



Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Matemática



Aplicações da Álgebra Linear em Processamento de Imagens Digitais

Autor: *Juliana Souza Paz*

Orientador: *Prof. Dr. Leandro Nery de Oliveira*

Disciplina: Trabalho de Conclusão de Curso B

Curso: Licenciatura em Matemática

Professores Responsáveis: Prof. Dr. Alex Carlucci Rezende
Profa. Dra. Luciene Nogueira Bertencello
Profa. Dra. Thaís Maria Dalbelo.

São Carlos, 12 de fevereiro de 2024

Aplicações da Álgebra Linear em Processamento de Imagens Digitais

Autor: *Juliana Souza Paz*

Orientador: *Prof. Dr. Leandro Nery de Oliveira*

Disciplina: Trabalho de Conclusão de Curso B

Curso: Licenciatura em Matemática

Professores Responsáveis: Prof. Dr. Alex Carlucci Rezende
Profa. Dra. Luciene Nogueira Bertencello
Profa. Dra. Thaís Maria Dalbelo.

Instituição: Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Matemática

São Carlos, 12 de fevereiro de 2024

Juliana Souza Paz

Prof. Dr. Leandro Nery de Oliveira
Orientador

Paz, Juliana Souza

Aplicações da álgebra linear em processamento de
imagens digitais / Juliana Souza Paz -- 2024.
71f.

TCC (Graduação) - Universidade Federal de São Carlos,
campus São Carlos, São Carlos

Orientador (a): Leandro Nery de Oliveira

Banca Examinadora: Leandro Nery de Oliveira,
Guillermo Antonio Lobos Villagra, Savio Brochini

Rodrigues

Bibliografia

1. Álgebra linear. 2. Processamento digital de imagens. I.
Paz, Juliana Souza. II. Título.

Ficha catalográfica desenvolvida pela Secretaria Geral de Informática
(SIn)

DADOS FORNECIDOS PELO AUTOR

Bibliotecário responsável: Arildo Martins - CRB/8 7180



FUNDAÇÃO UNIVERSIDADE FEDERAL DE SÃO CARLOS
COORDENAÇÃO DOS CURSOS DE GRADUAÇÃO EM MATEMÁTICA - CCM/CCET
Rod. Washington Luís km 235 - SP-310, s/n - Bairro Monjolinho, São Carlos/SP, CEP 13565-905
Telefone: (16) 33518221 - <http://www.ufscar.br>

DP-TCC-FA nº 22/2024/CCM/CCET

Graduação: Defesa Pública de Trabalho de Conclusão de Curso

Folha Aprovação (GDP-TCC-FA)

FOLHA DE APROVAÇÃO

JULIANA SOUZA PAZ

APLICAÇÕES DA ÁLGEBRA LINEAR EM PROCESSAMENTO DE IMAGENS DIGITAIS

Trabalho de Conclusão de Curso

Universidade Federal de São Carlos – Campus São Carlos

São Carlos, 07 de fevereiro de 2024

ASSINATURAS E CIÊNCIAS

Cargo/Função	Nome Completo
Orientador	Leandro Nery de Oliveira
Membro da Banca 1	Guillermo Antonio Lobos Villagra
Membro da Banca 2	Savio Brochini Rodrigues



Documento assinado eletronicamente por **Leandro Nery de Oliveira, Chefe de Departamento**, em 23/08/2024, às 14:39, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Savio Brochini Rodrigues, Professor(a) Adjunto(a)**, em 13/09/2024, às 16:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Guillermo Antonio Lobos Villagra, Professor(a) do Ensino Superior**, em 27/03/2025, às 15:31, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site <https://sei.ufscar.br/autenticacao>, informando o código verificador **1367670** e o código CRC **12F92DBB**.

Referência: Caso responda a este documento, indicar expressamente o Processo nº 23112.004445/2024-82

SEI nº 1367670

Modelo de Documento: Grad: Defesa TCC: Folha Aprovação, versão de 02/Agosto/2019

Agradecimentos

Não consigo expressar com palavras como foi difícil concluir este projeto, e por isso sou profundamente grata por ser firme diante de tantos obstáculos. Agradeço aos meus queridos pais, irmãs, familiares, meu namorado e os amigos que conquistei em São Carlos, assim como aos meus amigos do Industrial que sinto tanta falta.

Também é importante mencionar a minha gratidão à UFSCar, uma instituição que sempre foi um sonho para mim. Agradeço também ao meu orientador e a todos os professores com quem tive contato no Departamento de Matemática, bem como aos membros desta banca.

Essa experiência, foi de grande aprendizagem não apenas do ponto de vista acadêmico e profissional, mas também no âmbito pessoal. Aprendi que não devo desistir dos meus sonhos, no entanto, por vezes, é mais sensato aguardar, respirar e, só então, seguir em frente para concluir o que precisa ser feito.

Resumo

Este trabalho apresenta de forma clara como a Álgebra Linear é fundamental no processamento digital de imagens, uma área que surgiu junto com o avanço tecnológico e as técnicas de transmissão de imagens para facilitar sua visualização e análise. Com o aumento do número de imagens geradas e processadas digitalmente em diversas áreas, desde a medicina até a indústria automotiva, a Álgebra Linear tornou-se ainda mais importante, permitindo a representação e manipulação de imagens como matrizes numéricas. Para entender como a relação entre Álgebra Linear e imagens digitais funciona, foi necessário estudar a história e os conceitos essenciais dessa área. Além disso, exploraremos as aplicações das operações matriciais no processamento digital de imagens, com exemplos para ilustrar o desenvolvimento das técnicas em imagens. Também será apresentada uma aplicação prática da Álgebra Linear na Decomposição por valores singulares, que é uma técnica que facilita a compressão e reconstrução de imagens, além de ajudar no transporte. Este trabalho tem como objetivo apresentar as técnicas básicas do processamento digital de imagens, para fornecer uma base sólida para futuros estudos e por fim, explorar a parte geométrica da Álgebra Linear no contexto do processamento digital de imagens.

Palavras chave: Álgebra Linear, Matrizes, Processamento Digital de Imagens, Decomposição por valores singulares, Transformações Geométricas.

Abstract

This work clearly presents how Linear Algebra is fundamental in digital image processing, an area that emerged along with technological advances and image transmission techniques to facilitate their visualization and analysis. With the increase in the number of digitally generated and processed images in various areas, from medicine to the automotive industry, Linear Algebra has become even more important, allowing the representation and manipulation of images as numerical matrices. To understand how the relationship between Linear Algebra and digital images works, it was necessary to study the history and essential concepts of this area. In addition, we will explore the applications of matrix operations in digital image processing, with examples to illustrate the development of techniques in images. A practical application of Linear Algebra in Decomposition by singular values will also be presented, which is a technique that facilitates the compression and reconstruction of images, in addition to helping with transportation. This work aims to present the basic techniques of digital image processing, to provide a solid basis for future studies and, finally, to explore the geometric part of Linear Algebra in the context of digital image processing.

Keywords: Linear Algebra, Matrices, Digital Image Processing, Decomposition by singular values, Geometric Transformations.

Sumário

1	Processamento Digital de Imagens	2
1.1	Um pouco da história	2
1.2	O que é imagem digital?	3
1.2.1	Sistema de cores	4
2	Operações matriciais	7
2.1	Definição	7
2.2	Operações matriciais	8
2.2.1	Adição	8
2.2.1.1	A soma de imagens	9
2.2.2	Subtração	10
2.2.2.1	Subtração de imagens	11
2.2.3	Multiplicação por Escalar	12
2.2.3.1	Produto de uma imagem por um escalar	14
2.2.4	Produto entre matrizes	16
2.2.4.1	Multiplicação entre imagens	20
2.2.4.2	Multiplicação elemento por elemento de imagens	20
2.3	Aplicações: Normalização e Truncamento	22
2.4	Tipos de matrizes	30
2.4.1	Transposta	30
2.4.2	Identidade	33
2.4.3	Inversa	34
3	Decomposição de Valores Singulares	38
3.1	Definição	38
3.2	Subespaços vetoriais	39
3.3	Independência Linear e Bases	39
3.4	Produto interno	40
3.5	Autovetores e Autovalores	41
3.6	Diagonalização	42
3.7	Decomposição por Valores Singulares (DVS)	43
3.7.1	DVS aplicado em imagens	46

4	Transformações lineares	48
4.1	Definição	48
4.1.1	Núcleo e Imagem	51
4.1.2	Matrizes e Transformações Lineares	52
4.2	Transformações geométricas	53
4.2.1	Translação	53
4.2.1.1	Translação em imagens	54
4.2.2	Reflexão	56
4.2.2.1	Reflexão em imagens	60
4.2.3	Rotação	61
4.2.3.1	Rotação em imagens	64
	Referências Bibliográficas	67
A	Códigos utilizando OpenCV	69
A.0.1	Translação	69
A.0.2	Reflexão	70
A.0.3	Rotação	71

Introdução

A Álgebra Linear é um conteúdo extremamente importante na graduação de Matemática, com ela conseguimos resolver uma grande variedade de problemas matemáticos e aplicá-la em diversas áreas da ciência. É um dos problemas escolhidos para vermos suas aplicações é o processamento digital de imagens. Inicialmente exploramos a história dessa área de estudo, que surgiu no começo do século XX junto com a indústria jornalística projetando ideias para transmitir imagens via cabos submarinos. Também, no primeiro capítulo definimos o que é uma imagem digital para facilitar o entendimento ao relacionar imagens com matrizes nas próximas seções e explicar como funciona o sistema de cores para diferenciarmos imagens em tons de cinza de imagens coloridas.

No Capítulo 2, começamos a aprofundar as definições sobre matrizes, onde apresentamos as propriedades de cada operação matricial e após isso mostramos exemplos de como relacionar tais operações com imagens digitais. Neste segundo capítulo, abordamos a operação de multiplicação elemento por elemento em matrizes, que é utilizada de forma específica para exemplificar a segmentação de imagens. Essa técnica é possível porque uma imagem digital pode ser representada por uma matriz de valores que correspondem aos pixels, em que cada elemento da matriz representa o valor de intensidade do pixel em uma determinada posição da imagem. Dessa forma, a multiplicação elemento por elemento pode ser utilizada para realizar diversas operações de processamento de imagens.

No terceiro capítulo mostramos a Decomposição de Valores Singulares (DVS) e a utilização dela na colaboração para transmissão de imagens digitais. Para isso, iniciamos definindo espaços vetoriais e apresentando algumas definições e teoremas relevantes para a compreensão de como a DVS funciona. Enunciamos o teorema da decomposição DVS e o utilizamos para exemplificar como essa técnica pode ser aplicada em imagens. Apresentamos a importância da DVS na compressão de imagens, mostrando como ela é capaz de reduzir a quantidade de informações necessárias para representar uma imagem sem comprometer totalmente sua qualidade.

Por último, trabalhamos as transformações lineares para apresentar as transformações geométricas em imagens digitais. Utilizamos um pouco da computação para apresentar os códigos de (GARCÍA et al., 2015) para auxiliar nos exemplos. Nas considerações finais, fizemos uma breve recapitulação dos capítulos e refletimos sobre a contribuição deste trabalho para futuros estudos na área de processamento digital de imagens utilizando Álgebra Linear.

Capítulo 1

Processamento Digital de Imagens

1.1 Um pouco da história

O campo de processamento de imagens é vasto e abrange muitas áreas do conhecimento. De acordo com (GONZALEZ; WOODS, 2010), o início ocorreu no século XX, quando a indústria jornalística começou a usar cabos submarinos para transmitir imagens de Londres para Nova York. Essa forma de transmissão de imagem através de cabo foi chamada de sistema Bartlane e, no início da década de 1920, ajudou a reduzir o tempo necessário para transmitir imagens através do Oceano Atlântico, que antes levava mais de uma semana para cerca de três horas.

Naquela época, para que o sistema Bartlane funcionasse corretamente, era necessário usar uma ferramenta específica que reunia as imagens para o transporte através do cabo e, posteriormente, reestruturava as imagens ao recebê-las. Por exemplo, a imagem mostrada abaixo foi transportada e impressa utilizando fontes tipográficas para simular padrões de tons intermediários.

Figura 1.1: Fotografia digital produzida em 1921 (McFarlane)



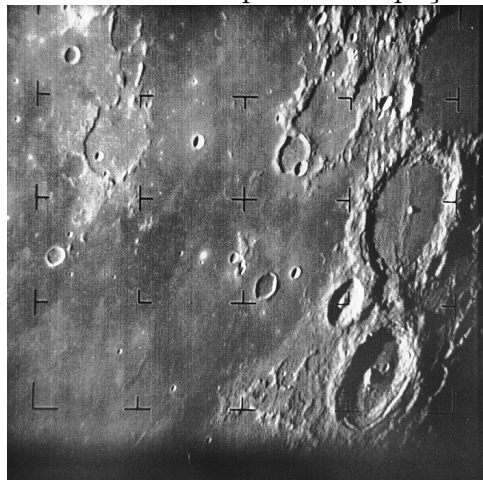
Fonte: (GONZALEZ; WOODS, 2010)

Podemos observar que o processamento de imagens passou por uma evolução significativa ao longo do tempo. Inicialmente, o sistema Bartlane foi capaz de codificar imagens apenas em 5 níveis de cinza, mas já em 1929, esta capacidade aumentou para 15 níveis.

Embora essa fosse uma grande conquista para a época, a manipulação de imagens ainda era feita de forma analógica, ou seja, por uma máquina fotográfica que utiliza processos manuais para ser revelada.

No entanto, com a chegada da tecnologia digital e o desenvolvimento dos primeiros computadores, a área de processamento de imagens começou a progredir rapidamente. O programa espacial norte-americano também ajudou no desenvolvimento dessa área. Em 1964, o Jet Propulsion Laboratory (Pasadena, Califórnia) começou a utilizar a computação para melhorar as imagens produzidas por sondas espaciais. Abaixo temos a primeira imagem da Lua capturada pelo Ranger 7 em 31 de julho de 1964.

Figura 1.2: A primeira foto da Lua por uma espaçonave norte-americana



Fonte: (GONZALEZ; WOODS, 2010)

Além disso, no final da década de 1960 e começo de 1970, esse ramo começou a ser aplicado também na Medicina, em imagens para análise médica o que permitiu o avanço no desenvolvimento de novos aparelhos de interpretação de imagens, o famoso equipamento para o Raio-X. Atualmente, várias áreas, como biologia e geografia, usam de técnicas do processamento para entender melhor seus objetos de estudos.

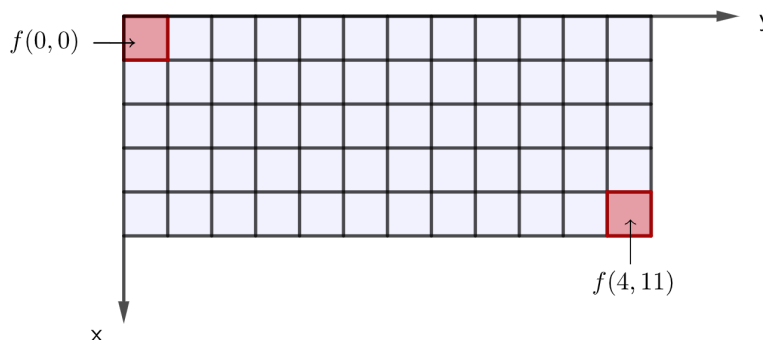
1.2 O que é imagem digital?

De acordo com (GONZALEZ; WOODS, 2010), uma imagem pode ser definida como uma função bidimensional $f(x, y)$, nas quais as coordenadas x e y no plano cartesiano representam a posição de cada pixel e a amplitude de f em cada par de coordenadas (x, y) é conhecida como a intensidade ou o nível de cinza da imagem no ponto correspondente. Quando as coordenadas x , y e os níveis de cinza são proporções finitas e discretas, a imagem é considerada uma imagem digital.

Uma imagem digital pode ser caracterizada por meio de uma matriz, onde cada elemento ij corresponde ao valor de intensidade do pixel na posição (i, j) . Em outras

palavras, cada entrada da matriz é representada por um valor que é o pixel e iremos exemplificar com a figura abaixo que caracteriza uma imagem digital de tamanho 5×12 .

Figura 1.3: Imagem digital com 5×12 pixels



Fonte: (JOY, 2020), adaptado.

Uma forma mais geral de representar a imagem acima é através de uma matriz $f(x, y)$ de dimensões $m \times n$, onde m é o número de linhas e n é o número de colunas da imagem. Essa matriz é definida da seguinte forma:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, n-1) \\ f(1, 0) & f(1, 1) & \dots & f(1, n-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(m-1, 0) & f(m-1, 1) & \dots & f(m-1, n-1) \end{bmatrix}.$$

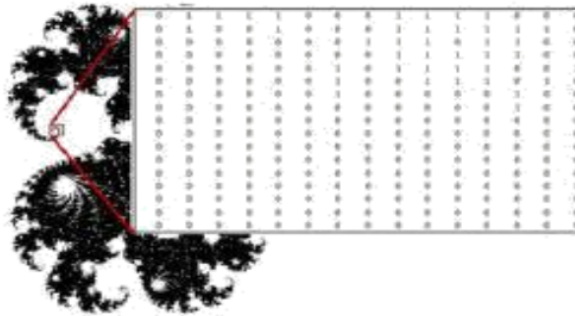
Em específico para a Figura 1.3, $m = 5$ e $n = 12$.

Resumindo, a imagem digital é uma representação matemática precisa e eficiente de uma imagem que permite sua manipulação e processamento por computadores. A definição acima torna possível a realização de diversas operações em imagens digitais, como melhoria de qualidade, compressão e transmissão de imagens, entre outras aplicações importantes no campo de processamento de imagens.

1.2.1 Sistema de cores

As imagens digitais podem trazer duas cores, como é o caso das imagens binárias ou booleanas. Digamos que uma imagem binária $f(x, y)$, cada pixel pode ser representado pelos valores 0 ou 1, que são tradicionalmente mapeados para preto e branco, respectivamente. Essa representação é feita por meio de uma matriz com entradas 0 ou 1. É importante notar que as imagens binárias são amplamente utilizadas em aplicações de processamento de imagens, especialmente em segmentação e detecção de bordas. A Figura 1.4 apresenta uma imagem binária exemplificando essa representação.

Figura 1.4: Imagem binária

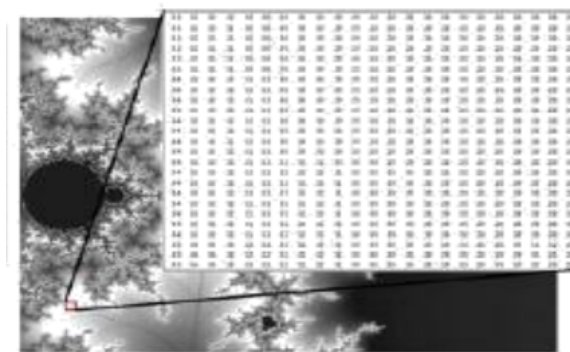


Fonte: (JOY, 2020)

As imagens em escala de cinza são usadas para representar a intensidade da luz e, portanto, o brilho. Em geral, os arquivos digitais utilizam números inteiros entre 0 e 255 para representar diferentes níveis de cinza, o que resulta em $256 = 2^8$ escalas de cinza diferentes. A Figura 1.5 apresenta uma imagem de 8 bits por pixel, o que significa que cada pixel pode ter 256 níveis de cinza diferentes.

Além disso, é importante destacar que as imagens em escala de cinza são amplamente utilizadas em aplicações de processamento de imagens, como a análise de imagens médicas, pois permitem uma melhor visualização das estruturas presentes na imagem sem a distração das cores.

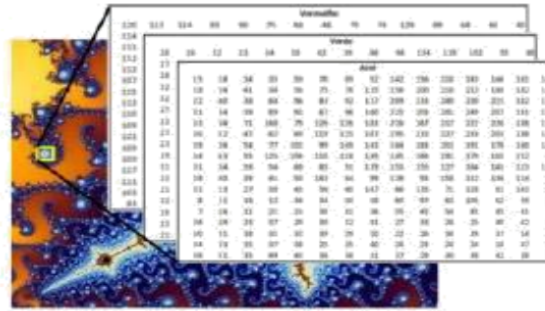
Figura 1.5: Imagem monocromática



Fonte: (JOY, 2020)

Quando se trata de imagens coloridas, a função $f(x, y)$ mede a intensidade de cor de cada pixel. Esta informação pode ser representada por três matrizes, cada uma delas representando a quantidade de vermelho, verde e azul presente na imagem. Este espaço de cores é conhecido como RGB, siglas para Red (vermelho), Green (verde) e Blue (azul), que representam as cores primárias que definem um espaço tridimensional com três direções ortogonais (R, G, B). Os valores inteiros utilizados nas matrizes variam entre 0 e 255. Assim, o sistema RGB permite representar $256^3 = 16.777.216$ cores diferentes. A Figura 1.6 (apresenta uma imagem colorida exemplificando essa representação.

Figura 1.6: Imagem digital colorida



Fonte: (JOY, 2020)

Por outro lado, temos o sistema de cores CMYK que é subtrativo, usado principalmente na impressão. Ele usa combinações de ciano, magenta, amarelo e preto para representar as cores. Diferentemente do RGB, quando todos os canais são misturados, produzem preto. É possível converter de um formato para o outro, mas a conversão pode resultar em perda de qualidade, pois os sistemas são diferentes e possuem um conjunto de cores distintos.

O sistema de cores de imagens digitais é fundamental para a representação de cores em dispositivos digitais, e a escolha do sistema correto depende do objetivo final da imagem, se para exibição em tela ou impressão.

Capítulo 2

Operações matriciais

Neste capítulo iremos apresentar os conceitos básicos necessários para compreender as técnicas do processamento de imagens. Serão mostradas definições, operações e propriedades de matrizes seguindo o livro de (BOLDRINI et al., 1980), (ANTON; RORRES, 2001) e (CALLIOLI; DOMINGUES, 1972). Utilizaremos essas definições para ilustrar exemplos de operações em imagens digitais, a fim de tornar o conteúdo mais acessível e prático.

2.1 Definição

Definimos como *matriz* um sistema formado por elementos dispostos em linhas e colunas. Esses elementos de uma matriz podem ser números (reais ou complexos), funções, ou até mesmo outras matrizes.

Definição 2.1. Sejam $A_{m \times n}$ uma matriz de m linhas e n colunas, e $a_{i,j}$ o elemento na i^a linha e j^a coluna. Então, a matriz $A_{m \times n}$ pode ser formalmente representada como:

$$A_{m \times n} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} = [a_{ij}]_{m \times n}.$$

Definição 2.2. As matrizes $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{r \times s}$ são consideradas iguais se possuírem o mesmo tamanho, isto é, se tiverem o mesmo número de linhas ($m = r$) e o mesmo número de colunas ($n = s$), e se todas as suas entradas correspondentes forem iguais. Isso pode ser representado como:

$$[a_{ij}] = [b_{ij}], \text{ para todo } i = 1, \dots, m \text{ e } j = 1, \dots, n.$$

2.2 Operações matriciais

As operações matriciais são uma parte fundamental da álgebra linear, que nos permite manipular e combinar matrizes de maneira eficiente e estruturada. Entre as operações que podem ser realizadas com matrizes, destacam-se a adição, subtração, multiplicação por escalar e multiplicação. Em seguida, faremos uma definição formal dessas operações, ilustrando-as com exemplos concretos e aplicando-as ao nosso campo de estudo principal, que é o processamento de imagens digitais.

2.2.1 Adição

Definição 2.3. Sejam A e B matrizes de mesma ordem, $A_{m \times n} = [a_{ij}]$ e $B_{m \times n} = [b_{ij}]$, resulta em uma nova matriz $m \times n$, denotada por $A + B$, cujos elementos são a soma dos elementos correspondentes de A e B . Em outras palavras,

$$A + B = [a_{ij} + b_{ij}]_{m \times n}, \text{ para todo } i = 1, \dots, m \text{ e } j = 1, \dots, n.$$

Exemplo 2.4.

$$\begin{bmatrix} 1 & 34 & 8 \\ 7 & 5 & 62 \\ 23 & 18 & 55 \end{bmatrix} + \begin{bmatrix} 0 & 3 & 1 \\ 2 & 47 & 2 \\ 16 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 37 & 9 \\ 9 & 52 & 64 \\ 39 & 19 & 60 \end{bmatrix}.$$

Ao analisarmos a adição de matrizes, podemos perceber que ela possui as mesmas propriedades da adição de números reais.

Propriedades 2.5. Dadas as matrizes A , B e C de mesma ordem $m \times n$, temos:

i) $A + B = B + A$ (comutatividade).

Demonstração.

Sejam $A = [a_{ij}]$ e $B = [b_{ij}]$ matrizes de mesma ordem, logo

$$\begin{aligned} A + B &= [a_{ij}] + [b_{ij}] \\ &= [a_{ij} + b_{ij}] \\ &= [b_{ij} + a_{ij}] \\ &= [b_{ij}] + [a_{ij}]. \end{aligned}$$

Como a_{ij} e b_{ij} são números reais então vale a propriedade comutativa. Então:

$$[b_{ij}] + [a_{ij}] = B + A.$$

Portanto, $A + B = B + A$.

□

ii) $A + (B + C) = (A + B) + C$ (associatividade).

Demonstração.

Sejam $A = [a_{ij}]$, $B = [b_{ij}]$ e $C = [c_{ij}]$ matrizes de mesma ordem,

$$\begin{aligned} A + (B + C) &= [a_{ij}] + ([b_{ij}] + [c_{ij}]) \\ &= [a_{ij}] + [b_{ij} + c_{ij}] \\ &= ([a_{ij}] + [b_{ij} + c_{ij}]) \\ &= [a_{ij} + b_{ij} + c_{ij}] \\ &= ([a_{ij} + b_{ij}] + [c_{ij}]) \\ &= ([a_{ij} + b_{ij}]) + [c_{ij}] \\ &= ([a_{ij}] + [b_{ij}]) + [c_{ij}]. \end{aligned}$$

Como a_{ij} , b_{ij} e c_{ij} são números reais então vale a propriedade associativa.

Logo, $A + (B + C) = (A + B) + C$.

□

iii) $A + 0 = A$, onde 0 denota a matriz nula $m \times n$.

Demonstração.

Sejam $A = [a_{ij}]$ e $0 = [b_{ij}]$, matrizes de mesma ordem, onde os elementos da matriz 0 são todos nulos.

Então, $A + 0 = C$, sendo $C = [c_{ij}]$ e seus elementos são:

$$\begin{aligned} [c_{ij}] &= [a_{ij}] + [b_{ij}], \\ &= [a_{ij} + b_{ij}], \\ &= [a_{ij} + 0], \\ &= [a_{ij}] = A. \end{aligned}$$

Portanto, $A + 0 = A$.

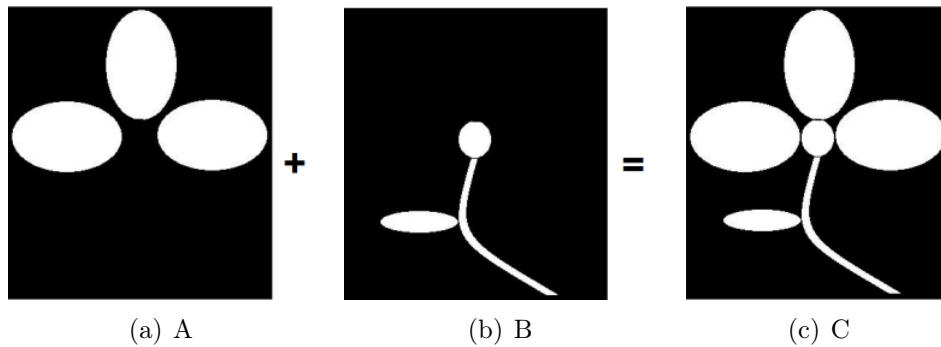
□

2.2.1.1 A soma de imagens

Segundo a definição da soma de matrizes, podemos comparar este processo com a junção de duas imagens de mesmo tamanho. Assim, sejam A e B duas matrizes (que representam imagens) de mesmo tamanho, ao somá-las, temos $A + B = C$, onde C é uma nova imagem com as características de ambas A e B .

A Figura 2.1 ilustra um exemplo concreto da adição de imagens.

Figura 2.1: Adição de imagens



(a) A

(b) B

(c) C

Fonte: (SILVA, 2014)

Além disso, a soma de matrizes também pode resultar na sobreposição de duas imagens, como demonstra a Figura 2.2, que apresenta a adição de duas imagens coloridas.

Figura 2.2: Adição de imagens coloridas



(a)

(b)

(c)

Fonte: (JOY, 2020)

2.2.2 Subtração

Definição 2.6. Sejam A e B matrizes de mesma ordem, a diferença $A - B$ é obtida pela subtração dos elementos de B pelos elementos correspondentes em A .

Usando a notação de matrizes, se $A = [a_{ij}]$ e $B = [b_{ij}]$, então

$$A - B = [a_{ij} - b_{ij}]_{m \times n}, \text{ para todo } i = 1, \dots, m \text{ e } j = 1, \dots, n..$$

Exemplo 2.7.

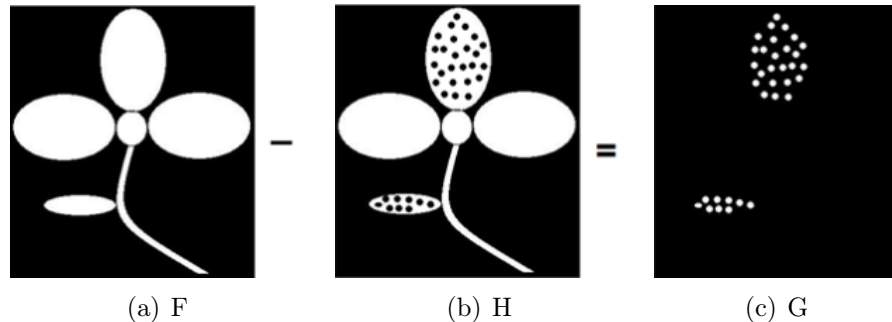
$$\begin{bmatrix} 1 & 34 & 8 \\ 7 & 5 & 62 \\ 23 & 18 & 55 \end{bmatrix} - \begin{bmatrix} 0 & 3 & 1 \\ 2 & 47 & 2 \\ 16 & 1 & 5 \end{bmatrix} = \begin{bmatrix} 1 & 31 & 7 \\ 5 & -42 & 60 \\ 7 & 17 & 50 \end{bmatrix}.$$

2.2.2.1 Subtração de imagens

A subtração de imagens segue a mesma lógica da subtração de matrizes. Sejam F e H duas matrizes (representando imagens) de mesma ordem, então a diferença $F - H = G$ resulta em uma nova imagem G que contém as características de F , porém sem aquelas que são comuns a H .

A Figura 2.3 ilustra o processo da subtração de imagens.

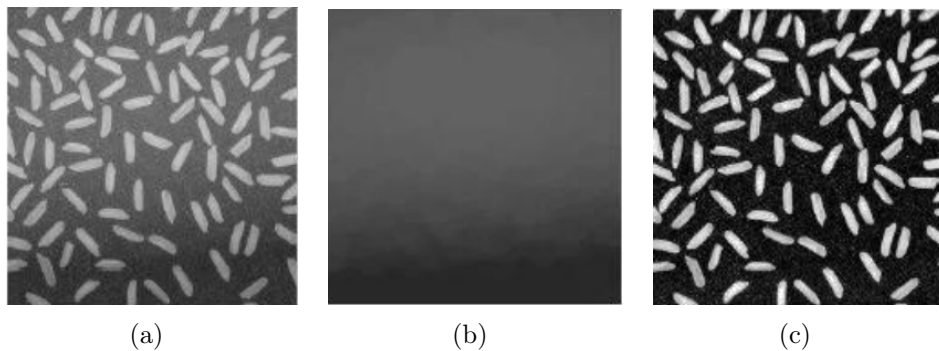
Figura 2.3: Subtração de duas imagens



Fonte: (SILVA, 2014)

Além disso, a subtração de uma imagem com outra pode ser usada para retirar o fundo. Podemos ver na Figura 2.4, onde vemos a subtração da imagem (b) à imagem (a) que equivale à imagem (c) sem fundo.

Figura 2.4: Subtração do fundo de uma imagem



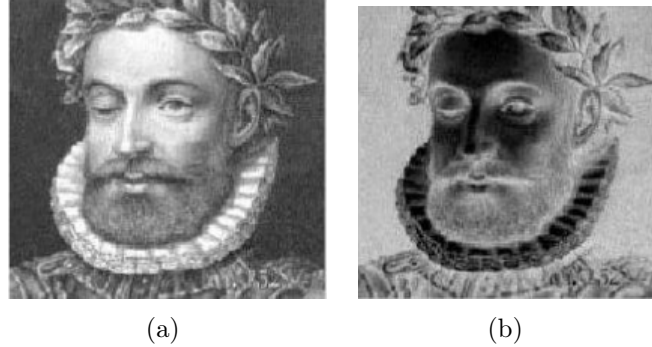
Fonte: (JOY, 2020)

A Figura 2.5 apresenta uma imagem negativa, que é obtida pela subtração de uma imagem em escala de cinza por uma imagem $255 \times I$ (produto de 255 pela matriz identidade). Essa técnica é amplamente utilizada em processamento de imagem para realçar o contraste e destacar certos detalhes na imagem.

A imagem $255 \times I$ é uma imagem que resulta da multiplicação de uma matriz identidade por 255. Em outras palavras, obtemos uma matriz em que todos os elementos na diagonal principal são iguais a 255 e todos os demais elementos são iguais a 0, resultando em uma imagem em que todos os pixels têm o valor máximo permitido pela escala de cinza.

Ao subtrair a imagem em escala de cinza da imagem $255 \times I$, estamos essencialmente subtraindo cada pixel da imagem em escala de cinza por 255. Isso tem o efeito de inverter a intensidade de cada pixel na imagem em escala de cinza, resultando em uma imagem negativa.

Figura 2.5: Subtração de uma imagem por $255 \times I$



Fonte: (JOY, 2020)

2.2.3 Multiplicação por Escalar

Definição 2.8. Sejam $A = [a_{ij}]_{m \times n}$ e $k \in \mathbb{R}$, então definimos uma nova matriz como

$$k \cdot A = [ka_{ij}]_{m \times n}, \text{ para todo } i = 1, \dots, m \text{ e } j = 1, \dots, n.$$

Exemplo 2.9.

$$8 \cdot \begin{bmatrix} 0 & 1 \\ 9 & -4 \end{bmatrix} = \begin{bmatrix} 8 \cdot 0 & 8 \cdot 1 \\ 8 \cdot 9 & 8 \cdot (-4) \end{bmatrix} = \begin{bmatrix} 0 & 8 \\ 72 & -32 \end{bmatrix}.$$

Propriedades 2.10. Dadas matrizes A e B de mesma ordem $m \times n$ e escalares k, k_1 e k_2 , temos:

$$i) \quad k(A + B) = kA + kB.$$

Demonstração.

Sejam $A = [a_{ij}]_{m \times n}$ e $B = [b_{ij}]_{m \times n}$ matrizes de mesmo tamanho e k um escalar. Então, temos:

$$\begin{aligned} k \cdot (A + B) &= k \cdot ([a_{ij}] + [b_{ij}]), \\ &= k \cdot [a_{ij} + b_{ij}], \\ &= [ka_{ij} + kb_{ij}], \\ &= k[a_{ij}] + k[b_{ij}], \\ &= kA + kB. \end{aligned}$$

Isto ocorre, pois a_{ij}, b_{ij} são números reais e a multiplicação escalar é associativa, podemos então aplicar a propriedade distributiva. Portanto, $k \cdot (A + B) = kA + kB$.

□

$$ii) (k_1 + k_2) \cdot A = k_1A + k_2A.$$

Demonstração.

Sejam $A = [a_{ij}]$ uma matriz e k_1, k_2 números. Como a multiplicação escalar é associativa, podemos realizar a distributiva de $(k_1 + k_2)$ dos dois lados da equação, então:

$$\begin{aligned} (k_1 + k_2) \cdot A &= (k_1 + k_2) \cdot [a_{ij}], \\ &= [(k_1 + k_2)a_{ij}], \\ &= [k_1a_{ij} + k_2a_{ij}], \\ &= [k_1a_{ij}] + [k_2a_{ij}], \\ &= k_1[a_{ij}] + k_2[a_{ij}], \\ &= k_1A + k_2A. \end{aligned}$$

□

iii) $0 \cdot A = 0_{m \times n}$, isto é, se multiplicarmos o número zero por qualquer matriz A , teremos a matriz nula.

Demonstração.

Sejam $A = [a_{ij}]$ e $0 = [b_{ij}]$ matrizes de mesma ordem, tal que os elementos da matriz $[b_{ij}]$ são todos iguais a zero. Então:

$$\begin{aligned} 0 \cdot A &= [b_{ij}] \cdot [a_{ij}], \\ &= [b_{ij} \cdot a_{ij}], \\ &= [0 \cdot a_{ij}], \\ &= [0], \\ &= [b_{ij}] = 0. \end{aligned}$$

Assim, todos os elementos da matriz $0 \cdot A$ são iguais a zero, e portanto, a matriz $0 \cdot A$ é a matriz nula. Portanto, a afirmação de que $0 \cdot A = 0$ é verdadeira para todas as matrizes A .

□

$$iv) k_1 \cdot (k_2A) = (k_1k_2) \cdot A.$$

Demonstração.

Sejam k_1, k_2 escalares e $A = [a_{ij}]$ uma matriz. Assim, temos

$$\begin{aligned} k_1 \cdot (k_2 A) &= k_1 \cdot (k_2 [a_{ij}]), \\ &= k_1 \cdot ([k_2 a_{ij}]), \\ &= (k_1 \cdot [k_2 a_{ij}]), \\ &= ([k_1 k_2 a_{ij}]), \\ &= (k_1 k_2 [a_{ij}]), \\ &= (k_1 k_2) \cdot [a_{ij}] = (k_1 k_2) \cdot A. \end{aligned}$$

□

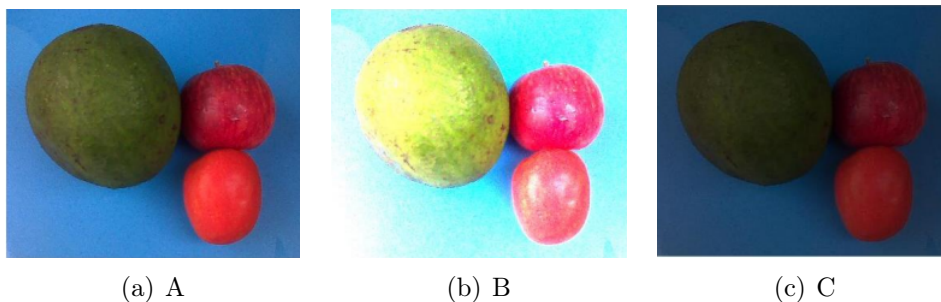
2.2.3.1 Produto de uma imagem por um escalar

Multiplicar uma imagem por um escalar é uma técnica que pode ser utilizada para realçar as diferenças entre imagens que possuem níveis de intensidades diferentes. Essa operação consiste em multiplicar cada valor de pixel da imagem por um valor escalar, resultando em uma imagem cujos níveis de intensidade são amplificados ou reduzidos proporcionalmente.

Essa técnica é frequentemente utilizada em processamento de imagens para melhorar a visualização de detalhes em imagens com baixo contraste ou para ajustar a escala de valores de pixels em imagens com diferentes faixas de intensidade. Além disso, o produto por escalar também pode ser utilizado para normalizar imagens ou para reduzir ruídos em imagens com baixa qualidade.

Uma aplicação para mostrar como funciona esse método é feito em uma sequência didática de (SILVA, 2014), na qual, mostra um exemplo de uma imagem A multiplicada pelos números reais 3 e 0,5 que podemos ver na Figura 2.6, sendo $B = 3 \cdot A$ e $C = 0,5 \cdot A$.

Figura 2.6: Produto de uma imagem por um escalar

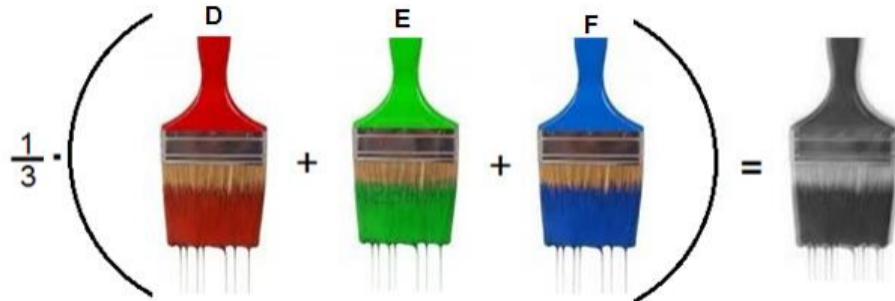


Fonte: (SILVA, 2014)

Além disso, (SILVA, 2014) apresenta um exemplo interessante que consiste na aplicação da média aritmética de três imagens D, E e F . Essas imagens são idênticas, exceto

pela mudança de cor em cada uma, sendo vermelho, verde e azul, respectivamente. O resultado é uma imagem em tons de cinza e podemos visualizar na Figura 2.7.

Figura 2.7: Média aritmética



Fonte: (SILVA, 2014)

O exemplo apresentado acima demonstra como a combinação de diferentes canais de cores pode levar a uma imagem em escala de cinza. A média aritmética é apenas um exemplo de técnica de combinação de canais de cores.

Utilizando a multiplicação de uma matriz por um escalar e a adição de matrizes é possível misturar imagens e criar efeitos de transição de imagem. Vimos anteriormente que a multiplicação por um escalar pode ser utilizada para aumentar ou diminuir o brilho de uma imagem, enquanto a adição de matrizes pode ser usada para combinar duas imagens.

Um exemplo disso é dado por (GOMES; VELHO, 1994) que define a operação de "dissolve"¹:

Definição 2.11. Dadas duas imagens f e g , e um número real $0 \leq t \leq 1$, definimos o dissolve h_t de f e g :

$$h_t = \text{dissolve}_t(f, g) = (1 - t)f + tg.$$

A Figura 2.8 mostra o resultado da aplicação da função "dissolve" em duas imagens com um parâmetro de mistura $t = 0,4$ que controla a proporção de cada imagem na mistura. Podemos verificar na equação abaixo:

$$\begin{aligned} h_t &= (1 - t)f + tg \\ &= (1 - 0,4) \cdot f + 0,4 \cdot g \\ &= 0,6 \cdot f + 0,4 \cdot g. \end{aligned}$$

Nesse caso específico, a imagem resultante é uma combinação das duas imagens originais, onde 60% da imagem (a) é misturada com 40% da imagem (b).

¹Esse é o conceito de homotopia: Sejam X e Y espaços topológicos. Duas funções contínuas $f, g \in C(X; Y)$ são chamadas de homotópicas, representa-se por $f \simeq g$, se existe uma função $H : X \times [0, 1] \rightarrow Y$ tal que $H(x, 0) = f(x)$ e $H(x, 1) = g(x)$. Na definição $H = h_t$.

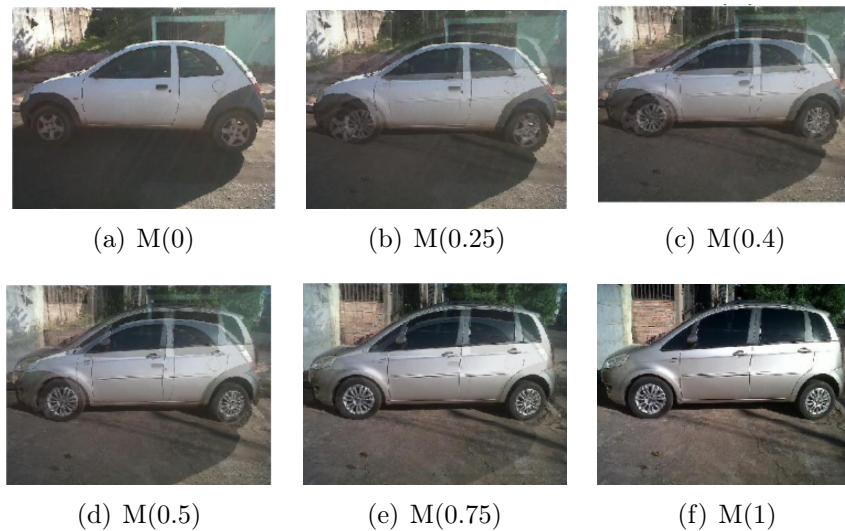
Figura 2.8: Mistura de duas imagens



Fonte: (GOMES; VELHO, 1994)

Observamos que para $t = 0$, a função "dissolve" resulta na primeira imagem f , e para $t = 1$, resulta na segunda imagem g . Outro exemplo dessa operação é feito em (SILVA, 2014), no qual, realiza a transição de duas imagens coloridas.

Figura 2.9: Transição de imagens



Fonte: (GOMES; VELHO, 1994)

A imagem $M(0)$ é a imagem f , $M(0, 25)$, $M(0, 4)$, $M(0, 5)$, $M(0, 75)$ são a misturas das imagens f e de g e, $M(1)$ é a imagem g .

2.2.4 Produto entre matrizes

O produto entre matrizes é uma operação que consiste em multiplicar duas matrizes. Para que possam ser multiplicadas, o número de colunas da primeira matriz deve ser igual ao número de linhas da segunda matriz. O elemento ij da matriz resultante é obtido multiplicando cada elemento da i -ésima linha da primeira matriz pelo elemento correspondente da j -ésima coluna da segunda matriz, e somando os resultados.

Esta é uma ferramenta importante em diversas áreas da matemática e da tecnologia, incluindo, o processamento de imagens que mostraremos após as definições.

Definição 2.12. Sejam $A = [a_{ij}]_{m \times n}$ e $B = [b_{rs}]_{n \times p}$. Definimos $A \cdot B = [c_{uv}]_{m \times p}$ tal que

$$c_{uv} = \sum_{k=1}^n a_{uk} b_{kv} = a_{u1} b_{1v} + \dots + a_{un} b_{nv}.$$

Exemplo 2.13.

$$\begin{bmatrix} 1 & 4 \\ 7 & 5 \\ 3 & 1 \end{bmatrix}_{3 \times 2} \cdot \begin{bmatrix} 0 & 1 \\ -9 & 7 \end{bmatrix}_{2 \times 2} = \begin{bmatrix} 1 \cdot 0 + 4 \cdot (-9) & 1 \cdot 1 + 4 \cdot 7 \\ 7 \cdot 0 + 5 \cdot (-9) & 7 \cdot 1 + 5 \cdot 7 \\ 3 \cdot 0 + 1 \cdot (-9) & 3 \cdot 1 + 1 \cdot 7 \end{bmatrix}_{3 \times 2} = \begin{bmatrix} -36 & 29 \\ -45 & 42 \\ -9 & 10 \end{bmatrix}.$$

Propriedades 2.14.

i) Em geral $AB \neq BA$.

Demonstração.

Para mostrar que esta propriedade é válida, é suficiente apresentar um exemplo de matrizes A e B tais que $AB \neq BA$.

Considere as seguintes matrizes:

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Então, temos:

$$AB = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix},$$

e

$$BA = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

Como $AB \neq BA$, temos que a igualdade $AB = BA$ não é válida para todas as matrizes A e B .

□

ii) $A \cdot (B + C) = AB + AC$.

Demonstração.

Sejam $A = [a_{ij}]$, $B = [b_{rs}]$ e $C = [c_{rs}]$ matrizes. Também, sejam $B + C = D = [d_{rs}]$, $AB = E = [e_{is}]$ e $AC = F = [f_{is}]$ então:

$$d_{rs} = b_{rs} + c_{rs}, \quad e_{is} = \sum_{k=1}^n a_{ik}b_{ks} \quad \text{e} \quad f_{is} = \sum_{k=1}^n a_{ik}c_{ks}.$$

Com isso, podemos escrever a soma dos elementos e_{is} e f_{is} como:

$$\begin{aligned} e_{is} + f_{is} &= \sum_{k=1}^n a_{ik}b_{ks} + \sum_{k=1}^n a_{ik}c_{ks}, \\ &= \sum_{k=1}^n (a_{ik}b_{ks} + a_{ik}c_{ks}), \\ &= \sum_{k=1}^n a_{ik} \cdot (b_{ks} + c_{ks}). \end{aligned}$$

Observamos que a última igualdade se segue diretamente da distributividade da multiplicação em relação à soma.

Por fim, para cada i e s , multiplicando os elementos a_{ij} e d_{rs} , temos:

$$\begin{aligned} \sum_{k=1}^n a_{ik}d_{ks} &= \sum_{k=1}^n a_{ik} \cdot (b_{ks} + c_{ks}), \\ &= \sum_{k=1}^n a_{ik}b_{ks} + \sum_{k=1}^n a_{ik}c_{ks}, \\ &= e_{is} + f_{is}. \end{aligned}$$

Portanto, concluímos que $A \cdot (B + C) = AB + AC$, como queríamos demonstrar. \square

Observação: Analogamente, se A e B são matrizes $m \times n$ e C uma matriz $n \times p$, temos $(A + B) \cdot C = AC + BC$.

iii) $(AB) \cdot C = A \cdot (BC)$ (associatividade)

Demonstração.

Sejam $A = [a_{ij}]$, $B = [b_{jk}]$ e $C = [c_{ks}]$ matrizes. Também, sejam $AB = D = [d_{ik}]$ e $BC = E = [e_{js}]$, onde

$$d_{ik} = \sum_{l=1}^n a_{il}b_{lk}, \quad e_{js} = \sum_{l=1}^p b_{jl}c_{ls}.$$

Para mostrar que $(AB) \cdot C = A \cdot (BC)$, precisamos mostrar que as matrizes resul-

tantes das operações são iguais. Assim, temos:

$$(AB) \cdot C = [d_{ik}] \cdot [c_{ks}] = [f_{is}],$$

onde

$$f_{is} = \sum_{k=1}^p d_{ik} c_{ks} = \sum_{k=1}^p \sum_{l=1}^n (a_{il} b_{lk}) c_{ks}.$$

Por outro lado, temos

$$A \cdot (BC) = [a_{ij}] \cdot [e_{js}] = [g_{is}],$$

onde

$$g_{is} = \sum_{j=1}^n a_{ij} e_{js} = \sum_{j=1}^n \sum_{l=1}^p a_{ij} (b_{jl} c_{ls}).$$

Logo, para mostrar que $(AB) \cdot C = A \cdot (BC)$, precisamos mostrar que $f_{is} = g_{is}$ para todo i e s . Assim, temos:

$$\begin{aligned} f_{is} &= \sum_{k=1}^p \sum_{l=1}^n (a_{il} b_{lk}) c_{ks}, \\ &= \sum_{l=1}^n a_{il} \sum_{k=1}^p b_{lk} c_{ks}, \\ &= \sum_{l=1}^n a_{il} e_{ls}, \\ &= g_{is}, \end{aligned}$$

onde usamos a definição de e_{js} e a associatividade da multiplicação de matrizes. Portanto, $(AB) \cdot C = A \cdot (BC)$. \square

iv) $0 \cdot A = 0$ e $A \cdot 0 = 0$

Demonstração.

Observe que pela propriedade distributiva demonstrada em *ii)*, temos

$$A \cdot 0 = A \cdot (0 + 0) = A \cdot 0 + A \cdot 0.$$

Agora subtraindo $A \cdot 0$ em ambos os lados da igualdade acima, tem-se que $A \cdot 0 = 0$. Analogamente, para $0 \cdot A$ temos:

$$0 \cdot A = (0 + 0) \cdot A = 0 \cdot A + 0 \cdot A.$$

Subtraindo $0 \cdot A$ dos dois lados da igualdade, notamos que $0 \cdot A = 0$.

Portanto, $0 \cdot A = 0$ e $A \cdot 0 = 0$. \square

2.2.4.1 Multiplicação entre imagens

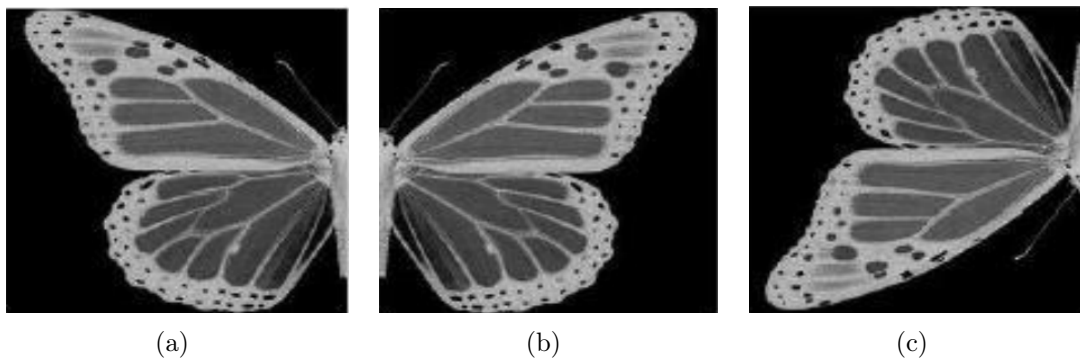
Com a multiplicação podemos inverter uma imagem, no processo abaixo como em (JOY, 2020) foi feita a multiplicação de uma imagem pela matriz J que é chamada de matriz contra identidade, na qual, é uma matriz de ordem n com elementos da diagonal secundária todos 1 e 0 nas outras entradas, exceto em j_{1n} e j_{n1} , que são 1.

É representada por:

$$J = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \dots & 0 \end{bmatrix}.$$

A multiplicação da imagem (a) pela matriz J é ilustrada na Figura 2.10. Na imagem (b), a matriz J é multiplicada pela direita, enquanto na imagem (c) é multiplicada pela esquerda.

Figura 2.10: Multiplicação de duas imagens



Fonte: (JOY, 2020)

2.2.4.2 Multiplicação elemento por elemento de imagens

Com base em (JOY, 2020), é possível estabelecer uma definição de multiplicação elemento por elemento que será utilizada de forma específica na aplicação de imagens digitais. É fundamental ressaltar que essa operação está sendo definida neste trabalho exclusivamente para esse propósito.

A multiplicação elemento por elemento de duas matrizes A e B , ambas de ordem $m \times n$, é definida como o produto de cada elemento da matriz A com o elemento correspondente da matriz B . Essa operação será representada por $A \odot B$ e resulta em uma nova matriz de ordem $m \times n$, onde cada elemento c_{ij} é dado por $c_{ij} = a_{ij} \cdot b_{ij}$.

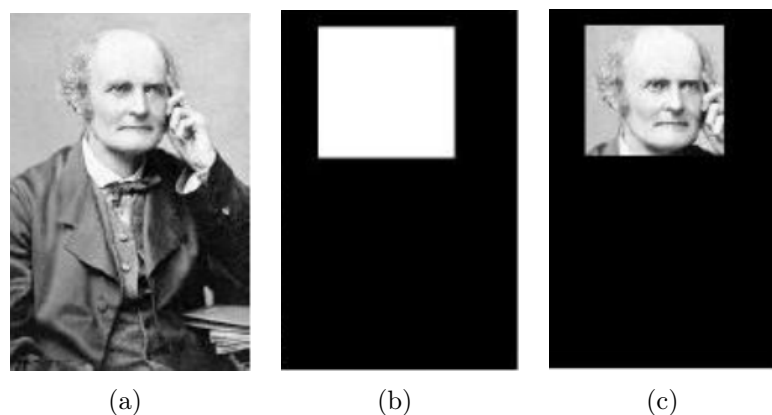
$$\begin{aligned}
 A \odot B &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \odot \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}, \\
 &= \begin{bmatrix} a_{11} \cdot b_{11} & a_{12} \cdot b_{12} & \dots & a_{1n} \cdot b_{1n} \\ a_{21} \cdot b_{21} & a_{22} \cdot b_{22} & \dots & a_{2n} \cdot b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} \cdot b_{m1} & a_{m2} \cdot b_{m1} & \dots & a_{mn} \cdot b_{mn} \end{bmatrix}.
 \end{aligned}$$

Como uma imagem digital pode ser representada por uma matriz de valores de intensidade de pixels, em que cada elemento da matriz representa o valor de intensidade do pixel em uma determinada posição na imagem, então a multiplicação elemento por elemento pode ser usada para realizar diversas operações de processamento de imagens.

Uma aplicação prática da multiplicação elemento por elemento em imagens digitais é a segmentação de imagens, em que se deseja separar a imagem em regiões ou objetos distintos. Para isso, pode-se utilizar uma matriz binária como máscara, em que os valores dos elementos da matriz são 1 para os pixels que devem ser mantidos na imagem segmentada e 0 para os pixels que devem ser removidos.

Essa máscara é, na verdade, uma imagem preta com o mesmo tamanho da imagem original, na qual apenas a região de interesse é definida como branca, enquanto o restante da imagem é preenchido com pixels pretos. O exemplo da Figura 2.11 mostra como funciona a segmentação da imagem (a) com a máscara (b) produzindo a terceira imagem (c).

Figura 2.11: Aplicação de máscara em uma imagem



Fonte: (JOY, 2020)

2.3 Aplicações: Normalização e Truncamento

Notamos que na seção 2.2.1.1 pela Figura 2.2 que em (c) há uma tonalidade clara no centro da imagem. Quando realizamos operações com imagens, podemos ter problemas em relação ao intervalo dos pixels na imagem resultante. Em (SIQUEIRA, 2014), esses problemas são chamados de *underflow* e *overflow*, nos quais, as entradas da matriz final podem ter valores negativos ou superiores do intervalo permitido. Para isso temos duas possibilidades para manter as entradas dentro do intervalo correto: o truncamento e a normalização.

A técnica de truncamento se refere a cortar os valores superiores e inferiores do máximo e mínimo do intervalo escolhido para o valor mínimo e máximo. Então, em uma imagem com L tons, os maiores valores que $L-1$ serão substituídos por $L-1$ e os valores negativos substituídos por 0 e isso pode produzir na imagem um destaque nas cores preto e branco perdendo algumas características da imagem.

Já o processo de normalização da imagem se refere a padronizar os valores da matriz utilizando um outro intervalo cujo extremo inferior e superior é igual, respectivamente, ao menor e maior valor da matriz. Assim, para uma imagem de L tons com intervalo $I = [0, L-1]$, utilizaremos os valores de mínimo c_{min} e máximo c_{max} de I como um intervalo novo I_1 para normalizar a matriz. A equação utilizada por (SIQUEIRA, 2014) adaptada para o processo é:

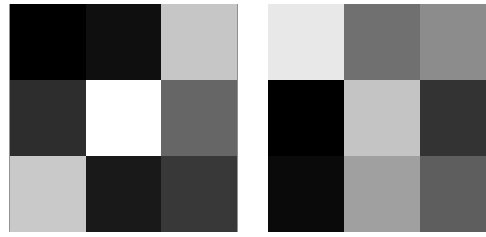
$$C_{norm} = \frac{L-1}{c_{max} - c_{min}} \cdot (c_n - c_{min}),$$

sendo C matriz resultante da operação, c_n cada entrada da matriz C e C_{norm} a matriz normalizada.

Vamos utilizar de exemplos de imagens em escala de 256 níveis de cinza, na qual, cada pixel está entre 0 e 255, para os processos acima. Começaremos com a operação de soma de imagens com um exemplo numérico e visual para matrizes 3×3 (pixels):

Exemplo 2.15. Sejam $A = \begin{bmatrix} 0 & 15 & 200 \\ 45 & 255 & 102 \\ 201 & 26 & 55 \end{bmatrix}$ e $B = \begin{bmatrix} 233 & 113 & 141 \\ 0 & 197 & 52 \\ 10 & 160 & 95 \end{bmatrix}$ as matrizes

respectivas das imagens abaixo:

Figura 2.12: Imagens das matrizes A e B (a) A (b) B

Fonte: (Autorial, 2023)

- (a) Obtenha a soma das matrizes
- A
- e
- B
- .

$$A + B = C$$

$$\begin{bmatrix} 0 & 15 & 200 \\ 45 & 255 & 102 \\ 201 & 26 & 55 \end{bmatrix} + \begin{bmatrix} 233 & 113 & 141 \\ 0 & 197 & 52 \\ 10 & 160 & 95 \end{bmatrix} = \begin{bmatrix} 233 & 128 & 341 \\ 45 & 452 & 154 \\ 211 & 186 & 150 \end{bmatrix}.$$

Obs: Veja que a matriz resultante C em algumas entradas ultrapassa o intervalo permitido $I = [0, 255]$ e por isso não conseguimos exibir a imagem resultante.

- (b) Realize as técnicas de truncamento e normalização para que a imagem
- C
- seja exibida.

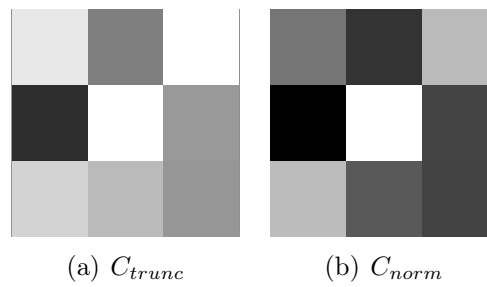
Para o truncamento iremos alterar na matriz C os valores que forem maiores que 255 por 255:

$$C_{trunc} = \begin{bmatrix} 233 & 128 & 255 \\ 45 & 255 & 154 \\ 211 & 186 & 150 \end{bmatrix}.$$

Para normalizar, iremos utilizar o intervalo $[c_{min}, c_{max}] = [45, 452]$ para que a matriz mantenha-se no intervalo correto: $[0, 255]$:

$$\begin{aligned} C_{norm} &= \frac{L - 1}{c_{max} - c_{min}} \cdot (c_n - c_{min}), \\ &= \frac{255}{452 - 45} \cdot \left(\begin{bmatrix} (233 - 45) & (128 - 45) & (341 - 45) \\ (45 - 45) & (452 - 45) & (154 - 45) \\ (211 - 45) & (186 - 45) & (150 - 45) \end{bmatrix} \right), \\ &= \frac{255}{407} \cdot \begin{bmatrix} 188 & 83 & 296 \\ 0 & 407 & 109 \\ 166 & 141 & 105 \end{bmatrix} = \begin{bmatrix} 118 & 52 & 185 \\ 0 & 255 & 68 \\ 104 & 88 & 66 \end{bmatrix}. \end{aligned}$$

As imagens das técnicas são:

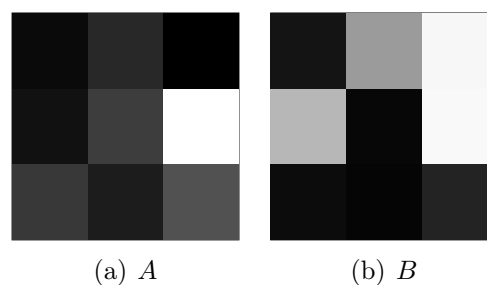
Figura 2.13: Imagem da matriz C truncada e normalizada

Fonte: (Autorial, 2023)

A consequência desse exemplo é a comparação entre os dois processos. Vemos que os dois métodos exibem a imagem C , porém pela técnica de truncamento há tons mais próximos ao branco ao contrário de (b) que mostra a imagem em mais tons de cinza, devido a definição.

Para a subtração de imagens iremos corrigir os valores dos pixels de cada entrada da imagem (matriz) quando algum valor ficar negativo.

Exemplo 2.16. Sejam $A = \begin{bmatrix} 10 & 41 & 0 \\ 18 & 61 & 255 \\ 56 & 29 & 82 \end{bmatrix}$ e $B = \begin{bmatrix} 20 & 156 & 247 \\ 183 & 7 & 251 \\ 14 & 5 & 36 \end{bmatrix}$ as matrizes respectivas das imagens abaixo:

Figura 2.14: Imagens das matrizes A e B 

Fonte: (Autorial, 2023)

(a) Obtenha a subtração das matrizes A e B .

$$A - B = C$$

$$\begin{bmatrix} 10 & 41 & 0 \\ 18 & 61 & 255 \\ 56 & 29 & 82 \end{bmatrix} - \begin{bmatrix} 20 & 156 & 247 \\ 183 & 7 & 251 \\ 14 & 5 & 36 \end{bmatrix} = \begin{bmatrix} -10 & -115 & -247 \\ -165 & 54 & 4 \\ 42 & 24 & 46 \end{bmatrix}.$$

Obs: Veja que a matriz resultante C tem algumas entradas com valor negativo e por isso não conseguimos exibir a imagem.

- (b) Realize as técnicas de truncamento e normalização para que a imagem C seja exibida.

Para o truncamento, na matriz C os valores negativos são anulados:

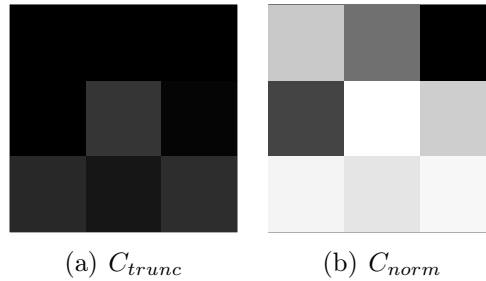
$$C_{trunc} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 54 & 4 \\ 42 & 24 & 46 \end{bmatrix}.$$

Para normalizar, iremos utilizar o intervalo $[c_{min}, c_{max}] = [-247, 54]$:

$$\begin{aligned} C_{norm} &= \frac{L-1}{c_{max} - c_{min}} \cdot (c_n - c_{min}), \\ &= \frac{255}{54 - (-247)} \cdot \left(\begin{bmatrix} (-10 - (-247)) & (-115 - (-247)) & (-247 - (-247)) \\ (-165 - (-247)) & (54 - (-247)) & (-4 - (-247)) \\ (42 - (-247)) & (24 - (-247)) & (46 - (-247)) \end{bmatrix} \right), \\ &= \frac{255}{301} \cdot \begin{bmatrix} 237 & 132 & 0 \\ 82 & 301 & 243 \\ 289 & 271 & 293 \end{bmatrix} = \begin{bmatrix} 201 & 112 & 0 \\ 69 & 255 & 206 \\ 245 & 230 & 248 \end{bmatrix}. \end{aligned}$$

As imagens dos processos acima são:

Figura 2.15: Imagem da matriz C truncada e normalizada

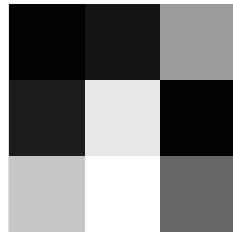


Fonte: (Autorial, 2023)

Quando a imagem é truncada há partes mais em preto e tons de cinza mais escuros devido a troca dos valores negativo pelo 0 que é a cor preto. Enquanto na imagem normalizada, algumas partes ficam mais claras representando de certa forma melhor as cores da figura A que não são comuns a imagem B .

Para multiplicação por escalar, também podemos utilizar as aplicações. Segue outro exemplo:

Exemplo 2.17. Sejam $A = \begin{bmatrix} 3 & 21 & 155 \\ 27 & 233 & 2 \\ 199 & 255 & 103 \end{bmatrix}$ e um escalar $k = 4$. A matriz respectiva da imagem A é:

Figura 2.16: Imagem da matriz A (a) A

Fonte: (Autorial, 2023)

- (a) Obtenha a multiplicação da matriz A pelo escalar k .

$$k \cdot A = C$$

$$4 \cdot \begin{bmatrix} 3 & 21 & 155 \\ 27 & 233 & 2 \\ 199 & 255 & 103 \end{bmatrix} = \begin{bmatrix} 12 & 84 & 620 \\ 108 & 932 & 8 \\ 796 & 1020 & 412 \end{bmatrix}.$$

- (b) Realize as técnicas de truncamento e normalização para que a imagem C seja exibida.

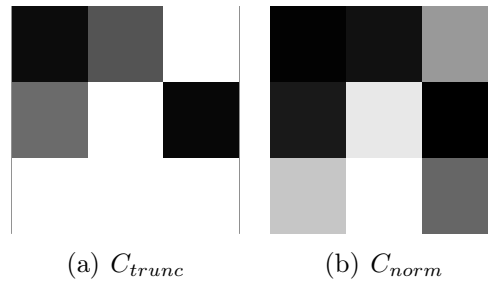
Para o truncamento, na matriz C os valores maiores que 255 mudamos para 255:

$$C_{trunc} = \begin{bmatrix} 12 & 84 & 255 \\ 108 & 255 & 8 \\ 255 & 255 & 255 \end{bmatrix}.$$

Para normalizar, iremos utilizar o intervalo $[c_{min}, c_{max}] = [8, 1020]$:

$$\begin{aligned} C_{norm} &= \frac{L-1}{c_{max} - c_{min}} \cdot (c_n - c_{min}), \\ &= \frac{255}{1020 - 8} \cdot \left(\begin{bmatrix} 12 - 8 & 84 - 8 & 620 - 8 \\ 108 - 8 & 932 - 8 & 8 - 8 \\ 796 - 8 & 1020 - 8 & 412 - 8 \end{bmatrix} \right), \\ &= \frac{255}{1012} \cdot \begin{bmatrix} 4 & 76 & 612 \\ 100 & 924 & 0 \\ 788 & 1012 & 404 \end{bmatrix} = \begin{bmatrix} 1 & 19 & 154 \\ 25 & 232 & 0 \\ 198 & 255 & 102 \end{bmatrix}. \end{aligned}$$

As imagens são:

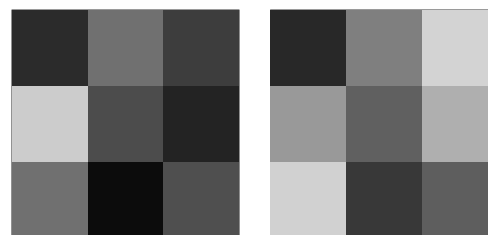
Figura 2.17: Imagem da matriz C truncada e normalizada

(a) C_{trunc} (b) C_{norm}

Fonte: (Autorial, 2023)

Agora, um exemplo para produto entre imagens:

Exemplo 2.18. Sejam $A = \begin{bmatrix} 17 & 111 & 61 \\ 203 & 77 & 36 \\ 112 & 14 & 80 \end{bmatrix}$ e $B = \begin{bmatrix} 42 & 127 & 211 \\ 154 & 96 & 175 \\ 209 & 55 & 87 \end{bmatrix}$. As matrizes respectivas da imagem são:

Figura 2.18: Imagem da matrizes A 

(a) A (b) B

Fonte: (Autorial, 2023)

(a) Obtenha a multiplicação da matriz A pela matriz B .

$$A \cdot B = C$$

$$\begin{aligned} & \begin{bmatrix} 17 & 111 & 61 \\ 203 & 77 & 36 \\ 112 & 14 & 80 \end{bmatrix} \cdot \begin{bmatrix} 42 & 127 & 211 \\ 154 & 96 & 175 \\ 209 & 55 & 87 \end{bmatrix} \\ &= \begin{bmatrix} 17 \cdot 42 + 111 \cdot 154 + 61 \cdot 209 & 17 \cdot 127 + 111 \cdot 96 + 61 \cdot 55 & 17 \cdot 211 + 111 \cdot 175 + 61 \cdot 87 \\ 203 \cdot 42 + 77 \cdot 154 + 36 \cdot 209 & 203 \cdot 127 + 77 \cdot 96 + 36 \cdot 55 & 203 \cdot 211 + 77 \cdot 175 + 36 \cdot 87 \\ 112 \cdot 42 + 14 \cdot 154 + 80 \cdot 209 & 112 \cdot 127 + 14 \cdot 96 + 80 \cdot 55 & 112 \cdot 211 + 14 \cdot 175 + 80 \cdot 87 \end{bmatrix} \\ &= \begin{bmatrix} 30557 & 16170 & 28319 \\ 27908 & 35153 & 59440 \\ 23580 & 19968 & 33042 \end{bmatrix}. \end{aligned}$$

- (b) Realize as técnicas de truncamento e normalização para que a imagem C seja exibida.

Para o truncamento, na matriz C os valores maiores que 255 mudamos para 255:

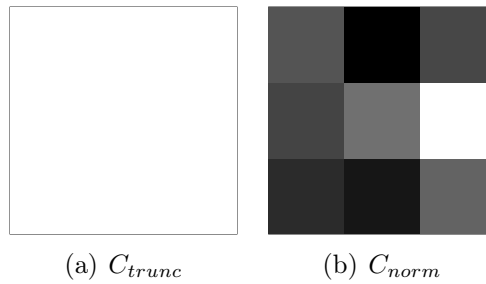
$$C_{trunc} = \begin{bmatrix} 255 & 255 & 255 \\ 255 & 255 & 255 \\ 255 & 255 & 255 \end{bmatrix}.$$

Para normalizar, iremos utilizar o intervalo $[c_{min}, c_{max}] = [16170, 59440]$:

$$\begin{aligned} C_{norm} &= \frac{L-1}{c_{max} - c_{min}} \cdot (c_n - c_{min}), \\ &= \frac{255}{59440 - 16170} \cdot \left(\begin{bmatrix} 30557 - 16170 & 16170 - 16170 & 28319 - 16170 \\ 27908 - 16170 & 35153 - 16170 & 59440 - 16170 \\ 23580 - 16170 & 19968 - 16170 & 33042 - 16170 \end{bmatrix} \right), \\ &= \frac{255}{43270} \cdot \begin{bmatrix} 14387 & 0 & 12149 \\ 11738 & 18983 & 43270 \\ 7410 & 3798 & 16872 \end{bmatrix} = \begin{bmatrix} 85 & 0 & 72 \\ 69 & 112 & 255 \\ 44 & 22 & 99 \end{bmatrix}. \end{aligned}$$

As imagens:

Figura 2.19: Imagem da matriz C truncada e normalizada



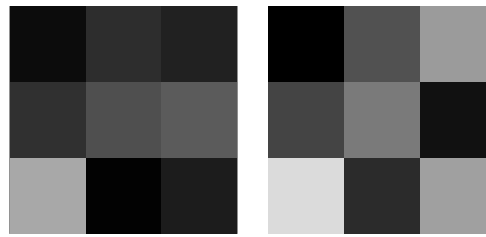
Fonte: (Autoral, 2023)

Observe que devido aos valores dos pixels (entradas) da imagem serem muito altos, após a multiplicação das duas figuras, a resultante no truncamento fica totalmente na cor branca, pois todos os valores ultrapassam de 255. No entanto, quando normalizamos conseguimos observar mais cores em tons cinza mais escuros e somente um pixel na cor branca.

Por fim, iremos exemplificar a operação de multiplicação elemento a elemento da seção [2.2.4.2](#) abaixo:

Exemplo 2.19. Sejam $A = \begin{bmatrix} 12 & 45 & 33 \\ 48 & 78 & 93 \\ 169 & 1 & 27 \end{bmatrix}$ e $B = \begin{bmatrix} 0 & 81 & 155 \\ 68 & 123 & 19 \\ 220 & 44 & 161 \end{bmatrix}$. As imagens

respectivas são:

Figura 2.20: Imagem da matrizes A (a) A (b) B

Fonte: (Autoral, 2023)

- (a) Obtenha a multiplicação elemento por elemento da matriz A pela matriz B .

$$\begin{aligned}
 A \odot B &= C, \\
 \begin{bmatrix} 12 & 45 & 33 \\ 48 & 78 & 93 \\ 169 & 1 & 27 \end{bmatrix} \odot \begin{bmatrix} 0 & 81 & 155 \\ 68 & 123 & 19 \\ 220 & 44 & 161 \end{bmatrix} &= \begin{bmatrix} 12 \cdot 0 & 45 \cdot 81 & 33 \cdot 155 \\ 48 \cdot 68 & 78 \cdot 123 & 93 \cdot 19 \\ 169 \cdot 220 & 1 \cdot 44 & 27 \cdot 161 \end{bmatrix}, \\
 &= \begin{bmatrix} 0 & 3.645 & 5115 \\ 3264 & 9594 & 1767 \\ 37180 & 44 & 4347 \end{bmatrix}.
 \end{aligned}$$

- (b) Realize as técnicas de truncamento e normalização para que a imagem C seja exibida.

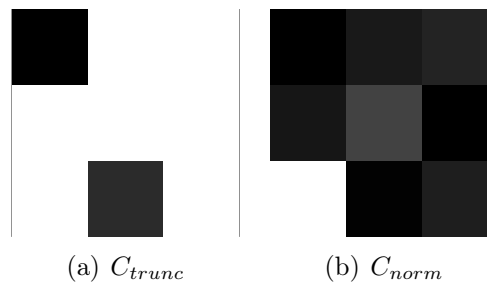
Para o truncamento, na matriz C os valores maiores que 255 mudamos para 255:

$$C_{trunc} = \begin{bmatrix} 0 & 255 & 255 \\ 255 & 255 & 255 \\ 255 & 44 & 255 \end{bmatrix}.$$

Para normalizar, iremos utilizar o intervalo $[c_{min}, c_{max}] = [0, 37180]$:

$$\begin{aligned}
 C_{norm} &= \frac{L-1}{c_{max} - c_{min}} \cdot (c_n - c_{min}), \\
 &= \frac{255}{37180 - 0} \cdot \left(\begin{bmatrix} 0 - 0 & 3645 - 0 & 5115 - 0 \\ 3264 - 0 & 9594 - 0 & 1767 - 0 \\ 37180 - 0 & 44 - 0 & 4347 - 0 \end{bmatrix} \right), \\
 &= \frac{255}{37180} \cdot \begin{bmatrix} 0 & 3645 & 5115 \\ 3264 & 9594 & 1767 \\ 37180 & 44 & 4347 \end{bmatrix} = \begin{bmatrix} 0 & 25 & 35 \\ 22 & 66 & 12 \\ 255 & 1 & 30 \end{bmatrix}.
 \end{aligned}$$

As imagens são:

Figura 2.21: Imagem da matriz C truncada e normalizada

Fonte: (Autorial, 2023)

2.4 Tipos de matrizes

2.4.1 Transposta

A transposição de uma matriz é uma operação que consiste em trocar as linhas de uma matriz por suas respectivas colunas, ou seja, dada uma matriz A com dimensões $m \times n$, sua transposta A^t é uma matriz com dimensões $n \times m$ tal que o elemento na linha i e coluna j de A^t é o elemento na linha j e coluna i de A .

Formalizando conforme (BOLDRINI et al., 1980):

Definição 2.20. Dada uma matriz $A = [a_{ij}]_{m \times n}$, podemos obter uma outra matriz $A^t = [b_{ij}]_{n \times m}$, cujas linhas são colunas de A , isto é, $b_{ij} = a_{ji}$. A^t é denominada transposta de A .

Exemplo 2.21.

$$\text{Se } A = \begin{bmatrix} 5 & 3 & -1 \\ 2 & -4 & 0 \end{bmatrix}_{2 \times 3}, \text{ então } A^t = \begin{bmatrix} 5 & 2 \\ 3 & -4 \\ -1 & 0 \end{bmatrix}_{3 \times 2}.$$

Definição 2.22. Uma matriz é simétrica se ela é igual à sua transposta, isto é, se $A = A^t$.

As principais propriedades estão demonstradas abaixo:

Propriedades 2.23.

i) $(A^t)^t = A$. Isto é, a transposta da transposta de uma matriz é ela mesma.

Demonstração.

Conforme a definição de transposta, a matriz $(A^t)^t$ é a transposta de A^t . Logo, para mostrar que $(A^t)^t = A$, precisamos mostrar que $(a_{ij}^t)^t = a_{ij}$ para todo i e j .

Temos que:

$$([a_{ij}^t])^t = [a_{ji}]^t = [a_{ij}],$$

onde a primeira igualdade é a definição de $(A^t)^t$, a segunda igualdade é a definição de transposição, e a terceira igualdade é a definição de A' .

Portanto, $([a_{ij}]^t)^t = [a_{ij}]$ para todo i e j , o que significa que $(A^t)^t = A$.

□

ii) $(A + B)^t = A^t + B^t$. Em palavras, a transposta de uma soma é igual à soma das transpostas.

Demonstração.

Seja A e B matrizes com as mesmas dimensões, assim a soma $A + B$ é igual a uma matriz $C = [c_{ij}] = [a_{ij}] + [b_{ij}]$ e $(A + B)^t = C^t$, sendo C^t a matriz transposta de C , então:

$$\begin{aligned} (A + B)^t &= C^t, \\ &= [c_{ji}], \\ &= [a_{ji}] + [b_{ji}], \\ &= [a_{ij}]^t + [b_{ij}]^t, \\ &= A^t + B^t, \end{aligned}$$

isto pois cada coluna de C^t é igual à respectiva linha de C e cada linha de C^t é igual à respectiva coluna de C .

Portanto, $(A + B)^t = A^t + B^t$.

□

iii) $(kA)^t = kA^t$, onde k é qualquer escalar.

Demonstração.

Suponha que tenhamos uma matriz A com elementos a_{ij} e uma matriz B definida como

$$B = kA = k[a_{ij}].$$

A transposta de B , denotada por B^t , é uma matriz em que os elementos da linha j são iguais aos elementos da coluna j de B e os elementos da coluna i são iguais aos elementos da linha i de B , ou seja,

$$B^t = k[a_{ji}].$$

Enquanto a transposta de A , denotada por $A^t = [a_{ji}]$. Então, temos

$$B^t = k[a_{ji}] = kA^t.$$

Portanto, a transposta do produto de k com A é igual ao produto de k com a transposta de A , ou seja, $(kA)^t = kA^t$.

□

iv) $(AB)^t = B^t A^t$ (Observe a ordem!)

Demonstração.

Sejam a_{ij} e b_{ij} os elementos da matriz A e B , respectivamente, e sejam c_{ij} os elementos da matriz AB . O elemento c_{ij} é dado por:

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}.$$

Podemos observar que o elemento correspondente na matriz $(AB)^t$ é:

$$(AB)^t = \sum_{k=1}^n a_{jk} b_{ki}.$$

Agora, vamos encontrar o elemento correspondente na matriz $B^t A^t$. O elemento B^t é igual a b_{ij} e o elemento A^t é igual a a_{ji} . Logo, o elemento correspondente na matriz $B^t A^t$ é dado por:

$$B^t A^t = \sum_{k=1}^n b_{jk} a_{ki}.$$

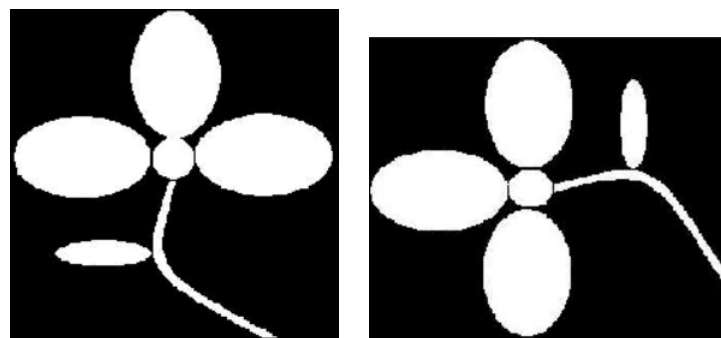
Como a transposição troca linhas por colunas, temos $(a_{ik} b_{kj})^t = b_{jk} a_{ki}$, ou seja, $(AB)^t = B^t A^t$.

Portanto, concluímos que $(AB)^t = B^t A^t$.

□

Em processamento de imagens temos como exemplo a Figura [2.22](#).

Figura 2.22: Transição de imagens



(a)

(b)

Fonte: (SILVA, 2014)

Note que cada linha de pixels da Figura 2.22 (a) é transformada numa coluna de pixels da Figura 2.22 (b).

2.4.2 Identidade

A matriz identidade é uma matriz quadrada que possui o número 1 na diagonal principal e zeros em todas as outras posições. A matriz identidade é geralmente denotada por I ou I_n , onde n é o número de linhas e colunas da matriz.

Definição 2.24. (BOLDRINI et al., 1980) A matriz identidade quadrada é aquela que $a_{ii} = 1$ e $a_{ij} = 0$, para $i \neq j$.

Por exemplo, a matriz identidade 3×3 é dada por:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

e a matriz identidade 4×4 é dada por:

$$I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

A matriz identidade tem algumas propriedades interessantes. É uma das mais importantes é:

Propriedades 2.25. Se A é uma matriz $m \times n$, então

$$AI = IA = A.$$

Demonstração.

Seja $A = [a_{ij}]$ uma matriz $m \times n$ e I a matriz identidade $n \times n$. O produto AI é dado por:

$$AI = \sum_{k=1}^n a_{ik} b_{kj},$$

em que b_{kj} é o elemento da linha k e coluna j da matriz identidade I , isto é, $b_{kj} = 1$ se $k = j$ e $b_{kj} = 0$ se $k \neq j$.

Então:

$$AI = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ij}b_{jj} + \cdots + a_{in}b_{nj} = a_{ij} + 0 + \cdots + 0 + \cdots + 0 = a_{ij}.$$

Logo, $AI = a_{ij}$ para todo i e j , o que implica que $AI = A$.

Já o produto IA é uma matriz $m \times n$, é dado por:

$$IA = \sum_{k=1}^n b_{ik} a_{kj}.$$

Como a matriz I é a matriz identidade, temos novamente que $b_{ik} = 1$ se $i = k$ e $b_{ik} = 0$ se $i \neq k$.

Assim:

$$IA = b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{ii}a_{ij} + \cdots + b_{in}a_{nj} = 0 + 0 + \cdots + a_{ij} + \cdots + 0 = a_{ij}.$$

Logo, $IA = a_{ij}$ para todo i e j , o que implica que $IA = A$.

Portanto, $AI = IA = A$ para qualquer matriz A .

□

A matriz identidade é importante em muitas áreas da matemática e é usada em vários cálculos e definições. Por exemplo, em álgebra linear, a matriz identidade é usada para definir o conceito de inversa de uma matriz que vamos apresentar na próxima seção. Em geral, a matriz identidade é uma ferramenta importante em muitas áreas da matemática e é um dos conceitos fundamentais em álgebra linear.

2.4.3 Inversa

Definição 2.26. (CALLIOLI; DOMINGUES, 1972) Uma matriz A de ordem n se diz *inversível* se, e somente se, existe uma matriz B , também de ordem n , de modo que:

$$AB = BA = I_n.$$

Esta matriz B , caso exista, é única e chama-se inversa de A , indica-se por A^{-1} .

A existência e a unicidade da matriz inversa são importantes porque ela permite resolver sistemas de equações lineares e realizar outras operações em matrizes.

Teorema 2.27. *Se B e C são ambas inversas da matriz A , então $B=C$.*

Demonstração.

Para mostrar que se B e C são ambas inversas da matriz A , então $B = C$, podemos utilizar as propriedades de inversa de uma matriz. Sabemos que B e C são inversas de A , logo

$$AB = BA = I$$

e

$$AC = CA = I.$$

Vamos começar provando que B é igual a C . Multiplicando a esquerda por B na segunda equação acima, temos $BAC = BCA = B$. Como $BA = I$, podemos substituir na primeira equação acima para obter:

$$BAC = IC = B \implies C = B.$$

□

Algumas propriedades importantes da matriz inversa incluem:

Propriedades 2.28.

i) Se A e B são invertíveis, então AB é invertível e $(AB)^{-1} = B^{-1}A^{-1}$.

Demonstração.

Primeiro, vamos mostrar que AB é invertível. Para isso, vamos mostrar que a matriz AB tem inversa. Sabemos que A e B são invertíveis, ou seja, existem matrizes A^{-1} e B^{-1} tais que $AA^{-1} = A^{-1}A = I$ e $BB^{-1} = B^{-1}B = I$.

Vamos multiplicar AB pela matriz $B^{-1}A^{-1}$:

$$(AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = AA^{-1} = I$$

e

$$(B^{-1}A^{-1})(AB) = B^{-1}(AA^{-1})B = B^{-1}B = I.$$

Assim, temos que $B^{-1}A^{-1}$ é uma matriz inversa de AB . Logo, AB é invertível.

Agora, vamos achar a matriz inversa de AB . E para isso, vamos usar a definição de matriz inversa e multiplicar $B^{-1}A^{-1}$ pela matriz AB :

$$(AB)(B^{-1}A^{-1}) = I.$$

Podemos multiplicar ambos os lados dessa equação por $(AB)^{-1}$ para obter:

$$\begin{aligned} (AB)^{-1}(AB)(B^{-1}A^{-1}) &= (AB)^{-1}I, \\ \implies B^{-1}A^{-1} &= (AB)^{-1}. \end{aligned}$$

Logo, a matriz inversa de AB é $B^{-1}A^{-1}$.

Por fim, vamos mostrar que a matriz encontrada é igual a $B^{-1}A^{-1}$ multiplicando $B^{-1}A^{-1}$ pela matriz AB :

$$\begin{aligned} (B^{-1}A^{-1})(AB) &= B^{-1}(AA^{-1})B = B^{-1}B = I, \\ \implies (B^{-1}A^{-1})(AB) &= I. \end{aligned}$$

Portanto, temos que a matriz inversa de AB é igual a $B^{-1}A^{-1}$ e se A e B são invertíveis, então AB é invertível e $(AB)^{-1} = B^{-1}A^{-1}$. \square

ii) Se A é invertível, então $(A^{-1})^{-1} = A$.

Demonstração.

Sabemos que se A é invertível, então existe uma matriz A^{-1} tal que $AA^{-1} = A^{-1}A = I$, onde I é a matriz identidade.

Calculando a inversa em ambos os lados, temos

$$\begin{aligned}(AA^{-1})^{-1} &= I^{-1}, \\ (A^{-1})^{-1}A^{-1} &= I.\end{aligned}$$

Multiplicando ambos os lados por A , obtemos

$$\begin{aligned}(A^{-1})^{-1}A^{-1}A &= IA, \\ (A^{-1})^{-1} &= A.\end{aligned}$$

Portanto, concluímos que $(A^{-1})^{-1} = A$ quando A é invertível. \square

iii) Se A é invertível, então A' é invertível e $(A')^{-1} = (A^{-1})'$.

Demonstração.

Para mostrar esta propriedade, primeiro vamos provar que A^t é invertível. Suponha que A é invertível. Então, por definição, existe uma matriz A^{-1} tal que $AA^{-1} = A^{-1}A = I$.

Para mostrar que A^t é invertível, devemos encontrar uma matriz X tal que $A^tX = XA^t = I$. Observe que $(A^{-1})^t$ é uma matriz X que satisfaz a propriedade, pois:

$$A'(A^{-1})^t = (A^{-1}A)^t = I' = I$$

e

$$(A^{-1})^tA^t = (AA^{-1})^t = I^t = I.$$

Então, A' é invertível.

Como A' é invertível então, multiplicando ambos os lados da equação $A'(A^{-1})' = I$ por $(A')^{-1}$ segue que $(A^{-1})' = (A')^{-1}$.

\square

A matriz inversa é uma ferramenta fundamental em álgebra linear e em várias outras áreas da matemática e é uma das principais maneiras de resolver sistemas de equações lineares e de entender a geometria de transformações lineares.

No capítulo [3](#), apresentamos uma aplicação importante de multiplicação, transposição e inversão de matrizes, por meio da técnica conhecida como Decomposição por Valores Singulares (DVS). Abordamos o conceito da DVS e os principais conceitos de Espaços Vetoriais necessários para compreender essa técnica.

Capítulo 3

Decomposição de Valores Singulares

Os espaços vetoriais são uma das estruturas mais importantes da matemática, e são amplamente utilizados em diversas áreas da ciência e tecnologia. Eles são definidos como um conjunto de vetores que satisfazem algumas propriedades algébricas, como a existência de uma operação de soma e uma operação de multiplicação por escalar.

Na área de processamento de imagens, os espaços vetoriais são utilizados em diversas técnicas, como a Decomposição por Valores Singulares (DVS). Esse método que consiste em decompor matrizes e é utilizada para compressão e reconstrução de imagens digitais.

As aplicações do estudo dos espaços vetoriais em processamento de imagens são apenas exemplos de sua importância e versatilidade. Vamos a seguir mostrar as definições, conceitos e demonstrações sobre esse tema seguindo (ANTON; RORRES, 2001) e depois apresentar uma aplicação na DVS.

3.1 Definição

Definição 3.1. Um conjunto não vazio V é um espaço vetorial sobre \mathbb{R} se, e somente se, satisfaz as seguintes condições:

- I - Existe uma operação de adição em V , denotada por $(u, v) \rightarrow u + v$, que possui as seguintes propriedades:
 - a) Comutatividade: $u + v = v + u$, para todo $u, v \in V$;
 - b) Associatividade: $u + (v + w) = (u + v) + w$, para todo $u, v, w \in V$;
 - c) Existência de elemento neutro: existe um elemento $0 \in V$ tal que $u + 0 = u$, para todo $u \in V$;
 - d) Existência de elemento oposto: para todo $u \in V$ existe um elemento $-u \in V$ tal que $u + (-u) = 0$.
- II. Existe uma operação de multiplicação em $\mathbb{R} \times V$, denotada por $(\alpha, u) \rightarrow \alpha u$, que possui as seguintes propriedades:

- a) Associatividade: $(\alpha\beta)u = \alpha(\beta u)$, para todo $\alpha, \beta \in \mathbb{R}$ e para todo $u \in V$;
- b) Distributividade em relação à adição em \mathbb{R} : $(\alpha + \beta)u = \alpha u + \beta u$, para todo $\alpha, \beta \in \mathbb{R}$ e para todo $u \in V$;
- c) Distributividade em relação à adição em V : $\alpha(u + v) = \alpha u + \alpha v$, para todo $\alpha \in \mathbb{R}$ e para todo $u, v \in V$;
- d) Existência de elemento neutro: $1v = v$, para todo $v \in V$.

3.2 Subespaços vetoriais

Definição 3.2. Um subespaço vetorial de V é um subconjunto não vazio $W \subseteq V$ que satisfaz as seguintes condições:

- a) $0 \in W$;
- b) $\forall u, v \in W, u + v \in W$;
- c) $\forall \alpha \in \mathbb{R}$ e $\forall u \in W, \alpha u \in W$.

Proposição 3.3. *Se W é um subespaço vetorial de V , então W é um espaço vetorial sobre o corpo \mathbb{R} .*

Demonstração.

Todas as propriedades da definição de espaço vetorial já são satisfeitas por V , que é um espaço vetorial. Além disso, como W é um subespaço de V , ele é fechado em relação às operações de adição e multiplicação por escalar que já existem em V . Portanto, todas as propriedades também serão satisfeitas por W , o que mostra que W é um espaço vetorial sobre \mathbb{R} .

□

3.3 Independência Linear e Bases

Definição 3.4. Um conjunto $L = \{u_1, u_2, \dots, u_n\} \subset V$ é dito linearmente independente (L.I.) se, e somente se, a única combinação linear que satisfaz a igualdade abaixo

$$\alpha_1 u_1 + \alpha_2 u_2 + \dots + \alpha_n u_n = 0,$$

para $\alpha_i \in \mathbb{R}$, é aquela em que todos os escalares α_i são iguais a zero, ou seja, quando $\alpha_1 = \alpha_2 = \dots = \alpha_n = 0$.

Definição 3.5. Dizemos que $L = \{u_1, u_2, \dots, u_n\} \subset V$ é linearmente dependente (L.D.) se, e somente se, L não é L.I., ou seja, é possível que escreva

$$\alpha_1 u_1 + \dots + \alpha_n u_n = 0$$

sem que os escalares α_i sejam todos iguais a zero.

Definição 3.6. Seja V um espaço vetorial qualquer e $S = \{v_1, v_2, \dots, v_n\}$ for um conjunto finito de vetores em V , dizemos que S é uma base de V se valerem as duas condições a seguir.

- (a) S é linearmente independente.
- (b) S gera V .

Teorema 3.7. Se $S = \{v_1, v_2, \dots, v_n\}$ for uma base de um espaço vetorial V , então cada vetor em V pode ser expresso de maneira única na forma $v = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$.

Demonstração.

Precisamos primeiro mostrar que a expressão de um vetor v como uma combinação linear de S é sempre possível, e que essa expressão é única. Como S é uma base de V , então cada vetor em V pode ser escrito como uma combinação linear dos vetores em S . Podemos escrever v como $v = c_1 v_1 + c_2 v_2 + \dots + c_n v_n$ para $c_1, c_2, \dots, c_n \in \mathbb{R}$. Para mostrar que essa expressão é única vamos supor que existam escalares $d_1, d_2, \dots, d_n \in \mathbb{R}$ tais que $v = d_1 v_1 + d_2 v_2 + \dots + d_n v_n$. Então, temos:

$$c_1 v_1 + c_2 v_2 + \dots + c_n v_n = d_1 v_1 + d_2 v_2 + \dots + d_n v_n.$$

Reescrevendo, temos:

$$\begin{aligned} c_1 v_1 + c_2 v_2 + \dots + c_n v_n - d_1 v_1 - d_2 v_2 - \dots - d_n v_n &= 0 \\ (c_1 - d_1)v_1 + (c_2 - d_2)v_2 + \dots + (c_n - d_n)v_n &= 0. \end{aligned}$$

Mas S é uma base, então os vetores v_1, v_2, \dots, v_n são linearmente independente e todos os coeficientes serão iguais a zero. Logo, temos que $c_1 - d_1 = c_2 - d_2 = \dots = c_n - d_n = 0$, o que implica que $c_1 = d_1, c_2 = d_2, \dots, c_n = d_n$.

Portanto, a expressão de v como uma combinação linear de S é única. \square

3.4 Produto interno

Definição 3.8. Um produto interno num espaço vetorial real V é uma função que associa um número real $\langle u, v \rangle$ a cada par de vetores em V de tal maneira que os seguintes axiomas são satisfeitos por quaisquer vetores u, v e w de V e qualquer escalar α .

- (a) $\langle u, v \rangle = \langle v, u \rangle$;
- (b) $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$;
- (c) $\langle \alpha u, v \rangle = \alpha \langle u, v \rangle$;
- (d) $\langle v, v \rangle \geq 0$ e $\langle v, v \rangle = 0$ se, e só se, $v = 0$.

Um espaço vetorial real com um produto interno é chamado de espaço com produto interno real.

Podemos definir o produto interno de dois vetores u e v em \mathbb{R}^n da seguinte maneira:

$$\langle u, v \rangle = u \cdot v = u_1v_1 + u_2v_2 + \dots + u_nv_n.$$

Definição 3.9. Se V for um espaço com produto interno real, então a norma (ou comprimento) de um vetor v em V é definida por

$$\|v\| = \sqrt{\langle v, v \rangle}$$

e a distância entre dois vetores é denotada por $d(u, v)$ e definida por

$$d(u, v) = \|u - v\| = \sqrt{\langle u - v, u - v \rangle}.$$

Dizemos que um vetor de norma 1 é um vetor unitário.

3.5 Autovetores e Autovalores

Definição 3.10. Se A for uma matriz $n \times n$, então um vetor não nulo x em \mathbb{R}^n é denominado autovetor de A (ou do operador matricial T_A) se Ax for múltiplo escalar de x , isto é,

$$Ax = \lambda x,$$

com algum escalar λ . O escalar λ é denominado autovalor de A (ou de T_A), e dizemos que x é um autovetor associado a λ .

Teorema 3.11. Se A for uma matriz $n \times n$, então λ é um autovalor de A se, e só se, λ satisfaz a equação

$$\det(\lambda I - A) = 0.$$

Essa equação é a equação característica de A .

Demonstração.

Se λ é um autovalor de A , então existe um vetor não nulo v tal que $Av = \lambda v$. Isso pode ser reescrito como $Av - \lambda v = 0$, ou seja, $(A - \lambda I)v = 0$. Como v é não nulo, a

matriz $A - \lambda I$ não é invertível, o que implica que seu determinante é zero. Logo, temos $\det(A - \lambda I) = 0$, que é a equação característica de A .

Por outro lado, se $\det(\lambda I - A) = 0$, então a matriz $\lambda I - A$ não é invertível, o que implica que existe um vetor não nulo v tal que $(\lambda I - A)v = 0$. Isso pode ser reescrito como $Av = \lambda v$, o que mostra que λ é um autovalor de A .

Portanto, que λ é um autovalor de A se e somente se $\det(\lambda I - A) = 0$.

□

3.6 Diagonalização

Teorema 3.12. *Se A for uma matriz $n \times n$, são equivalentes as afirmações seguintes.*

(a) A é diagonalizável.

(b) A tem n autovetores linearmente independentes.

Demonstração.

(a) \Rightarrow (b):

Se A é diagonalizável, então existe uma matriz diagonal D e uma matriz invertível P tais que $A = PDP^{-1}$, onde D é uma matriz diagonal cujos elementos na diagonal são os autovalores de A e P é a matriz cujas colunas são formadas pelos autovetores de A .

Sejam v_1, v_2, \dots, v_n os autovetores de A associados aos autovalores $\lambda_1, \lambda_2, \dots, \lambda_n$, respectivamente. Então, seja $P = [v_1 \ v_2 \ \dots \ v_n]$, temos que:

$$AP = PD,$$

o que implica que

$$Av_i = \lambda_i v_i, \quad i = 1, 2, \dots, n.$$

Isso significa que A tem n autovetores linearmente independentes, já que a matriz V é invertível. Portanto, a afirmação (a) implica na afirmação (b).

(b) \Rightarrow (a):

Suponha que A tenha n autovetores linearmente independentes. Então, seja $V = [v_1 \ v_2 \ \dots \ v_n]$ uma matriz cujas colunas são os autovetores de A . Como V é invertível, podemos escrever:

$$AV = VD,$$

onde D é uma matriz diagonal cujos elementos na diagonal são os autovalores de A . Isso implica que:

$$A = VDV^{-1}.$$

Portanto, A é diagonalizável. Assim, a afirmação (b) implica na afirmação (a).

Concluimos que as afirmações (a) e (b) são equivalentes.

□

3.7 Decomposição por Valores Singulares (DVS)

Definição 3.13. Se A for uma matriz $m \times n$ e $\lambda_1, \lambda_2, \dots, \lambda_n$ os autovalores de $A^t A$, então os números

$$\sigma_1 = \sqrt{\lambda_1}, \sigma_2 = \sqrt{\lambda_2}, \dots, \sigma_n = \sqrt{\lambda_n}$$

são denominados valores singulares de A .

Teorema 3.14. (ANTON; RORRES, 2001) Se A for uma matriz $m \times n$, então A pode ser expressa como

$$A = U\Sigma V^t$$

em que U é uma matriz ortogonal $m \times m$, V é uma matriz ortogonal $n \times n$ e Σ é uma matriz $m \times n$ cujas entradas diagonais são os valores singulares não nulos de A e cujas demais entradas são nulas.

A demonstração pode ser encontrada no Capítulo 9 da referência citada acima.

Exemplo 3.15. Segue um exemplo de uma decomposição DVS citada em (PESCO; BOR-TOLOSSI, 2013):

Demonstração.

Seja a matriz

$$A = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}.$$

Vamos calcular primeiro a matriz $A^t A$ para obter uma matriz quadrada e depois encontrar os autovalores e autovetores dela. Temos a matriz:

$$A^t A = \begin{bmatrix} -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}.$$

Agora, vamos calcular os autovalores que são as raízes da equação $\det(A^t A - \lambda I) = 0$, ou seja, são as raízes de

$$\det \begin{bmatrix} 1 - \lambda & -1 & 0 \\ -1 & 2 - \lambda & -1 \\ 0 & -1 & 1 - \lambda \end{bmatrix} = 0.$$

Resolvendo a equação, temos:

$$(1 - \lambda)^2(2 - \lambda) + (-2)(1 - \lambda) = -3\lambda + 4\lambda^2 - \lambda^3 = 0.$$

As raízes então são os autovalores: $\lambda_1 = 3$, $\lambda_2 = 1$ e $\lambda_3 = 0$. Vamos calcular agora os valores singulares de A , pois os valores singulares são as raízes quadradas dos autovalores

de $A^t A$, então eles são:

$$\sigma_1 = \sqrt{\lambda_1} = \sqrt{3}, \quad \sigma_2 = \sqrt{\lambda_2} = 1 \quad \text{e} \quad \sigma_3 = \sqrt{\lambda_3} = 0.$$

Calculando os autovetores associados aos autovalores de $A^t A$ para construir a matriz ortogonal V . Os autovetores associados a λ_1 com $(A^t A - \lambda_1 I) \cdot X = 0$, sendo $X = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$,

temos:

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & -1 & -1 \\ 0 & -1 & -2 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Resolvendo o sistema tem-se que $y = -2x$ e $z = x$, sendo $x, y, z \in \mathbb{R}$. Tomando $x = 1$, obtemos $v_1 = (1, -2, 1)$.

Para λ_2 com $(A^t A - \lambda_2 I) \cdot X = 0$:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 1 & -1 \\ 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Resolvendo o sistema tem-se que $x = -z$ e $y = 0$, sendo $x, y, z \in \mathbb{R}$. Tomando $x = -1$, obtemos $v_2 = (-1, 0, 1)$.

Para λ_3 com $(A^t A - \lambda_3 I) \cdot X = 0$:

$$\begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Resolvendo o sistema tem-se que $x = y = z$, sendo $x, y, z \in \mathbb{R}$. Tomando $x = 1$, obtemos $v_3 = (1, 1, 1)$. Normalizando os autovetores para serem utilizados na matriz ortogonal V , temos que:

$$\begin{aligned} v_1 &= \frac{1}{\sqrt{1 + (-2)^2 + 1}} \cdot (1, -2, 1) = \left(\frac{1}{\sqrt{6}}, -\frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}} \right) = \left(\frac{\sqrt{6}}{6}, -\frac{\sqrt{6}}{3}, \frac{\sqrt{6}}{6} \right), \\ v_2 &= \frac{1}{\sqrt{1 + 0 + 1}} \cdot (-1, 0, 1) = \left(-\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}} \right) = \left(-\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2} \right) \text{ e} \\ v_3 &= \frac{1}{\sqrt{1 + 1 + 1}} \cdot (1, 1, 1) = \left(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}} \right) = \left(\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3} \right). \end{aligned}$$

Desse modo, a matriz ortogonal V é:

$$\begin{bmatrix} \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{6}}{3} & 0 & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{3} \end{bmatrix}.$$

e ela diagonaliza $A^t A$.

Pelo Teorema [3.14](#), podemos observar que Σ é:

$$\begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \end{bmatrix} = \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

A matriz Σ é dessa forma, pois é uma matriz diagonal, na qual, seus elementos são os valores singulares não nulos de A .

Por fim, vamos calcular U na decomposição em valores singulares de A . Essa matriz é composta pelos autovetores normalizados associados aos autovalores não nulos de AA^t :

$$u_1 = \frac{1}{\sigma_1} \cdot Av_1 = \frac{1}{\sqrt{3}} \cdot \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \\ \frac{\sqrt{6}}{6} \end{bmatrix} = \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \end{bmatrix},$$

$$u_2 = \frac{1}{\sigma_2} \cdot Av_2 = 1 \cdot \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -\frac{\sqrt{2}}{2} \\ 0 \\ \frac{\sqrt{2}}{2} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{bmatrix}.$$

Segue que,

$$U = \begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}.$$

Portanto, podemos mostrar a matriz original A como produto de U, Σ e V^t pela decom-

posição:

$$\begin{aligned}
 A &= \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} = U\Sigma V^t \\
 &= \begin{bmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{6}}{6} & -\frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{6}}{6} & 0 & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{2} & \frac{\sqrt{3}}{3} \end{bmatrix}^t.
 \end{aligned}$$

□

3.7.1 DVS aplicado em imagens

O Teorema 3.14 é uma ferramenta poderosa para redução de dados em diversas áreas da ciência, e a aplicação em processamento de imagens é apenas um exemplo. No caso específico de transporte de imagens de satélite para a Terra, a utilização da decomposição por valores singulares (DVS) permite a transmissão de uma quantidade significativamente menor de dados, economizando recursos e garantindo a qualidade da informação visual.

De acordo com (PESCO; BORTOLOSSI, 2013) iremos tomar como exemplo uma imagem em tons de cinza de tamanho 1000 por 1000 pixels. Inicialmente, o satélite precisaria transmitir um total de 1 milhão de números, um para cada pixel da imagem. No entanto, somente os primeiros elementos da matriz Σ da decomposição por valores singulares são relevantes, enquanto os outros são considerados "irrelevantes".

Assim, é suficiente enviar apenas as 20 primeiras colunas de U e V , bem como os 20 primeiros valores singulares σ_{ii} (totalizando apenas $20 \cdot 1000 + 20 \cdot 1000 + 20 = 40020$ números a serem enviados). Ao receber esses dados no laboratório da Terra, podemos calcular a matriz:

$$\sigma_{1,1}u_1v_1^t + \sigma_{2,2}u_2v_2^t + \dots + \sigma_{20,20}u_{20}v_{20}^t,$$

que nos fornece uma aproximação da imagem original equivalente a 4,002%.

Para ilustrar a eficácia desse método, considere a fotografia do matemático Christian Felix Klein (1849 - 1925), que tem $720 \times 524 = 377289$ pixels. Utilizando a DVS e enviando apenas os primeiros 20 valores singulares e colunas de U e V , é possível recuperar uma imagem com qualidade visualmente similar à original, como mostrado na Figura 3.1.

Figura 3.1: Fotografia de Christian Feliz Klein original



Fonte: (PESCO; BORTOLOSSI, 2013)

Podemos calcular as matrizes para a decomposição DVS da matriz correspondente a esta imagem:

$$\sigma_{1,1}u_1v_1^t + \sigma_{2,2}u_2v_2^t + \dots + \sigma_{r,r}u_rv_r^t,$$

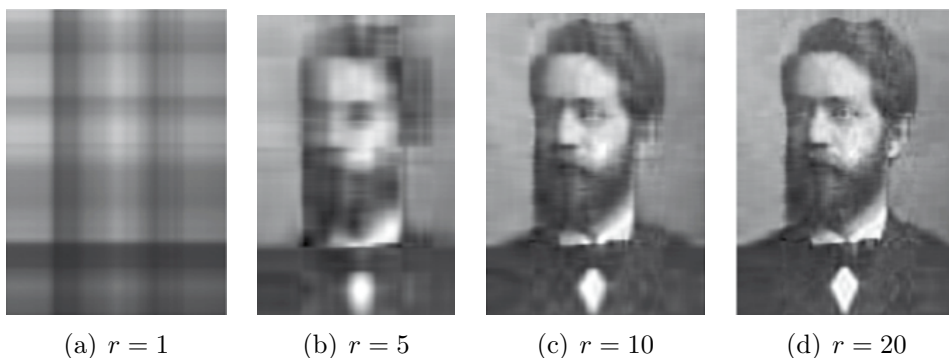
para $r = 1, 5, 10, 20$.

Observa-se na Figura 3.2 que, à medida que o valor de r aumenta, a imagem se aproxima cada vez mais da original. Quando $r = 20$ temos que

$$20 \cdot 720 + 20 \cdot 524 + 20 = 14400 + 10480 + 20 = 24900$$

que equivale a aproximadamente 6,6%. Vale ressaltar que a imagem original é restaurada quando $r = 524$, o que equivale ao número de linhas da matriz da imagem.

Figura 3.2: Imagem com aplicação da DVS



Fonte: (PESCO; BORTOLOSSI, 2013)

Portanto, a aplicação da DVS é extremamente útil no processamento de imagens, especialmente em casos em que a quantidade de dados a serem transmitidos é limitada ou quando se deseja economizar recursos de armazenamento e processamento.

Capítulo 4

Transformações lineares

Neste capítulo, aprofundamos nosso estudo nas transformações lineares, abordando especificamente os operadores de reflexão, rotação e translação. Realizamos uma introdução teórica sobre o tema, estabelecendo conexões entre essas transformações geométricas e seu uso no contexto do processamento digital de imagens. Exploramos a teoria, visando fornecer uma base para compreender a comunicação entre a matemática e a prática no processamento de imagens digitais.

4.1 Definição

Definição 4.1. Seja $T : V \rightarrow W$ uma função que mapeia um espaço vetorial V para um espaço vetorial W . A função T é uma transformação linear de V para W se satisfizer as duas condições abaixo para quaisquer vetores u e v em V e qualquer escalar k .

$$(i) \quad T(kv) = kT(v)$$

$$(ii) \quad T(u + v) = T(u) + T(v)$$

No caso de $V = W$, a transformação linear é denominada operador linear do espaço vetorial V .

Exemplo 4.2. Seja $V = \mathbb{R}^2$ e $W = \mathbb{R}^3$, mostre que $T : \mathbb{R}^2 \rightarrow \mathbb{R}^3$, tal que $T(x, y) = (2x, 0, x + y)$ é uma transformação linear.

Dados, $u, v \in \mathbb{R}^2$, sejam $u = (x_1, y_1)$ e $v = (x_2, y_2)$. Temos:

$$(i) \quad T(kv) = kT(v)$$

$$\begin{aligned} T(kv) &= T(k(x_2, y_2)) \\ &= T(kx_2, ky_2) \\ &= (2kx_2, 0, kx_2 + ky_2) \\ &= k(2x_2, 0, x_2 + y_2) \\ &= kT(v). \end{aligned}$$

Logo, a primeira condição é válida. Agora, temos:

$$(ii) \quad T(u + v) = T(u) + T(v)$$

$$\begin{aligned} T(u + v) &= T((x_1, y_1) + (x_2, y_2)) \\ &= T(x_1 + x_2, y_1 + y_2) \\ &= (2(x_1 + x_2), 0, (x_1 + x_2) + (y_1 + y_2)) \\ &= (2x_1, 0, x_1 + y_1) + (2x_2, 0, x_2 + y_2) \\ &= T(u) + T(v). \end{aligned}$$

e a segunda condição é válida. Portanto, T é uma transformação linear.

Teorema 4.3. (ANTON; RORRES, 2001) Se $T : V \rightarrow W$ for uma transformação linear, então

$$(a) \quad T(0) = 0.$$

$$(b) \quad T(u - v) = T(u) - T(v), \text{ quaisquer que sejam } u \text{ e } v \text{ em } V.$$

Demonstração.

Seja v um vetor qualquer em V . Pela Definição 4.1 (i) de transformação linear, temos:

$$T(0) = T(0v) = 0T(v) = 0.$$

Logo, (a) está demonstrada.

Agora, a parte (b) iremos usar a propriedade da Definição 4.1 (ii), então temos:

$$\begin{aligned} T(u - v) &= T(u + (-v)) \\ &= T(u + (-v)) \\ &= T(u) + T(-v) \\ &= T(u) + T((-1)v). \end{aligned}$$

Como consequência da condição (i) de 4.1, $T((-1)v) = (-1)T(v)$, então

$$\begin{aligned} T(u - v) &= T(u) + (-1)T(v) \\ &= T(u) - T(v). \end{aligned}$$

Logo, a parte (b) do teorema também está demonstrada. □

A seguir iremos mostrar com exemplos as transformações nula e identidade.

Exemplo 4.4. Transformação nula

Sejam V e W espaços vetoriais, a aplicação $T : V \rightarrow W$ tal que $T(v) = 0$, para qualquer vetor $v \in V$, é a transformação nula. Podemos observar que T é linear, pois

(i) Para qualquer escalar k e vetor v , temos

$$T(kv) = 0 = k0 = kT(v).$$

(ii) Para quaisquer vetores $u, v \in V$, temos

$$T(u + v) = 0 = 0 + 0 = T(u) + T(v).$$

Exemplo 4.5. Operador identidade

Seja V um espaço vetorial, a aplicação $I : V \rightarrow V$ definida por $I(v) = v$ é denominada operador identidade de V . Podemos observar que T é linear, pois

(i) Para qualquer escalar k e vetor v , temos

$$I(kv) = kv$$

e

$$kI(v) = kv,$$

igualando obtemos $I(kv) = kI(v)$.

(ii) Para quaisquer vetores $u, v \in V$, temos

$$I(u + v) = u + v = I(u) + I(v).$$

Teorema 4.6. *Se $V \rightarrow W$ for uma transformação linear, onde V um espaço vetorial de dimensão finita e $S = \{v_1, v_2, \dots, v_n\}$ é uma base de V , então a imagem de qualquer vetor $v \in V$ pode ser expressa como*

$$T(v) = c_1T(v_1) + c_2T(v_2) + \dots + c_nT(v_n),$$

tais que c_1, c_2, \dots, c_n são os coeficientes que expressam v como combinação linear dos vetores em S .

Demonstração.

Seja $T : V \rightarrow W$ uma transformação linear, onde V é um espaço vetorial de dimensão finita e $S = \{v_1, v_2, \dots, v_n\}$ é uma base de V .

Se v é expresso como combinação linear dos vetores em S , temos:

$$v = c_1v_1 + c_2v_2 + \dots + c_nv_n.$$

Agora, aplicando a transformação linear T em ambos os lados da equação:

$$T(v) = T(c_1v_1 + c_2v_2 + \dots + c_nv_n).$$

Pelas propriedades da Definição 4.1, podemos distribuir a transformação linear sobre a soma:

$$T(v) = c_1T(v_1) + c_2T(v_2) + \dots + c_nT(v_n).$$

Isso conclui a prova do teorema. □

4.1.1 Núcleo e Imagem

Definição 4.7. Seja $T : V \rightarrow W$ uma transformação linear. O conjunto dos vetores em V tais que $T(v) = 0$ é denominado núcleo de T e denotado por $Nuc(T)$ ou $ker(T)$.

Teorema 4.8. *Seja $T : V \rightarrow W$ uma transformação linear. O conjunto $Nuc(T)$, núcleo de T é um subespaço vetorial de V .*

Demonstração.

Queremos mostrar que $Nuc(T)$ é um subespaço de V . Vamos mostrar a seguir que são válidas as propriedades da Definição 3.2:

(i) Pelo Teorema 4.3 (a) sabemos que $T(0) = 0$, assim o vetor 0 elemento neutro de V está em $Nuc(T)$.

(ii) Seja $v_1, v_2 \in Nuc(T)$, tais que $T(v_1) = 0$ e $T(v_2) = 0$, e k um escalar qualquer. Observe que:

$$T(v_1 + v_2) = T(v_1) + T(v_2) = 0 + 0 = 0.$$

Logo, $v_1 + v_2$ também está no núcleo de T .

(iii) Seja $v \in Nuc(T)$ e k um escalar qualquer, então

$$T(kv) = kT(v) = k0 = 0.$$

Isso mostra que kv está no núcleo de T .

Portanto, o núcleo de T é um subespaço de V . □

Definição 4.9. Seja $T : V \rightarrow W$ uma transformação linear. A imagem é o conjunto de todos os vetores em $w \in W$ tais que exista pelo menos um vetor em V que vale $T(v) = w$. Podemos denotar a imagem por $Im(T)$.

Teorema 4.10. *Seja $T : V \rightarrow W$ uma transformação linear. O conjunto $Im(T)$, imagem da transformação T é um subespaço vetorial de W .*

Demonstração.

Queremos mostrar que a imagem de T , denotada por $Im(T)$, é um subespaço de W .

- (i) A imagem contém ao menos o vetor nulo, pois pelo Teorema 4.3 (a) $T(0) = 0$.
- (ii) Sejam $w_1, w_2 \in Im(T)$, então existem $v_1, v_2 \in V$ tais que $T(v_1) = w_1$ e $T(v_2) = w_2$.

Temos que:

$$w_1 + w_2 = T(v_1) + T(v_2) = T(v_1 + v_2).$$

Isso implica que $w_1 + w_2$ está na imagem de T .

- (iii) Seja $w \in Im(T)$ e k um escalar qualquer, então existe v em V tal que $T(v) = w$.

Assim,

$$kw = kT(v) = T(kv).$$

Isso mostra que $kw \in Im(T)$ e então a imagem de T é um subespaço de W .

□

4.1.2 Matrizes e Transformações Lineares

As transformações lineares podem ser descritas como multiplicação de matrizes, uma vez que toda matriz de dimensão $m \times n$ está associada a uma transformação linear $T : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Desse modo, uma transformação T aplicada ao vetor v pode ser representada como produto de uma matriz $M_{m \times n}$ pelo vetor coluna v , resultando em um vetor w tal que $w = Mv$.

Percebemos então que a transformação associa o vetor coluna v em \mathbb{R}^n ao vetor coluna w em \mathbb{R}^m , resultando na chamada transformação matricial ou operador matricial caso $m = n$. Denotaremos $T_M : \mathbb{R}^n \rightarrow \mathbb{R}^m$ e nesse caso, temos:

$$w = T_M(v)$$

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \vdots & \vdots & & \vdots \\ m_{m1} & m_{m2} & \dots & m_{mn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}.$$

Abaixo, exemplificamos a base canônica que será de grande ajuda nos exemplos e aplicações da próxima seção. É possível escrever todo elemento $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ em \mathbb{R}^2 como:

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = v_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

isto é, todo vetor do plano é uma combinação linear de $e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ e $e_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$. Analogamente, podemos fazer a base canônica para $\mathbb{R}^n \rightarrow \mathbb{R}^n$, porém a seguir estudamos as transformações lineares geométricas no \mathbb{R}^2 .

4.2 Transformações geométricas

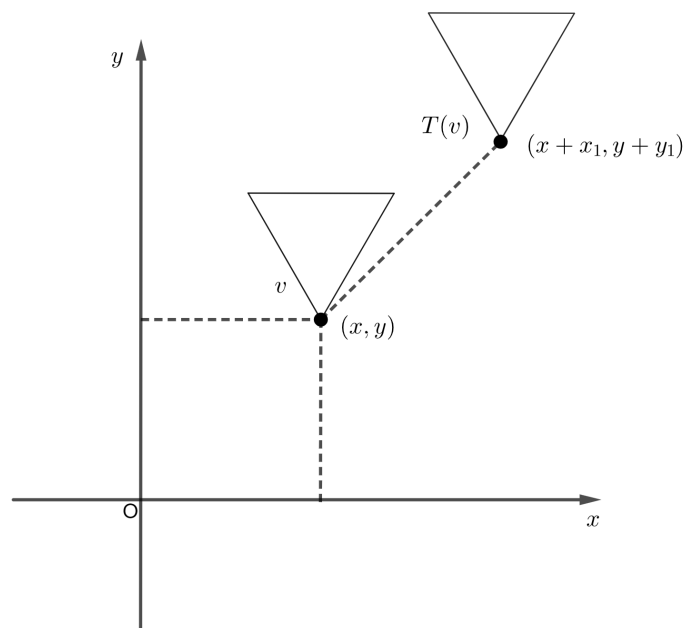
Apresentamos nesta seção uma abordagem baseada na geometria das transformações lineares, ilustrando-as com exemplos de transformações que ocorrem no plano (\mathbb{R}^2). Esses operadores são utilizados para modificar qualquer figura, seja criando novas ou apenas alterando sua posição. Neste aspecto, vamos aplicar as transformações geométricas apresentadas a seguir sobre imagens digitais, ou seja, vamos transladar, refletir e rotacionar imagens.

4.2.1 Translação

A translação é uma maneira de mover pontos em um espaço vetorial sem alterar a forma e o tamanho de qualquer figura. Um operador de translação é o operador T cuja ação em um ponto v é dada por $(x + x_1, y + y_1)$. A matriz de transformação de translação é:

$$T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}.$$

Figura 4.1: Transformação de Translação no plano



No entanto, é importante ressaltar que este operador não é linear e mesmo que a translação, seja uma transformação não linear, é comumente utilizada na manipulação de imagens digitais. Percebemos que a transformação de translação não é linear, pois

$$T(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \neq 0$$

o que contradiz a parte (a) do Teorema [4.3](#).

Exemplo 4.11. Aplique a translação $T(x, y) = (x + x_1, y + y_2)$, sendo $x_1 = 4$ e $y_1 = 2$ nos vértices $(1, 2)$, $(2, 1)$ e $(3, 2)$ do triângulo.

Desse modo, podemos fazer a translação de cada ponto citado:

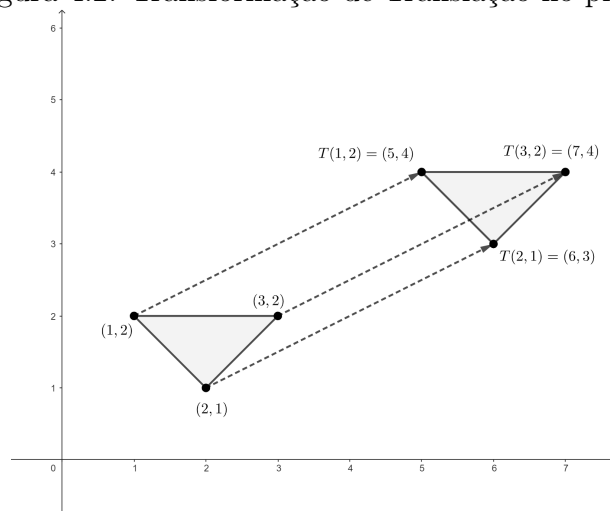
$$T(1, 2) = (1 + x_1, 2 + y_1) = (1 + 4, 2 + 2) = (5, 4);$$

$$T(2, 1) = (2 + x_1, 2 + y_1) = (2 + 4, 1 + 2) = (6, 3);$$

$$T(3, 2) = (3 + x_1, 2 + y_1) = (3 + 4, 2 + 2) = (7, 4).$$

E podemos representar graficamente da seguinte forma:

Figura 4.2: Transformação de Translação no plano



Fonte: (Autorial, 2024)

4.2.1.1 Translação em imagens

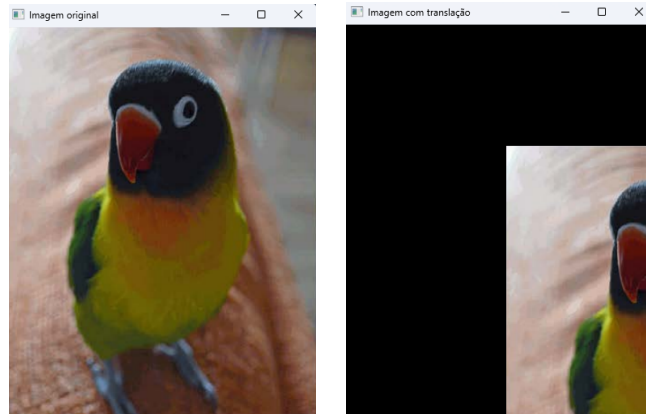
Como já falamos, a translação é uma mudança no posicionamento de uma imagem ou objeto. Conforme ([GARCÍA et al., 2015](#)), a matriz de transformação da translação por um vetor (x_1, y_1) é:

$$\begin{bmatrix} 1 & 0 & x_1 \\ 0 & 1 & y_1 \end{bmatrix}.$$

Para transladarmos uma imagem, precisamos multiplicar as coordenadas de cada pixel de uma imagem pela matriz de transformação, porém como esse processo pode ser muito

trabalhoso feito de forma manual, a biblioteca *OpenCV*¹ nos ajuda a ver como ocorre o processo. Além disso, (GARCÍA et al., 2015) estudam a transformação através dessa biblioteca e iremos mostrar a aplicação na Figura 4.3 (b), na qual, o deslocamento ocorreu com $x_1 = 200$ e $y_1 = 150$ pixels.

Figura 4.3: Translação em imagens



(a) Imagem original (b) Imagem transladada

Fonte: (GARCÍA et al., 2015), adaptado.

O código utilizado para aplicar a transformação segue abaixo:

Figura 4.4: Código utilizado para a Figura 4.3

```
#include "opencv2/opencv.hpp"
using namespace cv;
int main(int argc, char** argv)
{
    //Lendo a imagem original
    Mat src;
    src = imread("original.png");

    //Aplicando a transformação de translação
    Mat dst;
    Mat M = (Mat_<double>(2, 3) << 1, 0, 200, 0, 1, 150);
    warpAffine(src, dst, M, src.size());

    //Exibindo as imagens
    namedWindow("Imagem original", WINDOW_AUTOSIZE);
    imshow("Imagem original", src);

    namedWindow("Imagem com translação", WINDOW_AUTOSIZE);
    imshow("Imagem com translação", dst);

    waitKey();
    return 0;
}
```

Fonte: (GARCÍA et al., 2015), adaptado.

¹A Open Source Computer Vision Library (OpenCV), é uma biblioteca de software utilizada para visão computacional e aprendizado de máquina.

Em primeiro lugar, a biblioteca *OpenCV* é chamada para iniciar o código e permitir a exibição das imagens. Em seguida, a parte principal do código é declarar a variável *src*, do tipo *Mat* (matriz), para armazenar a imagem original. A matriz *M* é criada para representar a matriz de transformação e a função *warpAffine* da biblioteca é utilizada para aplicar a transformação *M* na matriz da imagem original *src*, armazenando o resultado. Posteriormente, são criadas duas janelas para exibir a imagem original e a imagem transladada.

Observe que a matriz de transformação está utilizando coordenadas homogêneas, pois conforme explicam os autores, essas coordenadas permitem a representação estável de transformações, a fim de realizar multiplicações com outras transformações, mantendo-as padronizadas (mesmo formato). Por isso, é adicionada uma terceira coordenada nos objetos geométricos em duas dimensões (2D) e será exemplificado dessa forma nas outras transformações em imagens também.

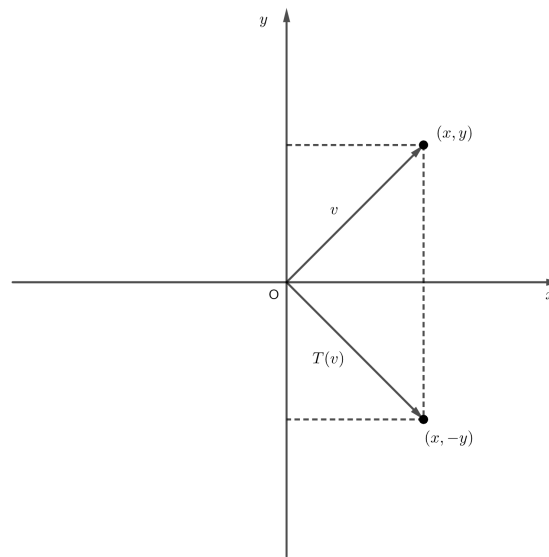
4.2.2 Reflexão

A reflexão ocorre quando queremos espelhar uma figura em relação a um eixo, a transformação linear de reflexão é dada por uma matriz com alguns elementos na diagonal principal com sinal negativo. A seguir mostraremos algumas formas de reflexão para $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$:

(a) Reflexão em torno do eixo x

A transformação $T(x, y) = (x, -y)$ é utilizada para reflexão do vetor $v = \begin{bmatrix} x \\ y \end{bmatrix}$ em torno do eixo x , que é representada matricialmente como:

$$T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

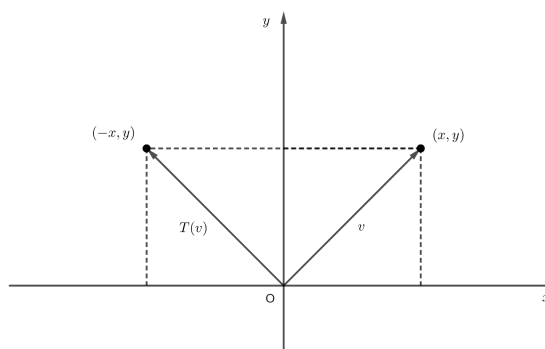
Figura 4.5: Reflexão em torno do eixo x 

Fonte: (SILVA, 2013), adaptado.

(b) Reflexão em torno do eixo y

A reflexão em torno do eixo y é expressa como $T(x, y) = (-x, y)$ e por matrizes:

$$T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Figura 4.6: Reflexão em torno do eixo y 

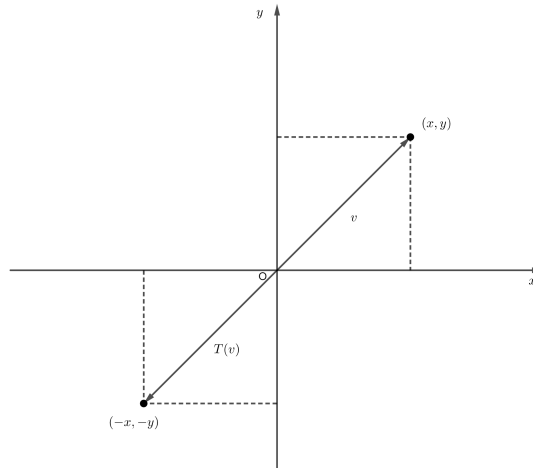
Fonte: (SILVA, 2013), adaptado.

(c) Reflexão na origem

A reflexão na origem é dada por $T(x, y) = (-x, -y)$, matricialmente temos:

$$T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Figura 4.7: Reflexão em torno da origem



Fonte: (SILVA, 2013), adaptado.

Exemplo 4.12. Seguindo o Exemplo 4.11, vamos refletir os vértices $(1, 2)$, $(2, 1)$ e $(3, 2)$ do triângulo, em torno dos eixos x, y e na origem.

Aplicando a transformação em cada vértice para cada eixo, obtemos:

(a) Eixo x :

$$T(1, 2) = (x, -y) = (1, -2);$$

$$T(2, 1) = (x, -y) = (2, -1);$$

$$T(3, 2) = (x, -y) = (3, -2).$$

(b) Eixo y :

$$T(1, 2) = (-x, y) = (-1, 2);$$

$$T(2, 1) = (-x, y) = (-2, 1);$$

$$T(3, 2) = (-x, y) = (-3, 2).$$

(c) Origem:

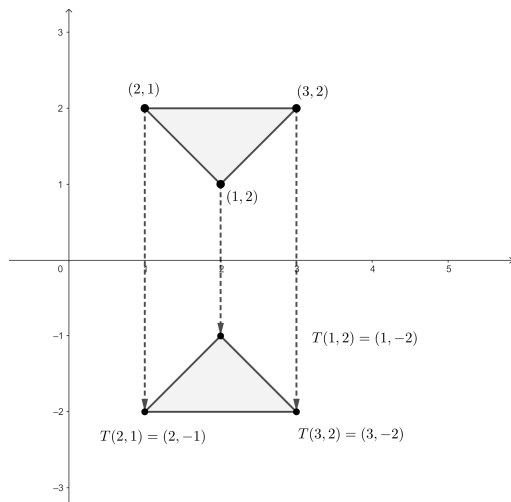
$$T(1, 2) = (-x, -y) = (-1, -2);$$

$$T(2, 1) = (-x, -y) = (-2, -1);$$

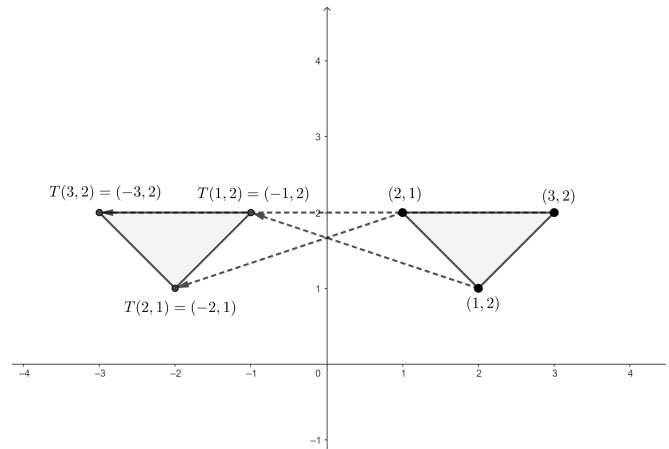
$$T(3, 2) = (-x, -y) = (-3, -2).$$

As figuras refletidas são:

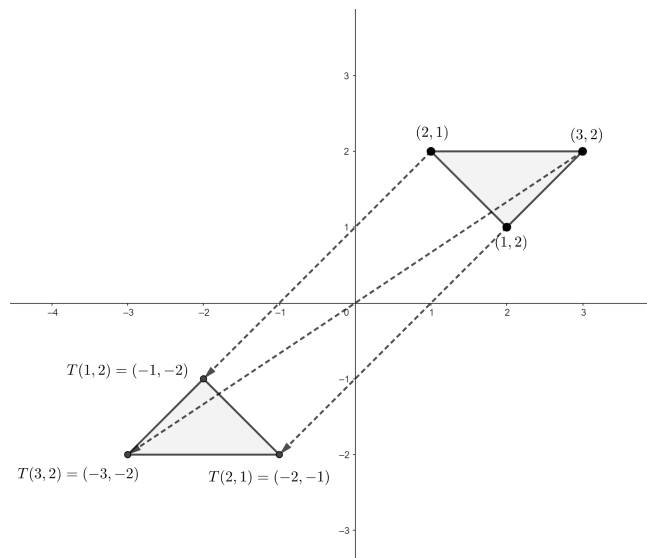
Figura 4.8: Reflexão do Exemplo 4.12



(a) Em torno eixo x



(b) Em torno eixo y



(c) Em torno da origem

Fonte: (Autoral, 2024)

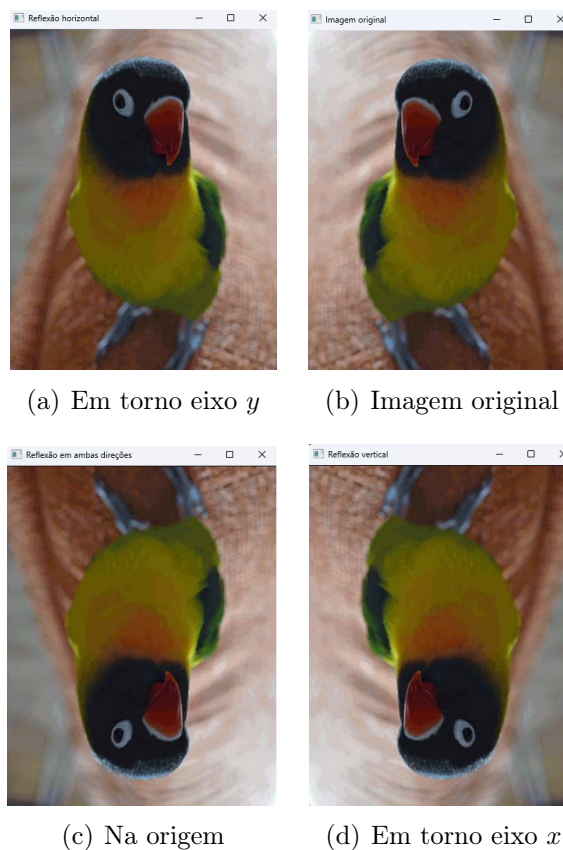
4.2.2.1 Reflexão em imagens

Neste contexto, precisamos verificar como modificar uma imagem com a transformação de reflexão. De acordo com (GARCÍA et al., 2015) é preciso aplicar uma translação na última coluna da matriz de transformação, pois a reflexão é feita sobre x e y . As matrizes de reflexão pelo *OpenCV* para o eixo x , y e a origem, respectivamente:

$$T_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & t_y \end{bmatrix}, \quad T_y = \begin{bmatrix} -1 & 0 & t_x \\ 0 & 1 & 0 \end{bmatrix} \quad \text{e} \quad T_O = \begin{bmatrix} -1 & 0 & t_x \\ 0 & -1 & t_y \end{bmatrix}.$$

O parâmetro t_x é o número de colunas da imagem e t_y o número de linhas da imagem. Exemplificamos a transformação de reflexão nos eixos por (GARCÍA et al., 2015) e o código utilizado logo abaixo.

Figura 4.9: Reflexão em imagens



Fonte: (GARCÍA et al., 2015), adaptado.

Figura 4.10: Código utilizado para a Figura 4.9

```

#include "opencv2/opencv.hpp"

using namespace cv;

int main(int argc, char** argv) {

    //Lendo a imagem original

    Mat src;
    src = imread("original.png");

    //Aplicando a transformação de reflexão

    Mat dsth, dstv, dst;
    Mat Mh = (Mat_<double>(2, 3) << -1, 0, src.cols, 0, 1, 0);
    Mat Mv = (Mat_<double>(2, 3) << 1, 0, 0, 0, -1, src.rows);
    Mat M = (Mat_<double>(2, 3) << -1, 0, src.cols, 0, -1, src.rows);
    warpAffine(src, dsth, Mh, src.size());
    warpAffine(src, dstv, Mv, src.size());
    warpAffine(src, dst, M, src.size());

    //Exibindo as imagens

    namedWindow("Imagem original", WINDOW_AUTOSIZE);
    imshow("Imagem original", src);

    namedWindow("Reflexão horizontal", WINDOW_AUTOSIZE);
    imshow("Reflexão horizontal", dsth);

    namedWindow("Reflexão vertical", WINDOW_AUTOSIZE);
    imshow("Reflexão vertical", dstv);

    namedWindow("Reflexão em ambas direções", WINDOW_AUTOSIZE);
    imshow("Reflexão em ambas direções", dst);

    waitKey();
    return 0;
}

```

Fonte: (GARCÍA et al., 2015), adaptado.

O código é parecido para a transformação de translação, mas determina três matrizes M_v , M_h e M , que são respectivamente, para a reflexão nos eixos x (vertical), y (horizontal) e origem (ambas direções). Após isso aplica a função *warpAffine* em cada uma dessas matrizes à matriz da imagem original, guardando a imagem resultante para exibir em seguida nas novas janelas criadas.

4.2.3 Rotação

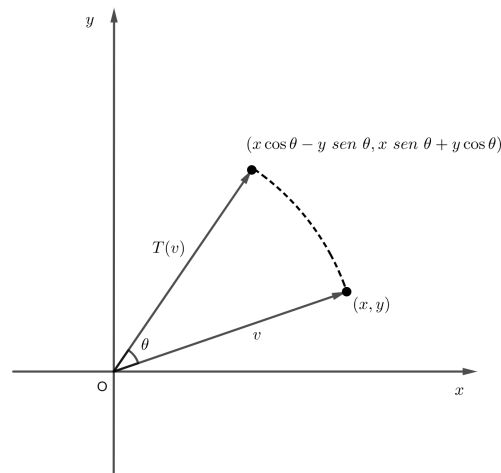
Apresentaremos agora a transformação de rotação no plano. A rotação permite mover um vetor em torno da origem de um ângulo θ através da multiplicação da matriz de transformação de rotação pelas coordenadas do vetor que queremos mover.

Definição 4.13. (SILVA, 2013) Uma transformação de rotação no plano em torno da origem de um ângulo θ é uma aplicação

$$R : \mathbb{R}^2 \rightarrow \mathbb{R}^2,$$

tal que

$$R \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Figura 4.11: Rotação em torno do ângulo θ 

Fonte: (SILVA, 2013), adaptado.

Exemplo 4.14. Aplique a rotação nos vértices $(1, 2)$, $(2, 1)$ e $(3, 2)$ do triângulo em torno da origem de um ângulo de $\theta = 60^\circ$.

Realizando a rotação de um ponto (x, y) em torno da origem pelo ângulo $\theta = 60^\circ$, vamos utilizar as equações:

$$R(x, y) = (x \cos(\theta) - y \sin(\theta), x \sin(\theta) + y \cos(\theta)).$$

Então para cada ponto, temos:

(a) Para $(1, 2)$:

$$\begin{aligned} R(1, 2) &= (1 \cos(60^\circ) - 2 \sin(60^\circ), 1 \sin(60^\circ) + 2 \cos(60^\circ)) \\ &= \left(1 \cdot \frac{1}{2} - 2 \cdot \frac{\sqrt{3}}{2}, 1 \cdot \frac{\sqrt{3}}{2} + 2 \cdot \frac{1}{2} \right) \\ &= \left(\frac{1}{2} - \sqrt{3}, \frac{\sqrt{3}}{2} + 1 \right) \\ &\approx (0.13, 2.23). \end{aligned}$$

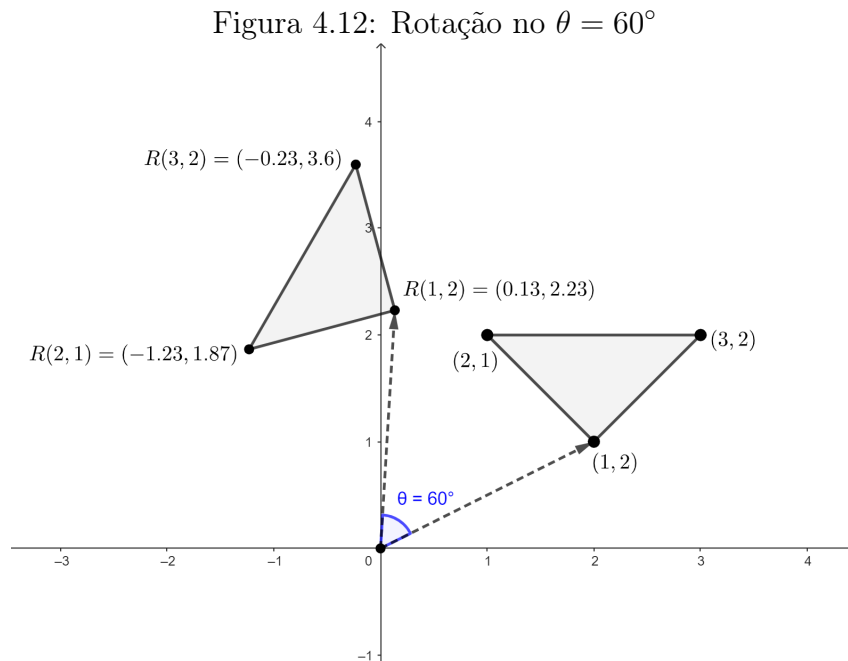
(b) Para (2, 1):

$$\begin{aligned} R(2, 1) &= (2 \cos(60^\circ) - 1 \operatorname{sen}(60^\circ), 2 \operatorname{sen}(60^\circ) + 1 \cos(60^\circ)) \\ &= \left(2 \cdot \frac{1}{2} - 1 \cdot \frac{\sqrt{3}}{2}, 2 \cdot \frac{\sqrt{3}}{2} + 1 \cdot \frac{1}{2} \right) \\ &= \left(1 - \frac{\sqrt{3}}{2}, \sqrt{3} + \frac{1}{2} \right) \\ &\approx (-1.23, 1.87). \end{aligned}$$

(c) Para (3, 2):

$$\begin{aligned} R(3, 2) &= (3 \cos(60^\circ) - 2 \operatorname{sen}(60^\circ), 3 \operatorname{sen}(60^\circ) + 2 \cos(60^\circ)) \\ &= \left(3 \cdot \frac{1}{2} - 2 \cdot \frac{\sqrt{3}}{2}, 3 \cdot \frac{\sqrt{3}}{2} + 2 \cdot \frac{1}{2} \right) \\ &= \left(\frac{3}{2} - \sqrt{3}, \frac{3\sqrt{3}}{2} + 1 \right) \\ &\approx (-0.23, 3.6). \end{aligned}$$

Representar a figura rotacionada:



Fonte: (Autorial, 2024)

4.2.3.1 Rotação em imagens

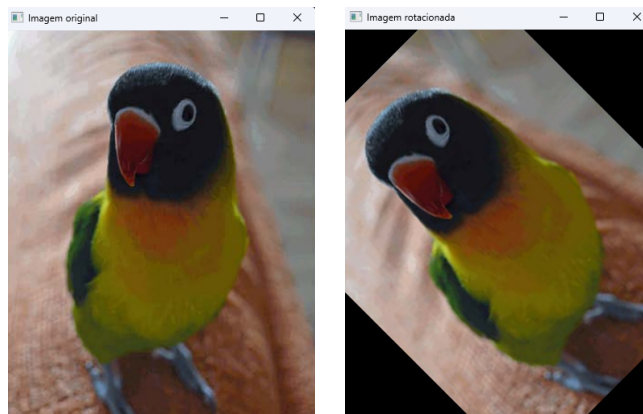
Segundo (GARCÍA et al., 2015), a transformação de rotação ocorre em torno de um ângulo θ (sentido anti-horário). Por *OpenCV*, a matriz de transformação de rotação:

$$\begin{bmatrix} k \cos \theta & k \sin \theta & (1 - k \cos \theta)x - (k \sin \theta)y \\ -k \sin \theta & k \cos \theta & (k \sin \theta)x + (1 - k \cos \theta)y \end{bmatrix}.$$

As coordenadas da rotação são x e y , e k é um escalar que é utilizado para controlar a proporção da transformação. Lembrando que se $k = 1$ não há alteração na dimensão e a rotação ocorre de forma usual.

A matriz de transformação da rotação mostrada acima é decorrente de algumas transformações geométricas. Com a biblioteca *OpenCv* é criada a transformação de rotação M para realizar uma rotação em torno da origem da imagem com ângulo de 45° com escalar $k = 1$. Seguindo o mesmo processo dos códigos das transformações anteriores, a transformação é aplicada à matriz da imagem original pela função *warpAffine*. Abaixo estão a imagem rotacionada:

Figura 4.13: Rotação em imagens



(a) Imagem original

(b) Em torno de $\theta = 45^\circ$

Fonte: (GARCÍA et al., 2015), adaptado.

e o código utilizado:

Figura 4.14: Código utilizado para a Figura 4.13

```
#include "opencv2/opencv.hpp"

using namespace cv;

int main(int argc, char** argv) {

    //Lendo a imagem original

    Mat src;
    src = imread("original.png");

    //Aplicando a transformação de rotação

    Mat dst;
    Mat M = getRotationMatrix2D(Point2f(src.cols / 2, src.rows / 2), 45, 1);
    warpAffine(src, dst, M, src.size());

    //Exibindo as imagens

    namedWindow("Imagem original", WINDOW_AUTOSIZE);
    imshow("Imagem original", src);

    namedWindow("Imagem rotacionada", WINDOW_AUTOSIZE);
    imshow("Imagem rotacionada", dst);

    waitKey();
    return 0;
}
```

Fonte: (GARCÍA et al., 2015), adaptado.

Assim, concluímos este capítulo, ressaltando a existência de outros tipos de transformações geométricas na Álgebra Linear que são capazes de alterar imagens digitais. No entanto, como se trata de uma introdução ao tema do processamento digital de imagens, foram abordadas apenas três delas, selecionadas pela autora, pois fazem parte das principais transformações vistas nas aulas de Matemática tanto no ensino básico quanto superior.

Conclusão

Com base neste trabalho, é possível chegar à conclusão de que a Matemática é uma área extremamente vasta e que a Álgebra Linear apresenta múltiplas abordagens possíveis. Além de fornecer um trabalho que possa ser consultado futuramente, este estudo contribui para que observamos como o processamento digital de imagens está relacionado a cada operação matricial. Os exemplos visuais podem tornar o estudo mais interativo e facilitar o entendimento dos conceitos abordados pela Álgebra Linear.

Durante o desenvolvimento desta monografia, analisamos também a Decomposição por Valores Singulares e suas aplicações no processamento de imagens, demonstrando assim a relevância e utilidade desta técnica. Ao utilizar tal abordagem, nosso trabalho ganha maior rigor técnico. A prática é muito útil na computação gráfica para transportar imagens de forma mais rápida e acessível comprimindo os pixels da imagem.

Na última parte, utilizamos um pouco da computação para representar as transformações geométricas, pois facilitou o estudo não precisando realizar tantos cálculos para imagens e podendo visualizar imagens digitais coloridas. Com isso, exploramos mais a fundo a relação entre processamento digital de imagens e Álgebra Linear, ampliando nossos conhecimentos e demonstrando a importância deste aprendizado em minha formação como licencianda em Matemática. Este trabalho é apenas uma pequena parte das possibilidades que a interseção entre essas duas áreas oferece.

Referências Bibliográficas

- ANDRADE, F. dos S. Imagens digitais e matrizes: Uma aplicação no ensino médio. *Universidade Estadual do Sudoeste da Bahia, Mestrado Profissional em Matemática em Rede Nacional – PROFMAT*, 2020.
- ANTON, H.; RORRES, C. *Álgebra linear com aplicações*. [S.l.]: Bookman Porto Alegre, 2001. v. 8.
- AZEVEDO, E.; CONCI, A.; VASCONCELOS, C. *Computação gráfica: Teoria e prática: geração de imagens*. [S.l.]: Elsevier Brasil, 2018. v. 1.
- BALDINO, R. R.; SILVA, R. d. Introdução ao processamento de imagens ou aplicação da álgebra linear. *Revista de Matemática e Estatística*, v. 19, p. 123–144, 2001.
- BOLDRINI, J. L. et al. *Álgebra linear*. [S.l.]: Harper & Row, 1980.
- CALLIOLI, C. A.; DOMINGUES, H. H. *Álgebra linear. Coleção Schaum. São Paulo: McGraw-Hill do Brasil*, 1972.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999.
- FURTADO, D. P.; FARIA, C. O. de; SASAKI, D. Aplicando a decomposição em valores singulares no processamento de imagens. *Cadernos do IME-Série Matemática*, n. 15, p. 86–103, 2020.
- GARCÍA, G. B. et al. *Learning image processing with OpenCV*. [S.l.]: Birmingham: Packt Publishing Ltd, 2015.
- GOMES, J.; VELHO, L. *Computação gráfica: Imagem*. [S.l.]: IMPA, 1994.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento digital de imagem*. Pearson, ISBN-10: 8576054019, v. 10, p. 11–27, 2010.
- JOY, D. Image processing through linear algebra. *Department of Mathematics, College of Engineering Trivandrum, Infokara*, v. 9, p. 742–753, 2020.
- MÜLLER, D. N.; DARONCO, E. L. Operações aritméticas em imagens. *Universidade Federal do Rio Grande do Sul, Porto Alegre*, 2000.
- PESCO, D. U.; BORTOLOSSI, H. J. Imagens digitais e matrizes. *Revista Gazeta Matemática*, n. 169, p. 44–48, 2013.
- SAÚDE, A. V. *Computação Gráfica e Processamento de Imagens*. [S.l.]: Editora e Distribuidora Educacional S.A., 2019.

SILVA, J. D. O. d. Processamento de imagens digitais e o ensino de matrizes. *Dissertação (Mestrado Profissional em Matemática) - Programa de Pós-Graduação Matemática em Rede Nacional, Universidade Federal do Oeste do Pará, Santarém, 2014.*

SILVA, R. F. d. Transformações geométricas no plano e no espaço. *Mestrado Profissional em Matemática em Rede Nacional PROFMAT/CCEN-UEPB, Universidade Federal da Paraíba, 2013.*

SIQUEIRA, I. A. C. Matrizes e uma introdução às imagens digitais. *Universidade Federal de Mato Grosso, Mestrado Profissional em Matemática: PROFMAT/SBM, 2014.*

Apêndice A

Códigos utilizando OpenCV

A.0.1 Translação

```
#include "opencv2/opencv.hpp"

using namespace cv;

int main( int argc, char** argv) {

//Lendo a imagem original

Mat src;
src = imread("original.png");

//Aplicando a transformação de translação

Mat dst;
Mat M = (Mat_<double>(2,3) « 1, 0, 200, 0, 1, 150);
warpAffine(src, dst, M, src.size());

//Exibindo as imagens

namedWindow( "Imagem original", WINDOW_AUTOSIZE);
imshow("Imagem original", src);

namedWindow("Imagem com translação", WINDOW_AUTOSIZE);
imshow("Imagem com translação", dst);

waitKey();
return 0;
```

```
}
```

A.0.2 Reflexão

```
#include "opencv2/opencv.hpp"

using namespace cv;

int main( int argc, char** argv) {

//Lendo a imagem original

Mat src;
src = imread("original.png");

//Aplicando a transformação de reflexão

Mat dsth, dstv, dst;
Mat Mh = (Mat_<double>(2,3) « -1, 0, src.cols, 0, 1, 0);
Mat Mv = (Mat_<double>(2,3) « 1, 0, 0, 0, -1, src.rows);
Mat M = (Mat_<double>(2,3) « -1, 0, src.cols, 0, -1, src.rows);
warpAffine(src, dsth, Mh, src.size());
warpAffine(src, dstv, Mv, src.size());
warpAffine(src, dst, M, src.size());

//Exibindo as imagens

namedWindow("Imagem original", WINDOW_AUTOSIZE);
imshow("Imagem original", src);

namedWindow("Reflexão horizontal", WINDOW_AUTOSIZE);
imshow("Reflexão horizontal", dsth);

namedWindow("Reflexão vertical", WINDOW_AUTOSIZE);
imshow("Reflexão vertical", dstv);

namedWindow("Reflexão em ambas direções", WINDOW_AUTOSIZE);
imshow("Reflexão em ambas direções", dst);

waitKey();
```

```
return 0;
}
```

A.0.3 Rotação

```
#include "opencv2/opencv.hpp"

using namespace cv;

int main( int argc, char** argv) {

//Lendo a imagem original

Mat src;
src = imread("original.png");

//Aplicando a transformação de rotação

Mat dst;
Mat M = getRotationMatrix2D(src.cols / 2, src.rows / 2), 45, 1);
warpAffine(src, dst, M, src.size());

//Exibindo as imagens

namedWindow("Imagem original", WINDOW_AUTOSIZE);
imshow("Imagem original", src);

namedWindow("Imagem rotacionada", WINDOW_AUTOSIZE);
imshow("Imagem rotacionada", dst);

waitKey();
return 0;
}
```