

Universidade de São Paulo



## **Trabalho 1**

# Introdução à Representação de Imagens

Processamento e Análise de Imagens

**Aluno**

Thiago Oliveira dos Santos n° 13696220

São Carlos  
2024

# 1 Objetivos

O objetivo deste trabalho é fornecer uma introdução ao processamento de imagens digitais, com foco na compreensão da representação de uma imagem digital em um formato monocromático. Isso envolve entender como os pixels são organizados para formar a imagem e como diferentes intensidades de cinza são representadas numericamente. Além disso, busca-se desenvolver uma intuição sobre a resolução da escala de cinza, ou seja, entender a quantidade de tons de cinza distintos que podem ser representados em uma imagem digital e como essa resolução afeta a qualidade e a percepção visual da imagem.

## 2 Introdução

### 2.1 Representação de uma imagem

Uma imagem pode ser representada como uma função  $f(x, y)$ , onde cada par ordenado  $(x, y)$  representa um elemento de imagem (ou pixel), e o valor de  $f$  representa a intensidade da luz que incide sobre esse ponto.

Essa função  $f$  é discreta; ou seja,  $x, y \in \mathbb{N}$ , e os valores de  $f$  são normalizados de acordo com a quantidade de tons de cinza a serem representados.

### 2.2 Escala de cinza

Supondo que, antes de normalizar  $f$ , esta tenha um valor mínimo e um valor máximo  $L_{min}$  e  $L_{max}$ , respectivamente, e que se deseja representar a imagem com um número definido de tons monocromáticos, número esse que, por conveniência, escolho ser  $2^n$ .

Daí, para normalizar essa função, o leitor deve imaginar que os limites  $L_{min}$  e  $L_{max}$  são deslocados para os valores 0 e  $L = 2^n$ , onde o primeiro representa um pixel completamente preto, e o segundo representa um pixel completamente branco.

Os valores intermediários são dados pela Equação 1:

$$L_{int} = \left\lfloor \frac{f(x, y) - L_{min}}{L_{max} - L_{min}} \times (L - 1) \right\rfloor \quad (1)$$

onde  $\lfloor \cdot \rfloor$  denota a função de piso. Essa fórmula distribui os valores de intensidade original de forma uniforme dentro da escala de cinza normalizada, permitindo representar todos os tons monocromáticos disponíveis de maneira equitativa.

Assim, a imagem digital passa a ser representada como uma matriz de  $x$  linhas e  $y$  colunas, na qual cada elemento indica a intensidade quantizada do pixel, na escala de cinza escolhida.

### 2.3 Resolução

A resolução de uma imagem, ou então, o grau de detalhes discerníveis em uma imagem, depende fortemente da quantidade de pixels, e da quantidade de valores de níveis de cinza discretos disponíveis para o pixel assumir. Portanto, os principais fatores que determinam a resolução de uma imagem são: a quantidade de linhas  $x$ ; a quantidade de colunas  $y$ , e a escala de cinza  $2^n$ .

Vale ressaltar que, embora possa parecer intuitivo que uma maior quantidade de

níveis de cinza resulte em uma imagem com maior resolução, isso nem sempre é verdadeiro. A percepção subjetiva da qualidade da imagem desempenha um papel importante. Uma imagem com mais tons de cinza pode parecer mais suave e com menos artefatos visuais, mas isso não necessariamente significa que ela contenha mais detalhes ou informações úteis. Portanto, a escolha da resolução da escala de cinza deve ser feita considerando cuidadosamente as características específicas da imagem e os requisitos da aplicação.

## 3 Metodologia e Resultados

### 3.1 Materiais anteriores

Antes de abordar os métodos utilizados para a resolução dos problemas propostos no trabalho, gostaria de enfatizar que numa outra ocasião, já tive que iniciar um trabalho a respeito do uso de classes em Python, para a manipulação de imagens em mapa de escala de cinza, ou .pgm. Com isso em mente, comecei a fazer leves alterações para a generalização da classe, para que essa aceitasse a manipulação de imagens com outros tipos de extensão. Dito isso, segue em anexo um rascunho da minha biblioteca de manipulação de imagens em Python: ➡

O leitor deve ter em mente que a partir desse ponto no documento, qualquer referência a uma imagem digital, levará em consideração um arquivo .pgm.

### 3.2 Abertura e Visualização

Para as seguintes seções, vide o arquivo do trabalho: ➡

Para a visualização da imagem, utilizei a biblioteca Matplotlib. No arquivo da biblioteca, tem a definição da função de impressão de uma imagem em escala de cinza. Tive de especificar que a escala fosse em cinza, por, caso contrário, a impressão apareceria colorida.

Listing 1: Diferença entre as impressões

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg

def print_image(imagem):
    img = mpimg.imread(imagem._local)
    plt.imshow(img)
    plt.axis('off') # Para desativar os eixos
    plt.show()

def print_grayscale_image(imagem):
    img = mpimg.imread(imagem._local)

    # Exibir a imagem em tons de cinza
    plt.imshow(img, cmap='gray')
    plt.axis('off') # Para desativar os eixos
    plt.show()

img.print_grayscale_image(imagem)
img.print_image(imagem)
```

No Código 1, o leitor pode ver as duas funções para impressão da imagem, como discutido acima. As figuras 1 e 2 ilustram os resultados dessas funções.




Figura 1: Caso da impressão em escala de cinza



Figura 2: Caso da impressão em cor

### 3.3 Determinação das dimensões

Uma vantagem em trabalhar com arquivos .pgm, é que as dimensões da imagem são indicadas diretamente na leitura do arquivo. Portanto, para a impressão das dimensões, eu apenas inicializei o arquivo na classe Image, e fiz a impressão de suas propriedades de dimensões, que no arquivo eu indico como sendo "*dimension*". Essa imagem em específico que estou usando, mostrada na Figura 1, possui 480 linhas, e 640 colunas. Caso o leitor queira confirmar, aqui está o arquivo para referência: 

### 3.4 Valores de Máximo e Mínimo

Outra vantagem dos arquivos .pgm, é que a intensidade máxima presente na imagem, é também indicada diretamente na leitura do arquivo. Portanto, fiz a impressão diretamente da propriedade "*max*", e isso já foi o suficiente para a

resolução dessa parte do problema. Para a determinação do mínimo, foi usado um algoritmo simples, que acha o menor valor de uma matriz, como mostrado no Código 2

Listing 2: Algoritmo para o menor valor dentro da matriz de pixels

```
@property
def min(self):
    for linha in self.pixels:
        if min(linha) < minimo:
            minimo = min(linha)
    return minimo
```

Com isso, foram determinados os máximos e mínimos de intensidade da imagem escolhida, sendo esses respectivamente 208 e 0. A partir destes valores, é discutido na seção 4 qual a melhor resolução para a imagem escolhida.

### 3.5 Perfil de Intensidade

Para este problema, foi usado novamente o Matplotlib, para a confecção do gráfico, através da função *plota\_perfil\_linha*. O eixo  $x$  representa a intensidade do pixel, e o  $y$ , a quantidade de vezes que tal intensidade apareceu na linha. Esses valores foram tirados do método *perfil\_linha*, que recebe um número que indica o índice da linha da imagem que se deseja plotar o perfil de intensidade como variável, e retorna uma lista com 256 intensidades (valor puramente arbitrário no momento), a qual indica, pelo índice de cada elemento, a intensidade de pixel, e pelo valor do elemento, a frequência com que a intensidade aparece na linha escolhida.

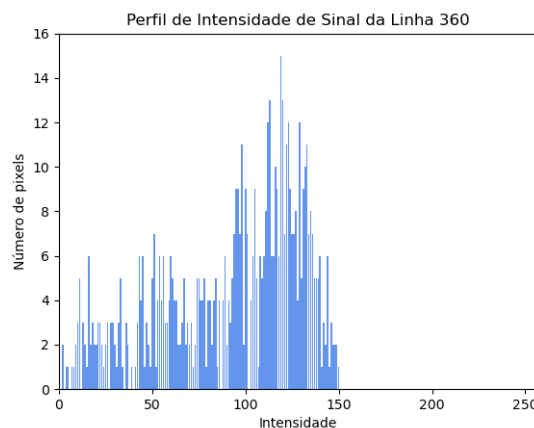


Figura 3: Perfil de intensidade da linha 360

Como pode-se ver pela Figura 3, a linha escolhida tem intensidades relativamente bem distribuídas, com exceção das intensidades mais claras.

## 4 Conclusão

Em resumo, este trabalho proporcionou uma introdução ao processamento e análise de imagens digitais, com foco na representação de imagens em escala

de cinza. Inicialmente, exploramos os fundamentos teóricos da representação digital de uma imagem, compreendendo a discreta função  $f(x, y)$  que descreve a intensidade de luz em cada pixel.

Em seguida, desenvolvemos uma biblioteca em Python para manipulação de imagens em formato .pgm, permitindo a abertura, visualização e análise dessas imagens. Utilizando ferramentas como Matplotlib, conseguimos imprimir imagens em tons de cinza e coloridas, facilitando a visualização e interpretação dos resultados.

A partir da análise da imagem escolhida, determinamos suas dimensões, como sendo  $480 \times 640$ , assim como os valores máximo e mínimo de intensidade, como sendo 208 e 0. Esses resultados nos permitem inferir, sem muito rigor, uma resolução de escala de cinza para imagem. Tendo em vista os argumentos apresentados na Seção 2.3, sabe-se que o aumento da quantidade dos níveis de cinza não necessariamente melhora a qualidade de uma imagem; e, tendo em vista as dimensões, e o valor máximo, não seria de todas, a melhor escolha, escolher uma resolução maior do que o nível de intensidade máximo da imagem. Além disso, convém-se que é melhor trabalhar com potências de base 2. Assim sendo, infiro que o ideal, seria normalizar as intensidades para a escala de 0 a 127. Isso resolve a problemática da resolução da escala de cinza.

Discutimos, também, a distribuição de intensidades ao longo de uma linha específica, e como foi construído o gráfico utilizado.

Embora este trabalho tenha fornecido uma compreensão básica da representação digital de imagens e explorado algumas técnicas iniciais de análise, reconhecemos que existem limitações e oportunidades de melhoria. Possíveis extensões incluem a implementação de algoritmos mais avançados de processamento de imagens e a exploração de diferentes tipos de imagens e formatos de arquivo.