ao longo deste texto, estaremos interessados em funções reais. A transformada de

Fourier de uma função real, todavia, é geralmente complexa, i.e.:

$$F(u) = R(u) + il(u)$$

onde $R(u)$ e $I(u)$ são as partes real e imaginária de $F(u)$, respectivamente.

Como iremos utilizar ao longo do texto, é importante expressar a forma exponencial de $F(u)$:

$$F(u) = |F(u)| \exp [i\phi(u)],$$

onde

$$|F(u)| = [R^2(u) + I^2(u)]^{\frac{1}{2}}$$

e

$$\phi(u) = \tan^{-1} \left[\frac{I(u)}{R(u)} \right].$$

Após a aplicação da transformada de Fourier, a magnitude - que é dada por $|F(u)|$ - dos coeficientes resultantes é calculada, e essa magnitude é conhecida como o espectro de Fourier da imagem. Normalmente, o valor central é o componente de maior magnitude, refletindo a contribuição da média da imagem original. No entanto, como veremos mais adiante, a ampla faixa dinâmica dos coeficientes de Fourier pode fazer com que outros valores apareçam em preto na exibição comum. Para efeitos de visualização, costumamos aplicar uma transformação logarítmica à imagem de magnitude, dessa forma, obtemos uma visualização mais equilibrada, revelando que a imagem contém componentes de todas as frequências - embora a magnitude diminua para frequências mais altas. Isso indica que as frequências baixas carregam mais informações sobre a imagem do que as altas.

Além da magnitude, o ângulo de fase - ou diagrama de fase - dado por $\phi(u)$ é outra parte fundamental da transformada de Fourier. A fase de cada ponto no domínio de Fourier determina o deslocamento temporal ou espacial da frequência correspondente. Embora a imagem de fase por si só não revele tantas informações sobre a estrutura espacial da imagem quanto a magnitude, ela é crucial para a reconstrução precisa da imagem original.

Ao aplicar a transformada de Fourier inversa utilizando apenas a magnitude e ignorando a fase, a imagem resultante pode parecer completamente corrompida, apesar de conter as mesmas frequências da imagem original. Isso demonstra que a informação de fase é vital para a correta reconstrução da imagem no domínio espacial. Portanto, embora frequentemente enfatizemos a magnitude no espectro de Fourier para análise e processamento, a fase desempenha um papel igualmente importante na manutenção da integridade estrutural da imagem.

Calculando o quadrado da magnitude, i.e.:

$$P(u) = |F(u)|^2 = R^2(u) + I^2(u)$$

obtemos o espectro de potência de $f(x)$ (ou densidade espectral). A densidade espectral é uma

função que descreve a distribuição da potência de um sinal ao longo das diferentes frequências. Em outras palavras, ela representa a intensidade do sinal em relação à frequência, permitindo identificar as frequências presentes e entender como elas contribuem para a energia total do sinal.

Podemos estender a transformada de Fourier para duas variáveis $f(x, y)$. Se $f(x, y)$ for integrável e contínua e $F(u, v)$ for integrável, o par de transformadas de Fourier existirá:

$$\mathcal{F}\{f(x, y)\} = F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-i2\pi(ux + vy)] dx dy$$

e

$$\mathcal{F}^{-1}\{F(u, v)\} = f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[i2\pi(ux + vy)] du dv$$

onde u e v são as variáveis de frequência.

Assim como no primeiro caso unidimensional, o espectro de Fourier, o ângulo de fase, e o espectro de potência, são, respectivamente:

$$|F(u, v)| = [R^2(u, v) + I^2(u, v)]^{\frac{1}{2}}$$

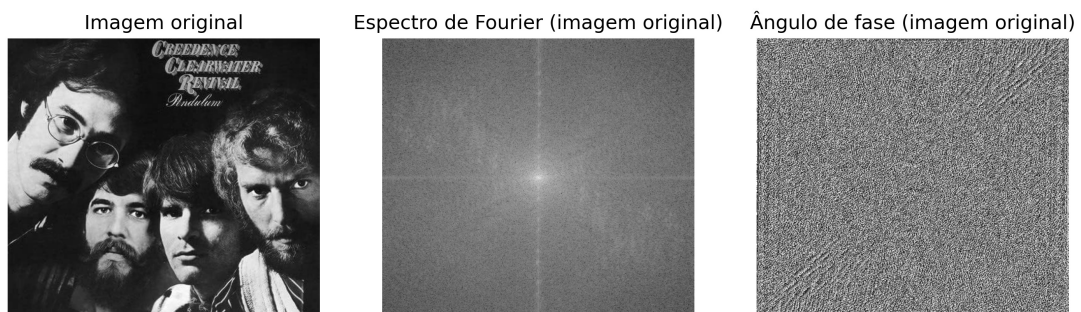
$$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$$

e

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v).$$

Usando a Figura 4.11, vejamos como fica o seu espectro e o seu ângulo de fase.

Figura 5.1 – Espectro e ângulo de fase.



Fonte: Autoria própria.

O apêndice A.1 contém detalhes sobre o código utilizado para a obtenção das imagens.

5.2 A TRANSFORMADA DISCRETA DE FOURIER (TDF)

A TDF é uma transformada que mapeia uma sequência de números reais (representando um sinal discreto no domínio do tempo ou do espaço) para uma outra sequência de números, geralmente complexos. Essa transformação permite analisar o conteúdo espectral de um sinal discreto, ou seja, identificar as diferentes frequências que compõem o sinal e suas respectivas amplitudes e fases. Nesta seção, apresentaremos a definição formal da transformada discreta de Fourier e suas propriedades mais importantes.

Antes de definirmos a transformada discreta de Fourier (TDF), iremos definir a sequência periódica em série discreta de Fourier (SDF), que servirá como base teórica para a TDF que trabalharemos em seguida.

Considere uma sequência periódica $f(x) = \{x_0, x_1, \dots, x_{N-1}\}$, com período N , tal que $f(x) = f(x + kN)$, onde k é um número inteiro. Esta sequência pode ser expressa utilizando a série discreta de Fourier. Diferente da série de Fourier aplicada a funções contínuas e periódicas, uma sequência periódica contém apenas N exponenciais complexas. Isso se deve ao fato de que a função $\exp [i2\pi kx/N]$ também é periódica em k , com o mesmo período N :

Seja $e_k(x) = \exp [i2\pi kx/N]$. Sabemos que $e_0(x) = e_N(x)$ e $e_1(x) = e_{N+1}(x)$. Vamos analisar essas igualdades.

Primeiro, para $k = 0$:

$$e_0(x) = \exp [0] = 1.$$

Agora, para $k = N$:

$$e_N(x) = \exp [i2\pi xN/N] = \exp [i2\pi x] = \cos(2\pi x) + i \operatorname{sen}(2\pi x).$$

Observamos que, para qualquer valor de x , $\cos(2\pi x) = 1$ e $\operatorname{sen}(2\pi x) = 0$, o que nos leva a concluir que $e_0(x) = e_N(x)$.

Para $k = 1$:

$$e_1(x) = \exp [i2\pi x/N].$$

Para $k = N + 1$:

$$e_{N+1}(x) = \exp [i2\pi x(N + 1)/N] = \exp [i2\pi x] \cdot \exp [i2\pi x/N].$$

Como já demonstrado, o termo $\exp [i2\pi x]$ é sempre igual a 1 para qualquer valor de x inteiro. Assim, $e_1(x) = e_{N+1}(x)$.

Desse modo,

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp [i2\pi ux/N], \quad x = 0, 1, 2, \dots, N-1;$$

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp [-i2\pi ux/N];$$

onde $u = 0, 1, 2, \dots, N-1$ é a variável independente de frequência - indica a frequência das componentes senoidais que compõem o sinal no domínio da frequência, e $x = 0, 1, 2, \dots, N-1$ é o número de amostras.

Essas duas últimas equações formam o par de transformadas que representam uma série discreta de Fourier, onde $F(u)$ e $f(x)$ são periódicas com período N .

Para derivar a expressão de $F(u)$, multiplicamos ambos os lados da equação por $\exp [-i2\pi xu/N]$ e somamos sobre x de 0 a $N-1$.

$$\sum_{x=0}^{N-1} f(x) \exp [-i2\pi ux/N] = \sum_{x=0}^{N-1} \sum_{x=0}^{N-1} F(u) \exp [i2\pi ux/N] \exp [-i2\pi ux/N].$$

Ao permutar os somatórios, obtemos:

$$\sum_{x=0}^{N-1} f(x) \exp [-i2\pi ux/N] = \sum_{x=0}^{N-1} F(u) \sum_{x=0}^{N-1} \exp [i2\pi ux/N] \exp [-i2\pi ux/N].$$

Utilizando a propriedade dos somatórios, que afirma que:

$$\sum_{n=0}^{N-1} K = K \cdot N,$$

onde K é uma constante e o somatório $\sum_{u=0}^{N-1} F(u)$ representa a sequência periódica dos coeficientes da série de Fourier: $F(0), F(1), F(2), \dots, F(N-1)$, denotada por $F(u)$, temos que

$$\sum_{x=0}^{N-1} f(x) \exp [-i2\pi ux/N] = F(u) \cdot N \Rightarrow \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp [-i2\pi ux/N] = F(u).$$

A sequência discreta de Fourier nos preparou para entender a TDF, que é a próxima etapa.

Considere uma sequência de duração finita $g(x)$ com comprimento N , onde $g(x) = 0$ fora do intervalo $0 \leq x \leq N-1$. Para simplificar a análise, assumimos que $g(x)$ é um período da sequência periódica $f(x)$ discutida anteriormente. Assim, temos:

$$f(x) = \sum_{r=-\infty}^{\infty} g(x + rN),$$

onde r é um número inteiro. A sequência finita $g(x)$ pode ser obtida da sequência periódica $f(x)$ extraindo um período:

$$g(x) = \begin{cases} f(x), & 0 \leq x \leq N - 1 \\ 0, & \text{fora deste intervalo.} \end{cases}$$

Por conveniência matemática, vamos agora definir uma sequência retangular $R_N(x)$ tal que:

$$R_N(x) = \begin{cases} 1, & 0 \leq x \leq N - 1 \\ 0, & \text{fora deste intervalo.} \end{cases}$$

Portanto, a sequência finita $g(x)$ pode ser expressa como:

$$g(x) = f(x)R_N(x).$$

Os coeficientes $F(u)$ da SDF e a sequência periódica $f(x)$ são periódicos com período N , como já discutido. Denotando os coeficientes de Fourier para a sequência finita $g(x)$ por $G(u)$, temos:

$$G(u) = F(u)R_N(u).$$

Como

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-i2\pi ux/N]$$

e

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[i2\pi ux/N],$$

temos, por fim, o par de transformadas discretas de Fourier:

$$G(u) = \frac{1}{N} \sum_{x=0}^{N-1} g(x) \exp[-i2\pi ux/N], \quad 0 \leq u \leq N - 1, \quad (5.1)$$

$$g(x) = \sum_{u=0}^{N-1} G(u) \exp [i2\pi ux/N], \quad 0 \leq x \leq N-1. \quad (5.2)$$

No caso de duas variáveis, o par de transformadas discretas de Fourier é dado por

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (5.3)$$

para $u = 0, 1, 2, \dots, M-1$ e $v = 0, 1, 2, \dots, N-1$, e

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right] \quad (5.4)$$

para $x = 0, 1, 2, \dots, M-1$ e $y = 0, 1, 2, \dots, N-1$.

A amostragem de uma função contínua, neste contexto, é realizada em uma grade bidimensional, onde as divisões têm larguras Δx e Δy ao longo dos eixos x e y , na devida ordem. Semelhante ao caso unidimensional, a sequência discreta $f(x, y) = \{f(x_0, y_0), f(x_1, y_1), \dots, f(x_{M-1}, y_{N-1})\}$ corresponde às amostras da função $f(x_0 + x\Delta x, y_0 + y\Delta y)$, sendo $x = 0, 1, 2, \dots, M-1$ e $y = 0, 1, 2, \dots, N-1$. A relação entre os incrementos de amostragem nos domínios do espaço e da frequência é dada por:

$$\Delta u = \frac{1}{M\Delta x}$$

$$\Delta v = \frac{1}{N\Delta y}.$$

Essa relação indica que, quanto maior a resolução de uma imagem (ou seja, quanto mais pixels ela possui), menor será o espaçamento entre as amostras no domínio da frequência, e vice-versa. Em outras palavras, uma imagem com alta resolução espacial, contendo muitos pixels, resulta em uma representação no domínio da frequência onde as amostras são mais próximas umas das outras, permitindo capturar mais detalhes de alta frequência. Por outro lado, uma imagem com menor número de pixels (ou baixa resolução espacial) gera uma representação no domínio da frequência com maior espaçamento entre as amostras, limitando a quantidade de informação detalhada que pode ser observada nas frequências mais altas.

A influência dos detalhes de alta frequência ocorre porque, na transformada de Fourier, as frequências mais altas representam variações rápidas na imagem, como bordas e texturas finas. Esses detalhes exigem uma resolução espacial maior (ou seja, mais pixels) para serem capturados com precisão. Quando a resolução espacial é aumentada, o intervalo entre as amos-

tras no domínio da frequência diminui, permitindo uma captura mais precisa dessas variações rápidas, pois há mais “pontos” disponíveis para representar frequências mais altas.

Em contrapartida, as baixas frequências representam variações suaves e gradientes largos, como grandes áreas com cores ou intensidades quase uniformes. Essas variações suaves não requerem uma resolução tão alta para serem representadas com fidelidade, pois os detalhes finos que definem mudanças rápidas estão ausentes. Isso significa que mesmo com uma resolução menor no domínio espacial, as frequências mais baixas continuam bem representadas no domínio da frequência.

Quando as imagens são amostradas em uma matriz quadrada $M = N$, temos que

$$F(u, v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

para $u, v = 0, 1, 2, \dots, N - 1$, e

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u, v) \exp \left[i2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right]$$

para $x, y = 0, 1, 2, \dots, N - 1$.

Considerando $F(u, v)$ e $f(x, y)$ como um par de transformadas de Fourier, a maneira como os fatores multiplicativos constantes são agrupados é uma escolha arbitrária. Na prática, as imagens geralmente são digitalizadas em matrizes quadradas, portanto, nosso foco principal será no par de transformadas de Fourier correspondente a essas equações. A formulação das equações ((5.3)) e (5.2) é ocasionalmente utilizada para lidar com imagens de dimensões não quadradas.

Diferentemente do caso contínuo, a existência da transformada discreta de Fourier não levanta preocupações, pois tanto $F(u)$ quanto $f(u)$ sempre existem. No caso unidimensional, por exemplo, isso pode ser demonstrado substituindo diretamente a equação ((5.2)) na equação ((5.1)):

$$\begin{aligned} F(u) &= \frac{1}{N} \sum_{r=0}^{N-1} \sum_{x=0}^{N-1} F(r) \exp \left[i2\pi \frac{rx}{N} \right] \exp \left[-i2\pi \frac{ux}{N} \right] \\ &= \frac{1}{N} \sum_{r=0}^{N-1} F(r) \left[\sum_{x=0}^{N-1} \exp \left[i2\pi \frac{rx}{N} \right] \exp \left[-i2\pi \frac{ux}{N} \right] \right] \\ &= F(u). \end{aligned} \tag{5.5}$$

A identidade acima decorre da seguinte condição de ortogonalidade:

$$\sum_{x=0}^{N-1} \exp \left[i2\pi \frac{rx}{N} \right] \exp \left[-i2\pi \frac{ux}{N} \right] = \begin{cases} N, & \text{se } r = u. \\ 0, & \text{caso contrário.} \end{cases}$$

De fato, para enxergar o resultado acima, vamos começar com a expressão

$$\frac{1}{N} \sum_{r=0}^{N-1} \sum_{x=0}^{N-1} F(r) \exp \left[i2\pi \frac{rx}{N} \right] \exp \left[-i2\pi \frac{ux}{N} \right].$$

Reorganizando os termos no somatório interno, temos:

$$= \frac{1}{N} \sum_{r=0}^{N-1} F(r) \sum_{x=0}^{N-1} \exp \left[i2\pi \frac{(r-u)x}{N} \right].$$

Caso $r = u$:

Se $r = u$, então a exponencial dentro da soma interna torna-se:

$$\exp \left[i2\pi \frac{(r-u)x}{N} \right] = \exp [0] = 1.$$

Portanto, a soma interna é

$$\sum_{x=0}^{N-1} 1 = N.$$

Substituindo na expressão original, obtemos:

$$\frac{1}{N} F(r) \cdot N = F(r).$$

Caso $r \neq u$:

Se $r \neq u$, temos a soma de uma sequência geométrica:

$$\sum_{x=0}^{N-1} \exp \left[i2\pi \frac{(r-u)x}{N} \right].$$

Essa é uma progressão geométrica com razão $q = \exp \left[i2\pi \frac{(r-u)}{N} \right]$ e N termos. A soma de uma progressão geométrica com N termos e razão $q \neq 1$ é:

$$\sum_{x=0}^{N-1} q^x = \frac{1 - q^N}{1 - q}.$$

Substituindo $q = \exp \left[i2\pi \frac{(r-u)}{N} \right]$, obtemos:

$$\sum_{x=0}^{N-1} \exp \left[i2\pi \frac{(r-u)x}{N} \right] = \frac{1 - \exp [i2\pi(r-u)]}{1 - \exp \left[i2\pi \frac{(r-u)}{N} \right]}.$$

Como $\exp [i2\pi(r-u)] = 1$, o numerador é zero:

$$1 - \exp [i2\pi(r-u)] = 0.$$

Assim,

$$\sum_{x=0}^{N-1} \exp \left[i2\pi \frac{(r-u)x}{N} \right] = 0.$$

Portanto, para $r = u$, a soma interna resulta em N , enquanto para $r \neq u$, ela resulta em 0.

Assim, nossa expressão original se reduz a

$$\frac{1}{N} \sum_{r=0}^{N-1} F(r) \cdot N\delta_{ru} = F(u),$$

onde δ_{ru} é o delta de Kronecker, que vale 1 se $r = u$ e 0 caso contrário.

Concluimos, portanto, que

$$F(u) = \frac{1}{N} \sum_{r=0}^{N-1} \sum_{x=0}^{N-1} F(r) \exp \left[i2\pi \frac{rx}{N} \right] \exp \left[-i2\pi \frac{ux}{N} \right].$$

Nota-se que a mudança da variável de u para r na fórmula ((5.2)) auxilia na compreensão da notação. Da mesma forma, ao inserir a equação ((5.1)) na equação ((5.2)), é possível obter uma igualdade relacionada a $f(x)$, evidenciando a existência do par de transformadas de Fourier dessas fórmulas. Um argumento análogo pode ser utilizado para o par de transformadas discretas de Fourier em duas dimensões.

5.3 PROPRIEDADES DA TRANSFORMADA BIDIMENSIONAL DE FOURIER

Nesta subseção, usando como fontes de referência (GONZALEZ; WOODS, 2000a) e (COOLEY; TUKEY, 1965), discutimos as propriedades essenciais da Transformada de Fourier, que são fundamentais para as análises subsequentes neste trabalho. Embora nosso foco principal seja na transformada bidimensional discreta, é instrutivo introduzir algumas dessas propriedades primeiramente no contexto da transformada contínua unidimensional. Esta abordagem facilita a compreensão dos conceitos básicos, que são depois generalizados para o caso bidimensional.

Uma característica importante a se considerar é a escala dinâmica dos espectros de Fourier. Em muitos casos, a amplitude do espectro é significativamente maior do que o que pode

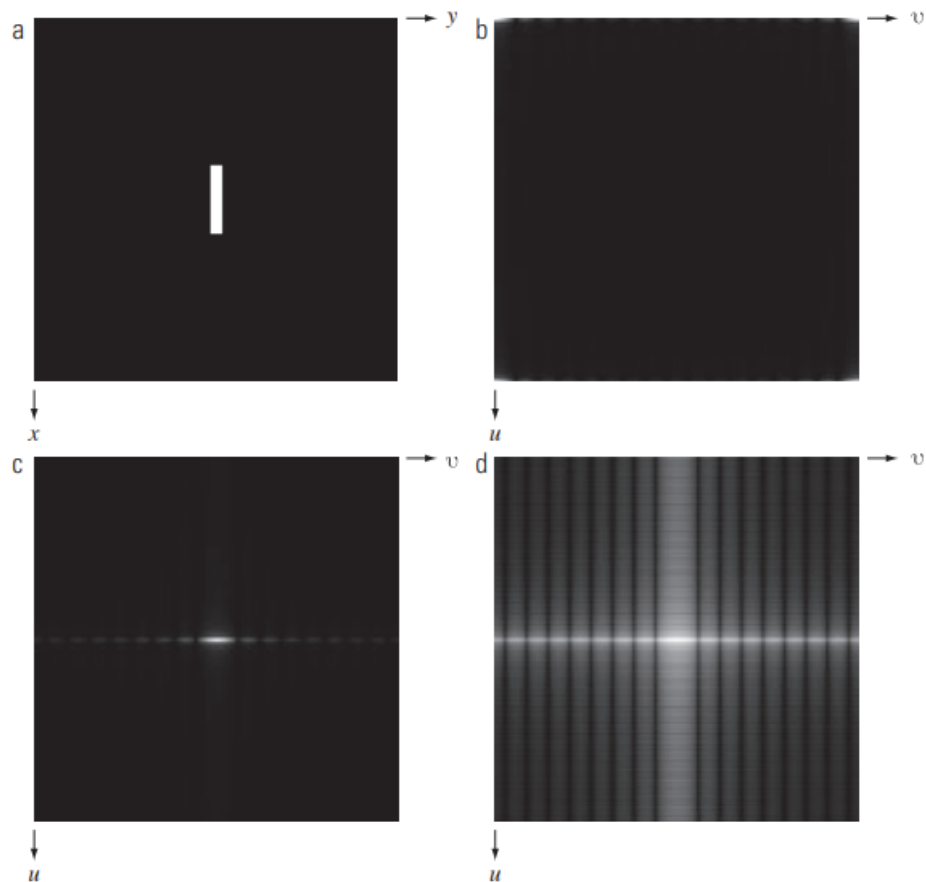
ser fielmente representado por dispositivos de exibição convencionais, isto é, esse espectro é tão grande e brilhante que as telas e máquinas que usamos não conseguem mostrar tudo de forma clara. Conseqüentemente, apenas as regiões mais brilhantes da imagem do espectro são visíveis na tela, o que pode dificultar a interpretação visual.

Para contornar esse problema, uma técnica amplamente utilizada é a aplicação de uma transformação logarítmica na magnitude do espectro. Especificamente, ao invés de exibir diretamente $|F(u, v)|$, utiliza-se a função

$$D(u, v) = c \log(1 + |F(u, v)|)$$

onde c é uma constante de escala, e a função logarítmica faz o espectro ficar menor, comprimido. Essa compressão permite uma melhor visualização das variações de intensidade ao longo do espectro, facilitando significativamente a análise visual.

Figura 5.2 – (a) Imagem original. (b) Espectro mostrando regiões com pontos claros nos quatro cantos. (c) Espectro centralizado. (d) Resultado mostrando detalhes realçados após uma transformação logarítmica.



Fonte: (GONZALEZ; WOODS, 2010).

5.3.1 Separabilidade

A propriedade de separabilidade da Transformada Discreta de Fourier (TDF) permite que ela seja calculada de forma eficiente em duas etapas, aplicando sucessivamente a TDF unidimensional nas linhas e nas colunas da imagem. Isso significa que, em vez de realizar uma operação complexa diretamente sobre toda a matriz bidimensional, podemos calcular a TDF por partes, o que economiza tempo e processamento.

As equações que demonstram essa separabilidade são:

$$F(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \exp \left[-i2\pi \frac{ux}{N} \right] \sum_{y=0}^{N-1} f(x, y) \exp \left[-i2\pi \frac{vy}{N} \right] \quad (5.6)$$

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \exp \left[i2\pi \frac{ux}{N} \right] \sum_{v=0}^{N-1} F(u, v) \exp \left[i2\pi \frac{vy}{N} \right] \quad (5.7)$$

Essas expressões mostram que a TDF bidimensional pode ser calculada em duas partes, aplicando-se a transformada em um eixo por vez (primeiro nas linhas e depois nas colunas, ou vice-versa).

A separabilidade torna-se mais clara na reescrita da Equação ((5.6)) assim:

$$F(u, v) = \sum_{x=0}^{N-1} F(x, v) \exp \left[-i2\pi \frac{ux}{N} \right], \quad (5.8)$$

onde

$$F(x, v) = \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) \exp \left[-i2\pi \frac{vy}{N} \right]. \quad (5.9)$$

Aqui, $F(x, v)$ é obtido aplicando a TDF unidimensional em cada linha da imagem $f(x, y)$. Esse resultado, $F(x, v)$, representa uma primeira “pré-transformada” que captura as frequências ao longo das linhas da imagem.

Depois, aplicamos novamente a TDF, agora nas colunas de $F(x, v)$, para obter o valor final $F(u, v)$, como indicado pela Equação ((5.8)). Esse método possibilita computar a TDF bidimensional de maneira mais eficiente.

Podemos também inverter a ordem das operações, primeiro transformando as colunas de $f(x, y)$ e depois as linhas. Isso mostra que a ordem das operações de linha e coluna é intercambiável, facilitando a implementação da TDF para imagens digitais.

A mesma lógica se aplica à transformada inversa, conforme indicada pela Equação ((5.7)), permitindo, assim, a recuperação da imagem original a partir de sua TDF de forma igualmente eficiente.

5.3.2 Translação

A propriedade da translação diz que, ao multiplicar a representação no domínio espacial por um exponencial, a origem do plano de frequências é deslocada. Da mesma forma, ao multiplicar todos os coeficientes de Fourier por um exponencial, a origem do plano espacial também é deslocada.

5.3.3 Deslocamento e Translação na Transformada de Fourier

Consideremos a correspondência entre as funções abaixo.

$$f(x, y) \exp \left[i2\pi \frac{(ux_0 + vy_0)}{N} \right] \Leftrightarrow F(u - u_0, v - v_0);$$

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v) \exp \left[-i2\pi \frac{(ux_0 + vy_0)}{N} \right].$$

A primeira equação nos mostra que, ao multiplicar a função no domínio espacial $f(x, y)$ por um fator exponencial, ocorre um deslocamento (ou translação) no domínio da frequência, movendo a origem para o ponto (u_0, v_0) . Isso significa que a Transformada de Fourier desloca a origem das frequências. De maneira análoga, ao multiplicar a transformada $F(u, v)$ por um fator exponencial e aplicar a transformada inversa, a origem do plano espacial é deslocada para (x_0, y_0) .

Em particular, neste capítulo, faremos uso frequente da primeira equação, com $u_0 = N/2$ e/ou $v_0 = N/2$, o que nos dá a seguinte expressão:

$$\exp \left[i2\pi \left(\frac{u_0 x}{N} + \frac{v_0 y}{N} \right) \right] = \exp \left[i2\pi \left(\frac{(N/2)x}{N} + \frac{(N/2)y}{N} \right) \right].$$

Agora, vamos simplificar o termo dentro da exponencial:

$$\exp \left[i2\pi \left(\frac{(N/2)x}{N} + \frac{(N/2)y}{N} \right) \right] = \exp \left[i2\pi \left(\frac{x}{2} + \frac{y}{2} \right) \right].$$

Isso pode ser reescrito como:

$$\exp [i\pi(x + y)].$$

Dado que a exponencial complexa tem a seguinte propriedade:

$$\exp[i\pi k] = \cos(\pi k) + i \operatorname{sen}(\pi k),$$

quando k é um número inteiro, temos duas possibilidades:

Se k for par, então:

$$\exp[i\pi k] = 1.$$

Se k for ímpar, então:

$$\exp[i\pi k] = -1.$$

Agora, aplicando isso à nossa expressão, obtemos:

Se $x + y$ for par, então:

$$\exp[i\pi(x + y)] = 1.$$

Se $x + y$ for ímpar, então:

$$\exp[i\pi(x + y)] = -1.$$

Portanto, a exponencial $\exp[i\pi(x + y)]$ alterna entre $+1$ e -1 , dependendo de ser $x + y$ par ou ímpar. Assim, podemos escrever:

$$\exp\left[i2\pi\left(\frac{(N/2)x}{N} + \frac{(N/2)y}{N}\right)\right] = (-1)^{x+y}.$$

Ou seja:

$$f(x, y) \cdot (-1)^{x+y} \Leftrightarrow F\left(u + \frac{N}{2}, v + \frac{N}{2}\right).$$

Isso implica que, ao multiplicar $f(x, y)$ por $(-1)^{x+y}$, a origem da transformada de Fourier de $f(x, y)$ é deslocada para o centro do espaço de frequências $N \times N$.

A partir da segunda equação dessa seção, podemos notar que um deslocamento em $f(x, y)$ não afeta a magnitude da sua transformada de Fourier. Isso ocorre porque:

$$F(u, v) \cdot \exp\left[-i2\pi\left(\frac{ux_0}{N} + \frac{vy_0}{N}\right)\right] = F(u, v),$$

pois a exponencial complexa tem sempre módulo igual a 1.

Essa propriedade é importante, pois ao visualizarmos a transformada de Fourier de uma imagem, frequentemente observamos apenas sua magnitude. Alterações de fase, como as causadas por translação, não afetam a aparência da imagem no domínio da frequência, e, portanto, não afetam o que estamos visualizando.

5.3.4 Simetria Conjugada

A propriedade de simetria conjugada diz que a magnitude da transformada de uma função real está centrada na origem. Especificamente, isso significa que a magnitude da transformada é simetricamente organizada em torno da origem (o ponto de frequência zero). Em termos mais práticos, para um ponto (u, v) no plano de frequência, existe um ponto conjugado oposto $(-u, -v)$ que compartilha a mesma magnitude mas com fases opostas. Vejamos,

$$\begin{aligned} F(-u, -v) &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[i2\pi \left(\frac{-ux}{N} + \frac{-vy}{N} \right) \right] \\ &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-i2\pi \left(\frac{ux}{N} + \frac{vy}{N} \right) \right] \\ &= F(u, v). \end{aligned}$$

5.3.5 Rotação

Esta propriedade da transformada de Fourier descreve como o espectro de Fourier de uma imagem é afetado por uma rotação. Em contraste com a translação, onde o espectro de Fourier não é alterado (apenas a origem do domínio da frequência se move), a rotação da imagem no domínio espacial provoca uma mudança correspondente no espectro de Fourier.

Especificamente, se a imagem de entrada $f(r, \theta)$ for rotacionada por um ângulo θ_0 , o espectro de Fourier da imagem também sofrerá uma rotação equivalente. A relação entre a imagem rotacionada e o espectro de Fourier resultante pode ser expressa da seguinte forma:

$$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \phi + \theta_0),$$

Onde (r, θ) são as coordenadas polares no domínio espacial, e (ω, ϕ) são as coordenadas polares no domínio da frequência. As relações entre as coordenadas cartesianas e polares nos dois domínios são dadas por:

$$x = r \cos \theta \quad y = r \sin \theta \quad u = \omega \cos \phi \quad v = \omega \sin \phi.$$

Portanto, essa propriedade implica que a rotação no domínio espacial ($f(r, \theta)$) resulta em uma rotação correspondente no espectro de Fourier ($F(\omega, \phi)$).

Ao contrário da translação, que não modifica a forma do espectro (apenas desloca sua posição no plano da frequência), a rotação afeta diretamente a estrutura e a orientação do espectro de Fourier. Isso é importante para tarefas como a detecção de objetos rotacionados, onde o espectro de Fourier pode ser analisado para identificar a presença de uma imagem independente de sua rotação.

5.3.6 Linearidade

Consideremos X e Y duas matrizes que representam imagens de entrada para a transformada de Fourier \mathcal{F} . Aplicar a TDF sobre $X + \alpha Y$ é a mesma coisa que aplicar $\mathcal{F}\{X\} + \alpha\mathcal{F}\{Y\}$.

De fato,

$$\begin{aligned}\mathcal{F}\{X + \alpha Y\} &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (f(x, y) + \alpha g(x, y)) \exp \left[-i2\pi \left(\frac{ux}{N} + \frac{vy}{N} \right) \right] \\ &= \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \exp \left[-i2\pi \left(\frac{ux}{N} + \frac{vy}{N} \right) \right] \\ &\quad + \alpha \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x, y) \exp \left[-i2\pi \left(\frac{ux}{N} + \frac{vy}{N} \right) \right] \\ &= \mathcal{F}\{X\} + \alpha\mathcal{F}\{Y\}.\end{aligned}$$

5.3.7 Convolução

Sendo uma das propriedades mais importantes, a convolução mostra que a transformada de Fourier aplicada após a convolução entre duas funções no domínio espacial (uma imagem e uma máscara espacial), é igual à multiplicação da transformada de Fourier dessas duas funções.

Seja a convolução de f e h , denotada por $f * h$, definida como:

$$f * h = \sum_{n=0}^{N-1} f(n) \cdot h(p - n), \quad p = 0, 1, \dots, N - 1.$$

Queremos mostrar que

$$\mathcal{F}\{f * h\} = \mathcal{F}\{f\} \cdot \mathcal{F}\{h\}.$$

Para isso, aplicamos a definição da TDF a $f * h$:

$$\begin{aligned}\mathcal{F}\{f * h\} &= \sum_{y=0}^{N-1} \left(\sum_{x=0}^{N-1} f(x) \cdot h(y - x) \right) \exp \left[-j2\pi \frac{uy}{N} \right] \\ &= \sum_{x=0}^{N-1} f(x) \sum_{y=0}^{N-1} h(y - x) \exp \left[-i2\pi \frac{uy}{N} \right].\end{aligned}$$

Para simplificar ainda mais, definimos uma nova variável $t = y - x$, o que implica $y = t + x$.

Substituindo, obtemos:

$$\sum_{x=0}^{N-1} f(x) \sum_{t=-x}^{N-1-x} h(t) \exp \left[-i2\pi \frac{u(t+x)}{N} \right].$$

Separando os termos exponenciais que dependem de t e x :

$$\sum_{x=0}^{N-1} f(x) \exp \left[-i2\pi \frac{ux}{N} \right] \sum_{t=-x}^{N-1-x} h(t) \exp \left[-i2\pi \frac{ut}{N} \right].$$

Lembremos agora que, como observado no início do capítulo, a TDF possui a propriedade de ser periódica, logo:

$$\sum_{t=-x}^{N-1-x} h(t) \exp \left[-i2\pi \frac{ut}{N} \right] = \sum_{t=0}^{N-1} h(t) \exp \left[-i2\pi \frac{ut}{N} \right].$$

Por fim, notemos que cada somatória corresponde à definição da TDF para f e h , respectivamente:

$$\sum_{x=0}^{N-1} f(x) \exp \left[-i2\pi \frac{ux}{N} \right] \sum_{t=0}^{N-1} h(t) \exp \left[-i2\pi \frac{ut}{N} \right] = \mathcal{F}\{f\} \mathcal{F}\{h\}.$$

5.4 TRANSFORMADA RÁPIDA DE FOURIER (FFT)

A transformada rápida de Fourier (FFT, do inglês *Fast Fourier Transform*) é um algoritmo eficiente para calcular a TDF de uma sequência, ou sua inversa, de forma rápida e com baixo custo computacional.

A principal vantagem da FFT sobre o cálculo direto da TDF é sua eficiência computacional. Enquanto a TDF tem um número de multiplicações e adições proporcional a (N^2) , onde N é o número de pontos da sequência (no caso de uma imagem $N \times N$, teríamos N^4 operações), a FFT reduz essa complexidade para $N \log_2 N$ operações, o que torna a transformação muito mais rápida, especialmente para grandes sequências, como podemos ver na Tabela 5.1.

O algoritmo FFT, desenvolvido por Cooley e Tukey em 1965, explora a simetria e a periodicidade dos cálculos da DFT para dividir o problema em subproblemas menores, que são mais fáceis de resolver. Ele pode ser implementado de diversas maneiras, sendo o mais comum o algoritmo de "radix-2", que requer que o número de pontos N seja uma potência de 2.

Dessa forma, para realizar os tratamentos digitais da próxima seção, será usada a função `np.fft.fft2()`, que é a função que calcula a Transformada de Fourier 2D em Python.

5.5 TRATAMENTOS DE IMAGENS USANDO FFT

Essa seção será dedicada a explorar alguns tratamentos de imagens através da função `np.fft.fft2()` do Python, como destacado a pouco. As máscaras espaciais usadas são encontradas em (GONZALEZ; WOODS, 2000c). Cabe destacar que a máscara usada no filtro passa-alta foi uma modificação do filtro Laplaciano, onde o mais comum é usar uma das duas máscaras:

Tabela 5.1 – Comparação entre N^2 e $N \log_2 N^2$

N	N^2 (TDF Direto)	$N \log_2 N$ (FFT)	Vantagem Computacional
2	4	2	2,00
4	16	8	2,00
8	64	24	2,67
16	256	64	4,00
32	1.024	160	6,40
64	4.096	384	10,67
128	16.384	896	18,29
256	65.536	2.048	32,00
512	262.144	4.608	56,89
1.024	1.048.576	10.240	102,40
2.048	4.194.304	22.528	186,18
4.096	16.777.216	49.152	341,33
8.192	67.108.864	106.496	630,15

Fonte: (GONZALEZ; WOODS, 2000b)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Essa alteração tem por objetivo realçar detalhes mais finos e bordas mais nítidas - devido aos coeficientes maiores nos vizinhos imediatos. Em contrapartida a máscara usada é mais sensível a ruídos.

Devo destacar a parceria fundamental para a realização deste capítulo, pois sua execução só foi possível graças ao auxílio e à colaboração do estudante do Departamento de Computação, Gabriel Rodrigues Malaquias. Sua contribuição foi imprescindível para o desenvolvimento e aplicação das técnicas discutidas ao longo do trabalho, oferecendo uma valiosa perspectiva prática e teórica.

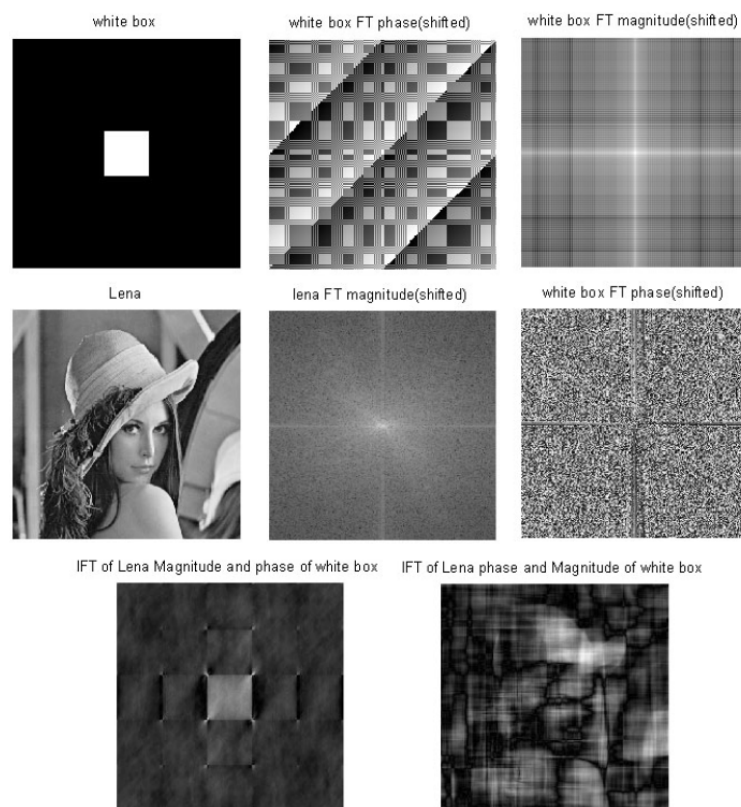
É igualmente importante ressaltar a relevância da propriedade da convolução no contexto do tratamento de imagens digitais. Os códigos desenvolvidos para realizar o processamento das imagens no domínio da frequência dependem diretamente do resultado dessa propriedade. A convolução, enquanto operação fundamental, tem um papel crucial, especialmente quando aplicada em imagens no domínio da frequência, pois é ela que possibilita a manipulação eficaz dos dados para a realização de diversos tipos de filtros e modificações. A correta aplicação dessa técnica foi essencial para alcançar os resultados esperados durante as experimentações.

No que se refere ao ângulo de fase, vale a pena destacar que, embora raramente seja utilizado em práticas de análise de imagens, ele continua a ser expresso a título de curiosidade. Em termos práticos, o ângulo de fase não é amplamente utilizado, mas sua presença é

absolutamente essencial quando se trata de reconstrução de imagens. O ângulo de fase é o componente que determina como as frequências estão dispostas no espaço, ou seja, ele informa a maneira como as diferentes frequências se alinham espacialmente para recriar a imagem de forma correta. Sem ele, a reconstrução fiel da imagem seria impossível, mesmo que o espectro de magnitude seja mantido.

Para ilustrar a importância do ângulo de fase na reconstrução da imagem, podemos observar a Figura 5.3. Nela, é possível ver de maneira clara como a troca do ângulo de fase impacta diretamente na qualidade e na integridade da imagem reconstruída. O efeito da alteração no ângulo de fase demonstra como esse componente é crucial para garantir que a informação espacial e as características da imagem sejam preservadas corretamente durante o processo de reconstrução.

Figura 5.3 – Duas imagens com seus respectivos espectros de magnitude e ângulos de fase, sendo reconstruídas com os ângulos de fase trocados.



Fonte: Site Stack Overlow.¹

¹ Disponível em: <<https://encurtador.com.br/DqxyV>>. Acesso em: 09 nov. 2024.

5.5.1 Filtro Passa-Baixa

O filtro passa-baixa permite atenuar as altas frequências, responsáveis por ruídos e detalhes finos, enquanto preserva as componentes de baixa frequência, que geralmente estão associadas à estrutura geral e às áreas suaves da imagem. Em termos práticos, o filtro passa-baixa é amplamente utilizado para suavizar ou desfocar imagens, removendo detalhes indesejados e proporcionando uma aparência mais uniforme e limpa.

Figura 5.4 – a) Imagem da capa do álbum Pendulum em escala de cinza; b) espectro de Fourier da imagem a); c) ângulo de fase da imagem a); d) espectro de Fourier da máscara espacial; e) produto final do filtro aplicado.



Fonte: Autoria própria

Usando a máscara espacial da Figura 4.14 e a Imagem 4.11, obtemos, através do código expresso no apêndice A.1, o resultado expresso na Figura 5.4.

5.5.2 Filtro Passa-Alta

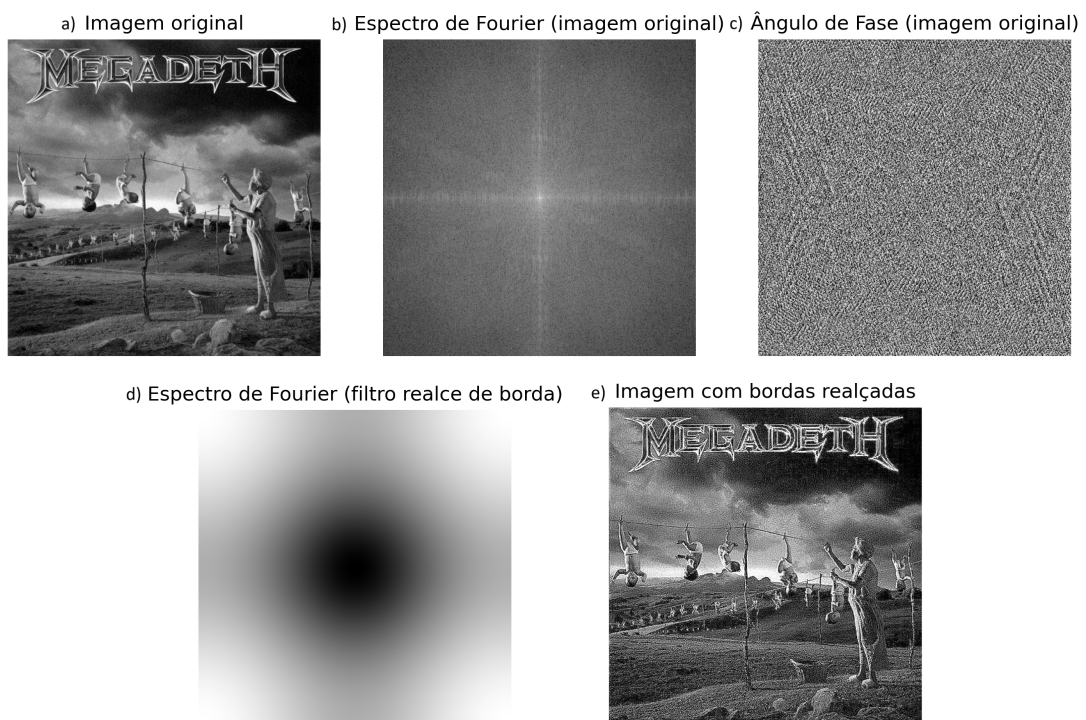
O filtro passa-alta é uma ferramenta poderosa no processamento de imagens digitais, sendo amplamente utilizado para destacar as variações abruptas de intensidade presentes nas imagens. Essas variações estão tipicamente associadas às bordas e aos detalhes finos da imagem, que correspondem a altas frequências no espectro de Fourier. Ao aplicar esse filtro, os componentes de baixa frequência, que são responsáveis pelas regiões suaves e homogêneas da imagem, são bloqueados ou atenuados. Dessa forma, o filtro permite que apenas as frequências mais altas, que carregam as informações de bordas e detalhes, sejam preservadas, enquanto as

informações mais "suaves" são removidas. Esse processo resulta em uma imagem com maior contraste e realce das áreas estruturais, proporcionando uma nitidez acentuada, especialmente em detalhes que poderiam passar despercebidos em uma visualização comum.

O filtro passa-alta aumenta a visibilidade dos detalhes e reduz o efeito de suavização causado por componentes de baixa frequência.

Como exemplo prático, a máscara espacial de nitidez ?? foi aplicada à imagem usando o código do apêndice A .2, resultando em uma imagem mais nítida, como mostrado na figura 5.5. Isso ilustra como o filtro passa-alta pode realçar detalhes finos e contornos, melhorando a clareza e a definição visual da imagem.

Figura 5.5 – a) Imagem transformada em escala de cinza; b) espectro de magnitude da imagem; c) ângulo de fase da imagem; d) espectro de magnitude da máscara espacial; e) resultado da imagem tratada.



Fonte: a) Site oficial da banda.²b), c), d), e) Autoria própria.

Máscara espacial utilizada para o realçamento de bordas:

$$\begin{bmatrix} 0 & -2 & 0 \\ -2 & 9 & -2 \\ 0 & -2 & 0 \end{bmatrix}$$

² Disponível em: <<https://www.megadeth.com/collections/discography/products/youthanasia>>. Acesso em: 09 nov. 2024.

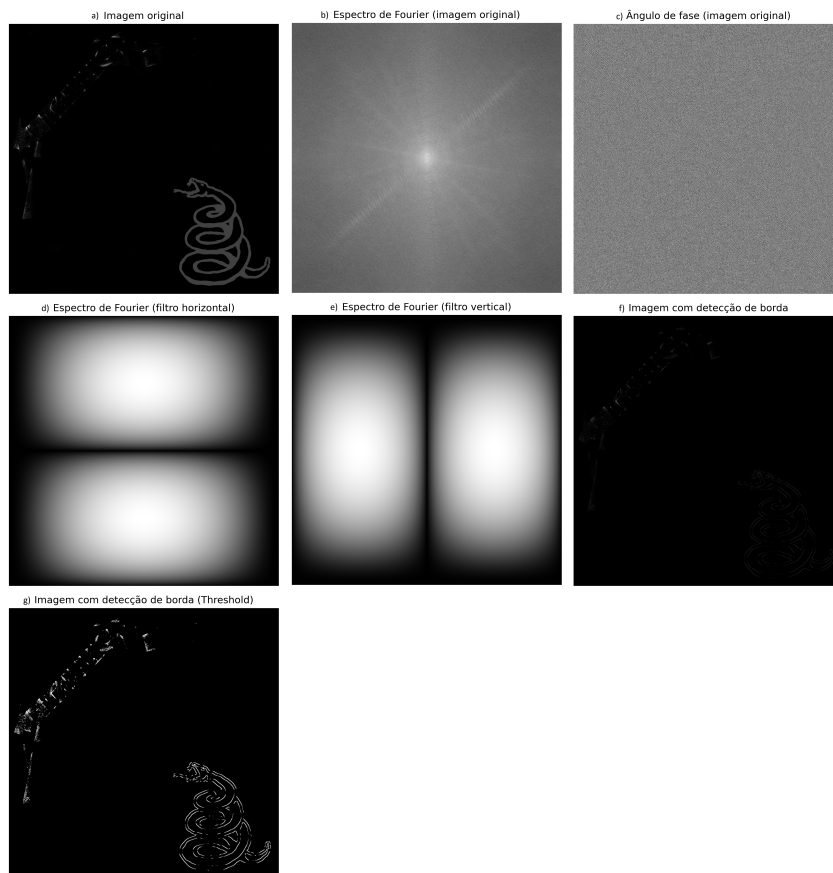
5.5.3 Filtro Detector de Bordas (Filtro de Sobel)

No processamento de imagens digitais, o filtro detector de bordas é utilizado para identificar e realçar as áreas de transição abrupta de intensidade, onde há uma mudança significativa entre pixels adjacentes, caracterizando as bordas de objetos na imagem. Entre os filtros de detecção de bordas, o filtro de Sobel se destaca por sua eficiência ao aplicar um gradiente de intensidade nas direções horizontal e vertical, proporcionando uma detecção de bordas mais precisa e suave, o qual será o foco da nossa abordagem.

Os filtros aqui utilizados são os mesmos da seção 4.3.3. Para relembrar:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Figura 5.6 – a) Imagem original; b) espectro de magnitude da imagem; c) ângulo de fase da imagem; d) espectro de magnitude da máscara espacial horizontal; e) espectro de magnitude da máscara espacial vertical; f) resultado da imagem tratada; g) resultado da imagem tratada com thresholded de 125.



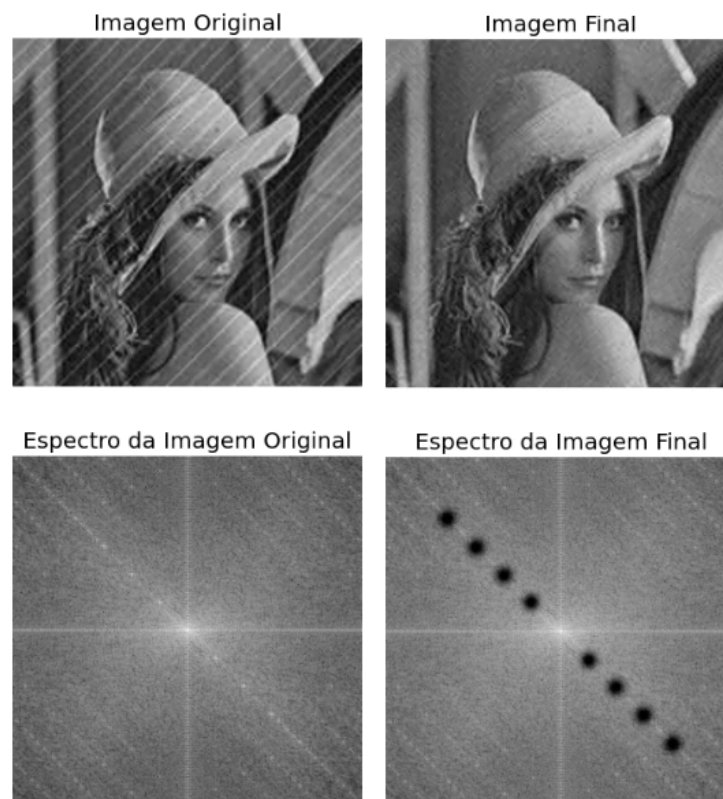
Fonte: a) Site oficial da banda.³b), c), d), e), f), g) Autoria própria.

O código utilizado para a obtenção dessas imagens pode ser consultado no apêndice A .3. Podemos notar que o uso do thresholded ajudou a evidenciar as bordas detectadas pelo filtro (que, até então, estava mais difícil de enxergar). Interessante notar também o comportamento do espectro de Fourier dos filtros horizontais e verticais das matrizes de Sobel, onde o filtro horizontal apresenta um espectro de duas faixas de mesmo sentido, ocorrendo o mesmo com o vertical.

5.5.4 Outros Exemplos de Tratamentos Bem-Sucedidos

Durante o desenvolvimento do projeto, exploramos a teoria matemática por trás da ferramenta de edição de imagens baseada na transformada de Fourier. Ao longo desse processo, fomos aprofundando nossos conhecimentos e apresentando exemplos práticos de suas aplicações. No entanto, devido a limitações técnicas no uso da ferramenta, pode-se ter a impressão de que ela não é tão potente quanto realmente é, ao analisar apenas os resultados obtidos até agora. Por essa razão, apresentaremos alguns exemplos de tratamentos de imagem utilizando a transformada de Fourier para demonstrar melhor sua capacidade.

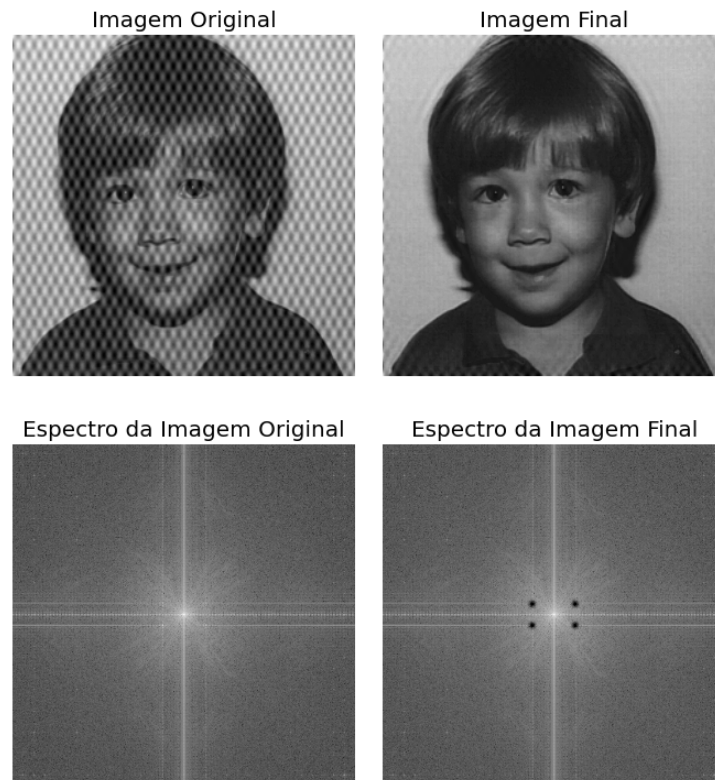
Figura 5.7 – Exemplo de remoção de ruído através do filtro de Notch.



Fonte: Blog do Isaac Vieira de Morais.⁴

³ Disponível em: <<https://www.metallica.com/releases/albums/the-black-album.html>>. Acesso em: 12 nov. 2024.

Figura 5.8 – Mais exemplos de remoção de ruído através do filtro de Notch.



Fonte: Blog do Isaac Vieira de Morais.⁵

O filtro de notch é uma ferramenta utilizada em processamento de sinais e imagens com o objetivo de atenuar ou eliminar uma faixa específica de frequências, como interferências periódicas. Ele é projetado para bloquear uma frequência central, ao mesmo tempo em que permite a passagem das frequências adjacentes, sem afetar significativamente as demais componentes do sinal ou da imagem. O filtro de notch é particularmente eficaz na remoção de ruídos periódicos, como aqueles gerados por fontes de interferência externas.

Atuando no domínio da frequência, para aplicar este filtro basta multiplicarmos a transformada de Fourier da imagem pelo filtro de notch (exatamente como fizemos usando as máscaras espaciais, onde multiplicamos as transformadas espaciais das máscaras espaciais pela transformada de Fourier da imagem).

⁴ Disponível em: <<https://isaacvmorais.blogspot.com/2014/02/filtros-notch.html>>. Acesso em: 12 nov. 2024.

⁵ Disponível em: <<https://isaacvmorais.blogspot.com/2014/02/filtros-notch.html>>. Acesso em: 12 nov. 2024.

6 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo explorar o tratamento de imagens digitais, abordando tanto o domínio espacial quanto o domínio da frequência. A partir de uma análise teórica e prática, investigamos diferentes técnicas de processamento de imagens, que envolvem operações em ambas as abordagens, e como essas operações afetam a qualidade, o conteúdo e as características visuais das imagens.

No domínio espacial, focamos em técnicas como filtragem, realce de bordas e suavização, que são fundamentais para manipulações diretas nos pixels da imagem. Essas operações, muitas vezes simples, desempenham um papel crucial na melhoria da qualidade de imagens e no destaque de características importantes, como contornos e transições de intensidade. A implementação de filtros como o filtro passa-baixa e o filtro de Sobel, por exemplo, demonstraram como a manipulação direta de pixels pode ser eficaz para tarefas de reconhecimento e análise de imagens.

Por outro lado, no domínio da frequência, o uso da Transformada de Fourier, especialmente por meio da Transformada Rápida de Fourier (FFT), proporcionou uma perspectiva diferente sobre o tratamento de imagens. A transformação das imagens para o domínio da frequência permitiu a aplicação de filtros mais avançados, como os filtros passa-alta e passa-baixa, que atuam sobre componentes de frequência da imagem. Essas técnicas são fundamentais para tarefas como a remoção de ruídos e a preservação ou realce de detalhes importantes em diferentes escalas de resolução.

Ao longo deste trabalho, ficou claro que o tratamento de imagens digitais exige uma compreensão abrangente das duas abordagens, espacial e de frequência, e como elas podem ser complementares. Enquanto o domínio espacial oferece uma manipulação intuitiva e direta dos pixels, o domínio da frequência permite uma análise mais profunda e eficaz de componentes ocultos na imagem, como frequências altas e baixas que influenciam a textura e a suavidade das imagens.

Uma contribuição importante deste trabalho foi a implementação prática das técnicas estudadas, utilizando a linguagem de programação Python. Através dessa implementação, foi possível demonstrar como os conceitos teóricos se aplicam diretamente à manipulação de imagens reais, proporcionando uma experiência de aprendizado mais concreta e interativa.

Em resumo, este trabalho contribui para uma melhor compreensão do processamento de imagens digitais, ao integrar os conceitos do domínio espacial e da frequência, e demonstra a importância de se utilizar ambas as abordagens para alcançar resultados mais eficazes e robustos.

REFERÊNCIAS

COOLEY, J. W.; TUKEY, J. W. An algorithm for the machine calculation of complex fourier series. **Mathematics of computation**, JSTOR, v. 19, n. 90, p. 297–301, 1965. Citado na página 43.

FEOFILOFF, P. **Bytes, números e caracteres**. 2019. <<https://www.ime.usp.br/~pf/algoritmos/aulas/bytes.html>>. Acessado em: 24 de ago. 2023. Citado na página 11.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000. Citado na página 43.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000. 85 p. Citado na página 51.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. [S.l.]: Editora Blucher, 2000. 137–143 p. Citado na página 50.

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagem**. **Pearson, ISBN-10: 8576054019**, v. 10, p. 11–27, 2010. Citado na página 44.

HOFFMAN, K. **Linear algebra**. [S.l.: s.n.], 1971. Citado na página 20.

PELLEGRINI, J. C. **Álgebra linear**. **São Paulo: IME–USP**, 2015. Citado na página 19.

APÊNDICE A – CÓDIGOS UTILIZADOS NO TRATAMENTO DE IMAGENS NO DOMÍNIO DA FREQUÊNCIA

A .1 FILTRO PASSA-BAIXA

```

import cv2
import numpy as np
import matplotlib.pyplot as plt
# Load the grayscale image
image = cv2.imread("Creedence.jpg", cv2.IMREAD_GRAYSCALE)
rows, cols = image.shape
# Step 1: Create the 7x7 averaging kernel
kernel_size = 7
average_kernel = np.ones((kernel_size, kernel_size), np.float32) / (kernel_size * kernel_size)
# Step 2: Pad the kernel to the size of the image
padded_kernel = np.zeros_like(image, dtype=np.float32)
padded_kernel[:kernel_size, :kernel_size] = average_kernel
# Step 3: Fourier transform both the image and the kernel
f_image = np.fft.fft2(image)
f_kernel = np.fft.fft2(padded_kernel)
# Step 4: Multiply the Fourier-transformed image by the Fourier-transformed kernel
f_image_filtered = f_image * f_kernel
# Step 5: Perform the inverse Fourier transform to get the blurred image
img_back = np.fft.ifft2(f_image_filtered)
img_back = np.abs(img_back)
# Display the original image, its frequency domain, kernel in frequency domain, and blurred result
plt.figure(figsize=(20, 5))
# Original image plt.subplot(151), plt.imshow(image, cmap='gray')
plt.title('Imagem original'), plt.axis('off')
# Frequency domain of the original image
plt.subplot(152), plt.imshow(np.log(np.abs(np.fft.fftshift(f_image)) + 1), cmap='gray')
plt.title('Espectro de Fourier (imagem original)'), plt.axis('off')
# Phase angle of the original image
plt.subplot(153), plt.imshow(np.angle(f_image), cmap='gray')
plt.title('Ângulo de fase (imagem original)'), plt.axis('off')
# Frequency domain of the kernel
plt.subplot(154), plt.imshow(np.log(1 + np.fft.fftshift(np.abs(f_kernel)))), cmap='gray')

```

```
plt.title('Espectro de Fourier (filtro de desfoque)'), plt.axis('off')
# Blurred image
plt.subplot(155), plt.imshow(img_back, cmap='gray')
plt.title('Imagem desfocada'), plt.axis('off')
plt.savefig("Creedence_desfocado.png", dpi=300, bbox_inches='tight')
```

A .2 FILTRO PASSA-ALTA

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
# Load the grayscale image
image = cv2.imread("mega.jpg", cv2.IMREAD_GRAYSCALE)
rows, cols = image.shape
# Step 1: Create the 7x7 averaging kernel
kernel_size = 3
#average_kernel = np.array([[0, -2, 0], [-2, 9, -2], [0, -2, 0]])
average_kernel = np.array([[0, -1, 0], [-1, 5, -1], [0, -1, 0]])
#average_kernel = np.zeros_like((kernel_size, kernel_size), np.float32) / (kernel_size *
kernel_size)
# Step 2: Pad the kernel to the size of the image
padded_kernel = np.zeros_like(image, dtype=np.float32)
padded_kernel[:kernel_size, :kernel_size] = average_kernel
# Step 3: Fourier transform both the image and the kernel
f_image = np.fft.fft2(image)
f_kernel = np.fft.fft2(padded_kernel)
# Step 4: Multiply the Fourier-transformed image by the Fourier-transformed kernel
f_image_filtered = f_image * f_kernel
# Step 5: Perform the inverse Fourier transform to get the blurred image
img_back = np.fft.ifft2(f_image_filtered)
img_back = np.abs(img_back)
img_back = np.uint8(np.clip(img_back, 0, 255))
# Display the original image, its frequency domain, kernel in frequency domain, and blurred
result
plt.figure(figsize=(20, 5))
# Original image
plt.subplot(151), plt.imshow(image, cmap='gray')
plt.title('Imagem original'), plt.axis('off')
# Frequency domain of the original image
```

```

plt.subplot(152), plt.imshow(np.log(np.abs(np.fft.fftshift(f_image)) + 1), cmap='gray')
plt.title('Espectro de Fourier (imagem original)'), plt.axis('off')
# Phase angle of the original image
plt.subplot(153), plt.imshow(np.angle(f_image), cmap='gray')
plt.title('Ângulo de Fase (imagem original)'), plt.axis('off')
# Frequency domain of the kernel
plt.subplot(154), plt.imshow(np.log(1 + np.fft.fftshift(np.abs(f_kernel))), cmap='gray')
plt.title('Espectro de Fourier (filtro realce de borda)'), plt.axis('off')
# Blurred image
plt.subplot(155), plt.imshow(img_back, cmap='gray')
plt.title('Imagem com bordas realçadas'), plt.axis('off')
plt.savefig("megadeth_realce.png", dpi=300, bbox_inches='tight')

```

A .3 FILTRO DETECTOR DE BORDAS (FILTRO DE SOBEL)

```

import cv2
import numpy as np
import matplotlib.pyplot as plt
# Load the grayscale image
image = cv2.imread("metallica.jpeg", cv2.IMREAD_GRAYSCALE)
rows, cols = image.shape
# Step 1: Create the 7x7 averaging kernel
horizontal_kernel_size = 3
vertical_kernel_size = 3
horizontal_kernel = np.array([[ -1, -2, -1], [ 0, 0, 0], [ 1, 2, 1]])
vertical_kernel = np.array([[ -1, 0, 1], [-2, 0, 2], [-1, 0, 1]])
# Step 2: Pad the kernel to the size of the image
horizontal_padded_kernel = np.zeros_like(image, dtype=np.float32)
horizontal_padded_kernel[:,horizontal_kernel_size, :horizontal_kernel_size] = horizontal_kernel
vertical_padded_kernel = np.zeros_like(image, dtype=np.float32)
vertical_padded_kernel[:,vertical_kernel_size, :vertical_kernel_size] = vertical_kernel
# Step 3: Fourier transform both the image and the kernel
f_image = np.fft.fft2(image)
f_horizontal_kernel = np.fft.fft2(horizontal_padded_kernel)
f_vertical_kernel = np.fft.fft2(vertical_padded_kernel)
# Step 4: Multiply the Fourier-transformed image by the Fourier-transformed kernel
f_image_filtered = f_image * f_horizontal_kernel * f_vertical_kernel
# Step 5: Perform the inverse Fourier transform to get the blurred image

```

```

img_back = np.fft.ifft2(f_image_filtered)
img_back = np.abs(img_back)
# Apply the threshold: if pixel value < 125, turn it to black (0), otherwise to white (255)
_, img_thresholded = cv2.threshold(img_back, 125, 255, cv2.THRESH_BINARY)
# Display the original image, its frequency domain, kernel in frequency domain, blurred
result, and img_back
plt.figure(figsize=(20, 15)) # Adjusted figure size for 3 rows and 3 columns
# Original image
plt.subplot(331), plt.imshow(image, cmap='gray')
plt.title('Imagem original'), plt.axis('off')
# Frequency domain of the original image
plt.subplot(332), plt.imshow(np.log(np.abs(np.fft.fftshift(f_image)) + 1), cmap='gray')
plt.title('Espectro de Fourier (imagem original)'), plt.axis('off')
# Phase angle of the original image
plt.subplot(333), plt.imshow(np.angle(f_image), cmap='gray')
plt.title('Ângulo de fase (imagem original)'), plt.axis('off')
# Frequency domain of the horizontal kernel
plt.subplot(334), plt.imshow(np.log(1 + np.fft.fftshift(np.abs(f_horizontal_kernel))), cmap='gray')
plt.title('Espectro de Fourier (filtro horizontal)'), plt.axis('off')
# Frequency domain of the vertical kernel
plt.subplot(335), plt.imshow(np.log(1 + np.fft.fftshift(np.abs(f_vertical_kernel))), cmap='gray')
plt.title('Espectro de Fourier (filtro vertical)'), plt.axis('off')
# img_back (result after inverse FFT)
plt.subplot(336), plt.imshow(img_back, cmap='gray')
plt.title('Imagem com detecção de borda'), plt.axis('off')
# Blurred image (thresholded)
plt.subplot(337), plt.imshow(img_thresholded, cmap='gray')
plt.title('Imagem com detecção de borda (Threshold)'), plt.axis('off')
# Save the figure as an image
plt.tight_layout()
plt.savefig("metallica_black_deteccao.png", dpi=300, bbox_inches='tight')

```

Exceto quando indicado o contrário, a licença deste item é descrito como
Attribution-NonCommercial-NoDerivs 3.0 Brazil