

**Universidade do Estado do Amazonas (UEA)**

**Escola Superior de Tecnologia (EST)**

**Curso:** Engenharia de Computação

**Data:** 4 de dezembro de 2019

**Disciplina:** Processamento Digital de Imagens

**Professor:** Ricardo Barbosa

**Alunos:**

CARLOS DIEGO FERREIRA DE ALMEIDA

JEAN PHELIPE DE OLIVEIRA LIMA

JOÃO VICTOR MELO DE OLIVEIRA

LUIZ CARLOS SILVA DE ARAÚJO FILHO

## Atividade Prática

### Transformações de Intensidade em Imagens

O desenvolvimento dos métodos de processamento digital de imagens são de grande importância por dois motivos: melhoram a informação visual para interpretação humana e aumentam o desempenho para armazenamento, transmissão e representação de imagens em computadores. Todos esses métodos são aplicados, hoje, a partir de linguagens de programação como Python3 através de APIs como OpenCV. Esse trabalho visa abordar conceitos de PDI e de OpenCV para apresentar uma solução à atividade prática nele apresentada.

## 1 Fundamentação Teórica

Nesta seção são descritos os principais conceitos e técnicas que serviram de base para o desenvolvimento deste trabalho.

### 1.1 Processamento Digital de Imagens

O campo do processamento digital de imagens (PDI) se refere ao processamento de imagens digitais por um computador digital. Observe que uma imagem digital é composta de um número finito de elementos, cada um com localização e valor específicos. Esses elementos são chamados de elementos pictóricos, elementos de imagem, pels e *pixels*. *Pixel* é o termo mais utilizado para representar os elementos de uma imagem digital.

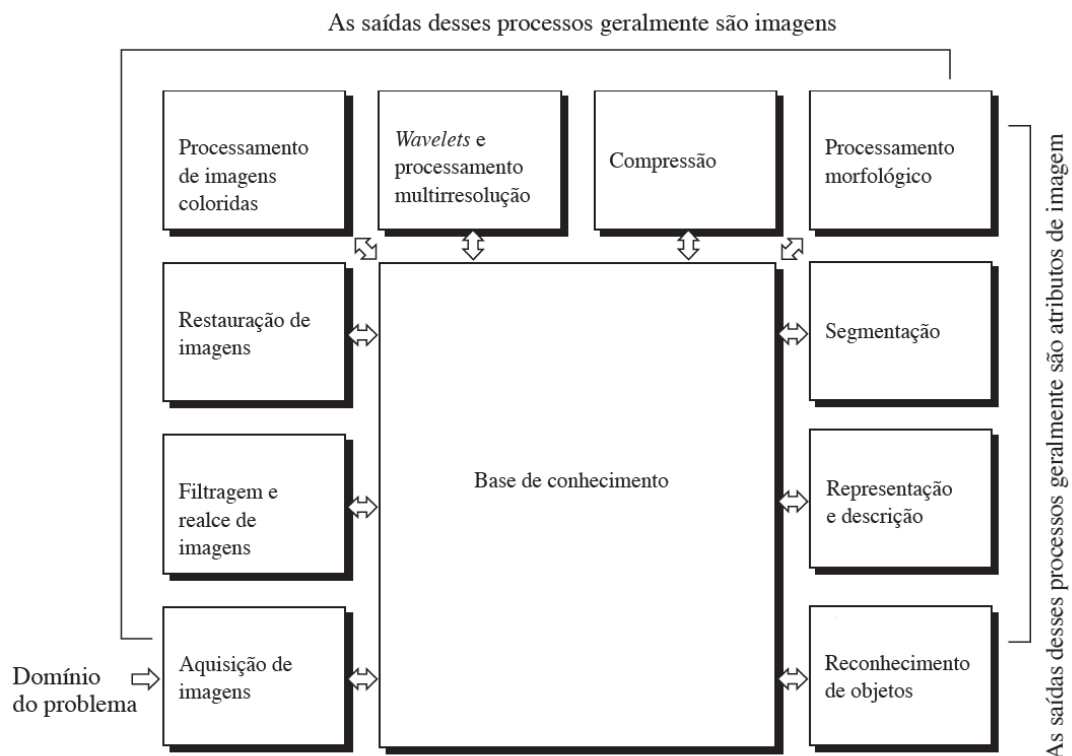
Uma imagem pode ser definida como uma função bidimensional,  $f(x, y)$ , em que  $\mathbf{x}$  e  $\mathbf{y}$  são coordenadas espaciais (no plano), e a amplitude de  $\mathbf{f}$  em qualquer par de coordenadas  $(x, y)$  é chamada de intensidade ou nível de cinza da imagem nesse ponto. Quando  $\mathbf{x}$ ,  $\mathbf{y}$  e os valores de intensidade de  $\mathbf{f}$  são quantidades finitas e discretas, chamamos de imagem digital.

Há três tipos de processos computacionais em PDI: processos de níveis baixo, médio e alto. Os processos de nível baixo envolvem operações primitivas, como o pré-processamento de imagens para reduzir o ruído, o realce de contraste e o aguçamento de imagens. Um processo de nível baixo é caracterizado pelo fato de tanto a entrada

quanto a saída serem imagens. O processamento de imagens de nível médio envolve tarefas como a segmentação (separação de uma imagem em regiões ou objetos), a descrição desses objetos para reduzi-los a uma forma adequada para o processamento computacional e a classificação (reconhecimento) de objetos individuais. Um processo de nível médio é caracterizado pelo fato de suas entradas, em geral, serem imagens, mas as saídas são atributos extraídos dessas imagens (isto é, bordas, contornos e a identidade de objetos individuais). Por fim, o processamento de nível alto envolve “dar sentido” a um conjunto de objetos reconhecidos, como na análise de imagens e, no extremo dessa linha contínua, realizar as funções cognitivas normalmente associadas à visão.

### 1.1.1 Passos Fundamentais em Processamento Digital de Imagens

A Figura 1 resume todas as metodologias que podem ser aplicadas em imagens para diferentes propósitos e objetivos. Para o escopo da atividade, são necessários apenas os quadros que são descritos nos próximos parágrafos.



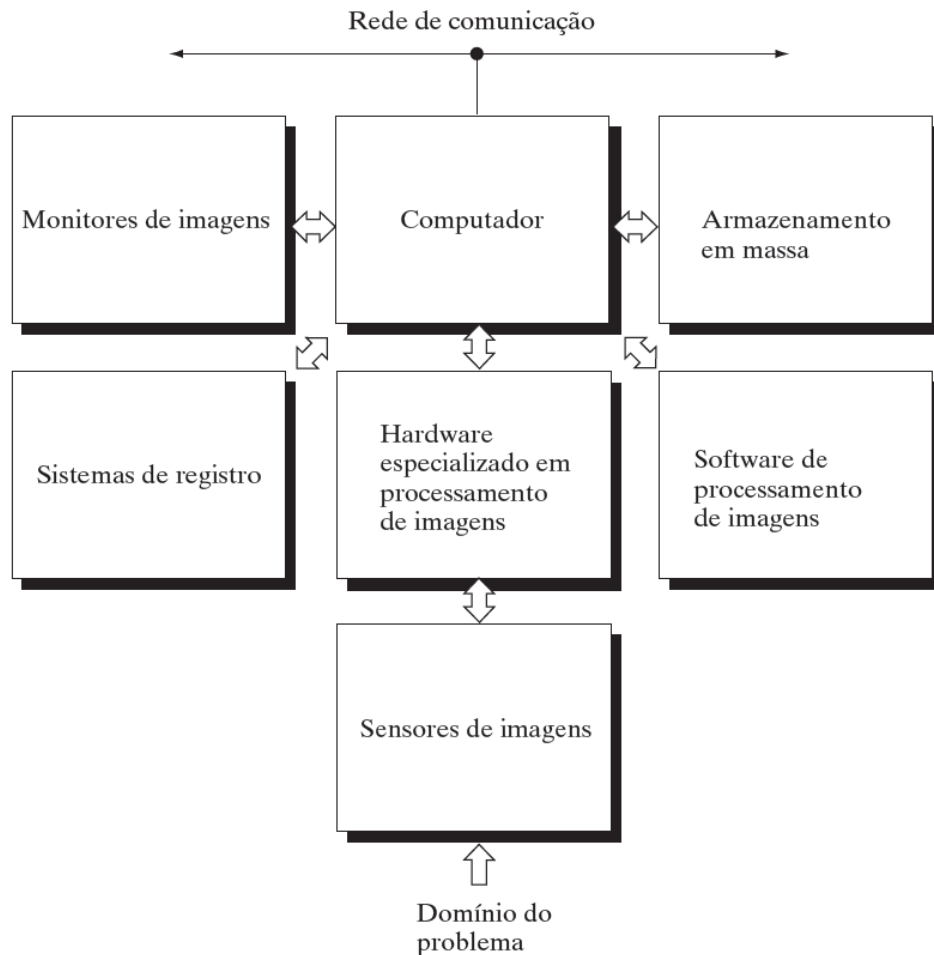
**Figura 1:** Passos fundamentais em PDI.

A **aquisição de imagens** é o primeiro processo da Figura 1. A aquisição pode ser tão simples quanto receber uma imagem que já esteja em formato digital ou através de um sensor de aquisição de imagem (ver seção 1.2.1). Em geral, o estágio de aquisição de imagens envolve um pré-processamento, por exemplo, o redimensionamento de imagens.

O **realce de imagens** é o processo de manipular uma imagem de forma que o resultado seja mais adequado do que o original para uma aplicação específica. A palavra *específica* é importante neste contexto, porque estabelece desde o início que as técnicas de realce são orientadas de acordo com o problema. Dessa forma, por exemplo, um método bastante útil para realçar imagens radiográfica pode não ser a melhor abordagem para realçar imagens de satélite capturadas na banda infravermelha do espectro eletromagnético.

### 1.1.2 Componentes de um Sistema de Processamento de Imagem

A Figura 2 mostra os componentes básicos que constituem um sistema de uso geral típico para o processamento digital de imagens. A função de cada componente é discutida nos próximos parágrafos.



**Figura 2:** Componentes de um Sismate de PDI.

No que se refere ao **sensoriamento**, dois elementos são necessários para a aquisição de imagens digitais. O primeiro é um dispositivo físico sensível à energia irradiada pelo objeto cuja imagem desejamos capturar. O segundo, chamado de digitalizador, é um dispositivo utilizado para converter a saída do dispositivo físico de sensoriamento em formato digital. Por exemplo, em uma filmadora digital, os sensores produzem uma saída elétrica proporcional à intensidade da luz. O digitalizador converte essa saída em dados digitais.

O **hardware especializado** em processamento de imagens normalmente consiste no digitalizador mencionado acima, além de um hardware capaz de desempenhar outras operações primárias, como uma unidade lógica e aritmética (ALU, sigla em inglês), que realiza operações aritméticas e lógicas em paralelo em toda a imagem. Um exemplo de como uma ALU pode ser utilizada está no cálculo da média de uma imagem à medida que ela é digitalizada, com o propósito de redução de ruídos.

O **computador** em um sistema de processamento de imagens é um computador de uso geral, que pode variar de um computador pessoal a um supercomputador. Em aplicações especiais, algumas vezes computadores especializados são utilizados para atingir

o nível necessário de desempenho. Para realização da atividade prática, um sistema de processamento de imagens de uso geral foi utilizado. Nesses sistemas, praticamente qualquer computador pessoal bem equipado é suficiente para as tarefas de processamento de imagens *offline*.

O **software** para o processamento de imagens consiste em módulos especializados que realizam tarefas específicas. Um bom pacote computacional também inclui a possibilidade de o usuário escrever códigos que, no mínimo, utilizem os módulos especializados. Os pacotes de aplicativos mais sofisticados permitem a integração desses módulos e dos comandos gerais de *software* a partir de pelo menos uma linguagem computacional. Para a realização da atividade prática, o pacote utilizado é o OpenCV e os códigos são escritos na linguagem de programação Python3.

A capacidade de **armazenamento em massa** é indispensável em aplicações de processamento de imagens. Uma imagem do tamanho de  $1.024 \times 1.024$  *pixels*, na qual a intensidade de cada pixel requer 8 *bits*, necessita de um espaço de armazenamento de 1 *megabyte*, se a imagem não for comprimida. Ao lidar com milhares, ou até milhões, de imagens, o armazenamento adequado em um sistema de processamento de imagens pode ser um desafio.

Os *monitores de imagem* utilizados hoje em dia são, em sua maioria, monitores de TV em cores (preferencialmente de tela plana). Os monitores são controlados pelas placas de vídeo (gráficas ou de imagens), que são parte integral de um sistema computacional. Raramente os requisitos das aplicações de visualização de imagens não podem ser satisfeitos pelas placas de vídeo disponíveis comercialmente como parte do sistema computacional.

A *rede de comunicação* é quase um componente padrão de qualquer sistema computacional em uso hoje em dia. Em razão do grande volume de dados inerente às aplicações de processamento de imagens, a principal preocupação na transmissão de imagens é a largura de banda. Em redes dedicadas, isso normalmente não constitui um problema, mas as comunicações com sites remotos pela Internet nem sempre são eficientes. Felizmente, essa situação está melhorando rapidamente como resultado do advento da fibra óptica e de outras tecnologias de banda larga.

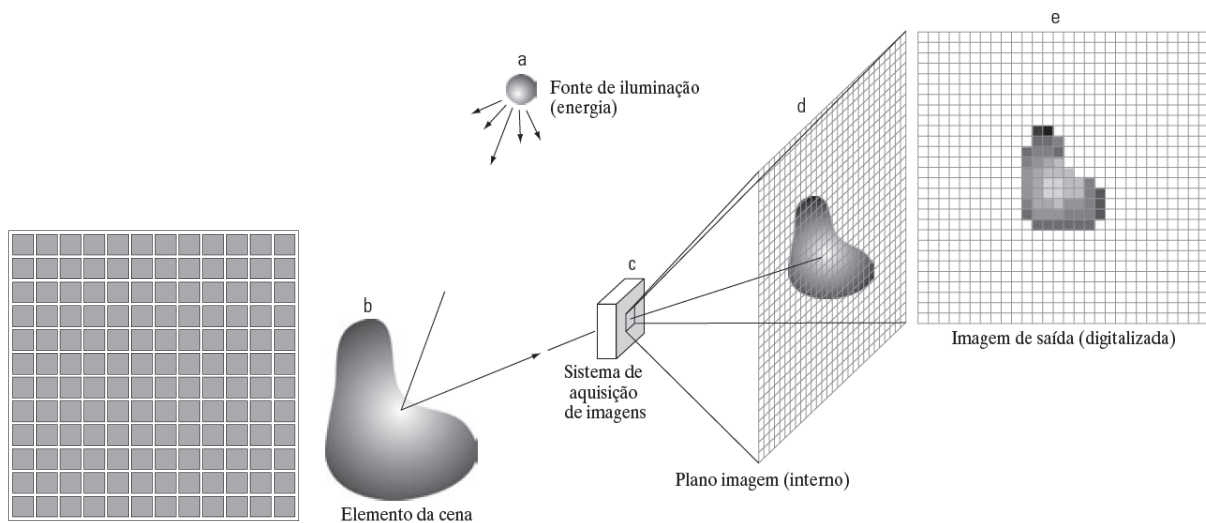
## 1.2 Fundamentos da Imagem Digital

Esta subseção descreve alguns conceitos relacionados a imagem digital que são aplicados na atividade prática.

### 1.2.1 Sensores de Aquisição de Imagens

A luz é uma onda eletromagnética onde a energia da iluminação é refletida pelos objetos ou transmitida através deles. Os sensores tem a funcionalidade básica de transformar a energia de iluminação em imagens digitais. A ideia é simples: a energia que entra é transformada em tensão pela combinação da energia elétrica de entrada e do material do sensor, sensível a um tipo específico de energia que está sendo detectado. A forma de onda da tensão de saída é a resposta do(s) sensor(es), e uma quantidade digital é obtida de cada sensor por meio da digitalização de sua resposta.

Há diversos tipos de sensores na literatura e na indústria. Um tipo bastante utilizado é o sensor de arranjo matricial. Esse é o arranjo predominante encontrado nas câmeras digitais. Um sensor típico para essas câmeras é uma matriz CCD, que pode ser fabricada com uma grande variedade de propriedades sensoras e dispostas em arranjos matriciais de  $4.000 \times 4.000$  elementos ou mais. A resposta de cada sensor é proporcional à integral da



**Figura 3:** Sensor de área matricial.

**Figura 4:** Exemplo do processo de aquisição de imagem.

energia luminosa projetada sobre a superfície do sensor. Pelo fato de a matriz de sensores da Figura 3 ser bidimensional, sua principal vantagem é que uma imagem completa pode ser obtida projetando o padrão de energia na superfície da matriz.

A principal forma na qual os sensores matriciais são utilizados é mostrada na Figura 4. Essa figura mostra a energia de uma fonte de iluminação sendo refletida de um elemento de uma cena. A primeira função realizada pelo sistema de aquisição de imagens da Figura 4(c) é coletar a energia de entrada e projetá-la em um plano imagem. Se a iluminação for luz, a entrada frontal do sistema de aquisição de imagens é uma lente ótica que projeta a cena vista sobre o plano focal da lente, como mostra a Figura 4(d). O arranjo de sensores, que coincide com o plano focal, produz saídas proporcionais à integral da luz recebida em cada sensor. Circuitos digitais e analógicos realizam uma varredura nessas saídas e as convertem em um sinal analógico, que é então digitalizado por um outro componente do sistema de aquisição de imagens. A saída é uma imagem digital, como mostra o esquema da 4(e).

### 1.2.2 Amostragem e Quantização de Imagens

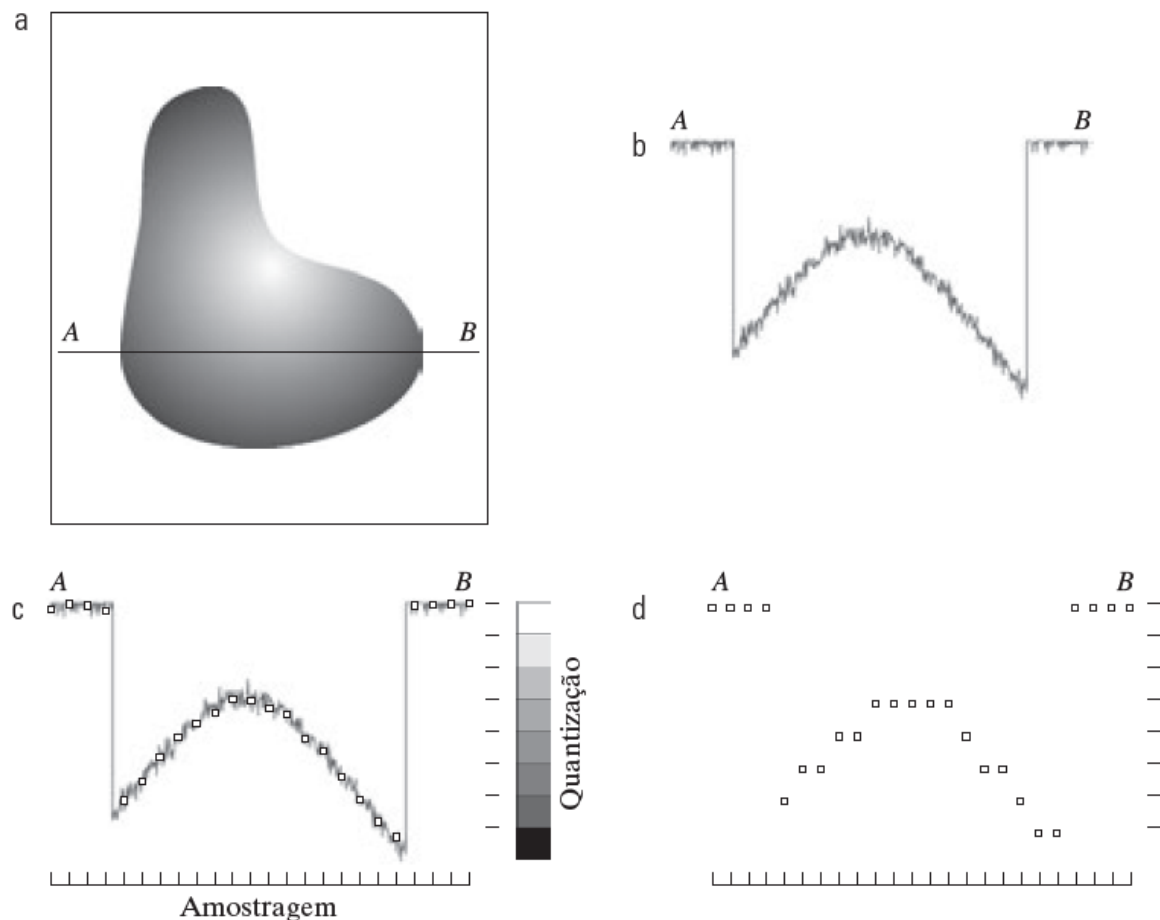
A saída da maioria dos sensores consiste de uma forma de onda de tensão contínua cuja amplitude e o comportamento no espaço estão relacionados ao fenômeno físico que está sendo captado pelos sensores. Para criar uma imagem digital a partir de dados capturados por sensores precisamos converter os dados contínuos que foram captados para o formato digital. Isso envolve dois processos: amostragem e quantização.

A ideia básica por trás da *amostragem* e da *quantização* é ilustrada na Figura 2.16. A Figura 2.16(a) mostra uma imagem contínua  $f$  que queremos converter em formato digital. Uma imagem pode ser contínua em relação às coordenadas  $x$  e  $y$  e também em relação à amplitude. Para convertê-la ao formato digital, temos de fazer a amostragem da função em ambas as coordenadas e na amplitude. A digitalização dos valores de coordenada é chamada de amostragem. A digitalização dos valores de amplitude é chamada de quantização.

A função unidimensional da Figura 5(b) é um gráfico que representa os valores de amplitude (nível de intensidade) da imagem contínua ao longo do segmento de reta  $AB$

na Figura 5(a). As variações aleatórias se devem ao ruído da imagem. Para realizar a amostragem dessa função, colhemos amostras igualmente espaçadas ao longo da linha  $AB$ , como mostra a Figura 5(c). A posição de cada amostra no espaço é indicada por uma pequena marca vertical na parte inferior da figura. As amostras são representadas por pequenos quadrados brancos superpostos na função. O conjunto dessas localizações discretas nos dá a função de amostragem. No entanto, os valores das amostras ainda cobrem (verticalmente) uma faixa contínua de valores de intensidade. Para formar uma função digital, os valores de intensidade também devem ser convertidos (quantizados) em quantidades discretas. O lado direito da Figura 5(c) mostra a escala de intensidade dividida em oito intervalos discretos, variando do preto ao branco. As marcas verticais indicam o valor específico atribuído a cada um dos oito níveis de intensidade. Os níveis de intensidade contínuos são quantizados atribuindo um dos oito valores para cada amostra. Essa atribuição é feita dependendo da proximidade vertical de uma amostra a uma marca indica dora.

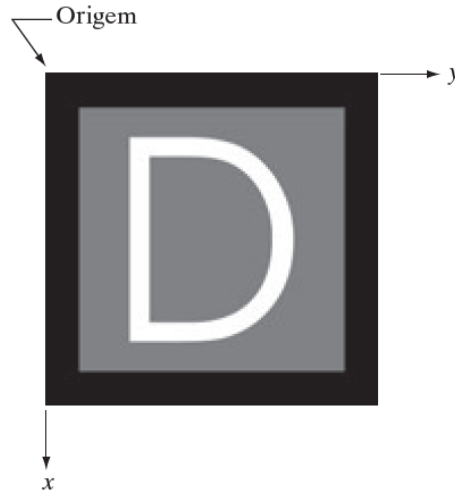
As amostras digitais resultantes da amostragem e da quantização são mostradas na 5(d). Ao começar na parte superior da imagem e realizar esse procedimento linha por linha, produz-se uma imagem digital bidimensional. Está implícito na 5 que, além do número discreto de níveis utilizados, a precisão atingida na quantização depende muito do conteúdo de ruído do sinal da amostragem.



**Figura 5:** Processo de produção uma imagem digital.

### 1.2.3 Representação de Imagens Digitais

Seja  $f(s, t)$  uma função de imagem contínua de duas variáveis contínuas,  $s$  e  $t$ . Convertamos essa função em uma imagem digital por meio de amostragem e quantização. Suponha que realizemos a amostragem da imagem contínua em uma matriz 2D,  $f(x, y)$ , contendo  $M$  linhas e  $N$  colunas, sendo que  $(x, y)$  são coordenadas discretas. Utilizamos números inteiros para essas coordenadas discretas:  $x = 0, 1, 2, \dots, M - 1$  e  $y = 0, 1, 2, \dots, N - 1$ . Em geral, o valor da imagem em quaisquer coordenadas  $(x, y)$  é expresso por  $f(x, y)$ , onde  $x$  e  $y$  são números inteiros.



**Figura 6:** Imagem representada como uma matriz de intensidade visual.

A representação da Figura 6 é a mais comum representação de imagem digital. Ela mostra  $f(x, y)$  como uma imagem que seria visualizada em um monitor ou uma fotografia. Aqui, o nível de cinza de cada ponto é proporcional ao valor da intensidade  $f$  desse ponto. Nessa figura, temos apenas três valores de intensidade igualmente espaçados. Se a intensidade for normalizada para o intervalo  $[0, 1]$ , cada ponto da imagem tem o valor 0, 0,5 ou 1.

Matrizes numéricas são utilizadas para o processamento de imagens digitais e desenvolvimento de algoritmos que as manipulam. Na forma de equação, escrevemos a representação de uma matriz numérica  $M \times N$  como

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & f(0, 2) & \dots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & f(1, 2) & \dots & f(1, N - 1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & f(M - 1, 2) & \dots & f(M - 1, N - 1) \end{bmatrix}$$

Os dois lados dessa equação são formas equivalentes de representar quantitativamente uma imagem digital. O lado direito é uma matriz de números reais. Cada elemento dessa matriz é um *pixel*.

Em algumas discussões é vantajoso utilizar uma notação matricial mais tradicional para expressar uma imagem digital e seus elementos:

$$X = \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & \dots & x_{0,N-1} \\ x_{1,0} & x_{1,1} & x_{1,2} & \dots & x_{1,N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{M-1,0} & x_{M-1,1} & x_{M-1,2} & \dots & x_{M-1,N-1} \end{bmatrix}$$

As matrizes  $f$  e  $X$  são equivalentes, já que  $a_{ij} = f(x = i, y = j) = f(i, j)$ .

Podemos formatar as matrizes acima para representar uma imagem em um vetor coluna de tamanho  $MN \times 1$ . Esse processo é frequentemente necessário quando estamos trabalhando com imagens em linguagens de programação.

Observe na 6 que a origem de uma imagem digital se localiza na parte superior esquerda, com o eixo  $x$  positivo se estendendo para baixo e o eixo  $y$  positivo se estendendo para a direita. Isso se deve ao modo de como imagens são renderizadas em monitores, porém mais importante que isso é o fato de que o primeiro elemento de uma matriz é, por convenção, o elemento do canto superior esquerdo, de forma que a escolha da origem de  $f(x, y)$  nesse ponto faz sentido matematicamente.

#### 1.2.4 Resolução Espacial

A resolução espacial é uma medida do menor detalhe discernível em uma imagem. Quantitativamente, a resolução espacial pode ser expressa em várias formas, sendo que as mais comuns são pares de linha por unidade de distância e *pixels* por unidade de distância.

Para serem significativas, as medidas de resolução espacial devem ser expressas com relação a unidades espaciais. O tamanho da imagem por si só não diz tudo. Dizer que uma imagem tem, digamos, uma resolução de  $1.024 \times 1.024$  *pixels* não faz muito sentido se as dimensões espaciais da imagem não forem especificada.

### 1.3 Transformações de Intensidade

As duas principais categorias do processamento espacial são transformações de intensidade e filtragem espacial. As transformações de intensidade operam individualmente nos *pixels* de uma imagem, principalmente para fins de manipulação de contraste e limiarização de imagem. A filtragem espacial lida com a realização de operações como o realce de imagens, trabalhando na vizinhança de cada pixel de uma imagem. Para a atividade deste trabalho, apenas transformações de intensidade foram utilizadas.

#### 1.3.1 Domínio Espacial e Realce

A expressão *domínio espacial* se refere ao próprio plano imagem, e os métodos de processamento de imagens nessa categoria se baseiam na manipulação direta de *pixels* em uma imagem.

Os processos no domínio espacial podem ser expressos por:

$$g(x, y) = T[f(x, y)] \quad (1)$$

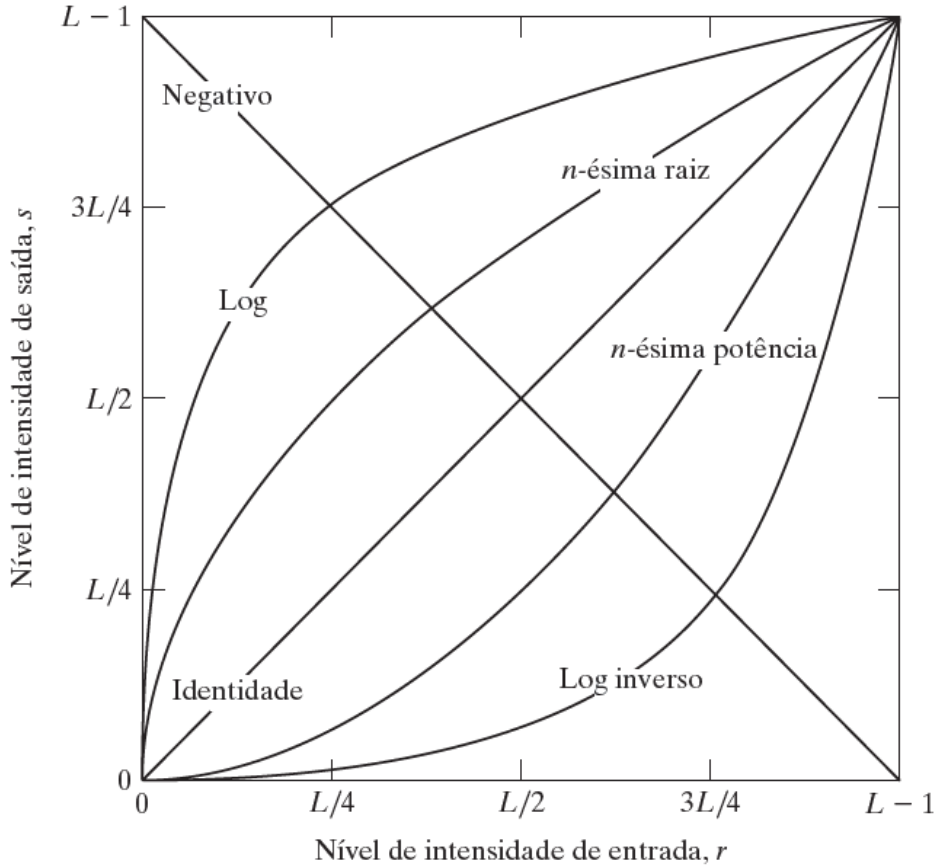
onde  $f(x, y)$  é a imagem de entrada,  $g(x, y)$  é a imagem de saída, e  $T$  é um operador em  $f$  definido em uma vizinhança do ponto  $(x, y)$ .

*Realce* é o processo de manipular uma imagem de forma que o resultado seja mais adequado do que o original para uma aplicação específica. Não existe uma "teoria" geral para o realce de imagens. Quando uma imagem é processada para a interpretação visual, o observador é o juiz em relação ao desempenho de um método particular.



### 1.3.2 Transformações Logarítmicas

As transformações de intensidade estão entre as mais simples de todas as técnicas de processamento de imagens. Os valores dos *pixels* antes e depois do processamento serão representados por  $r$  e  $s$ , respectivamente.



**Figura 7:** Funções básicas de transformação de intensidade.

A Figura 7 mostra três tipos básicos de funções de transformações de intensidade frequentemente utilizadas para o realce de imagens: linear (transformações de negativo e de identidade), logarítmica (transformações de *log* e *log inverso*) e de potência (transformações de *n-ésima potência* e *n-ésima raiz*). A função identidade é o caso trivial no qual as intensidades de saída são idênticas às intensidades de entrada. Ela foi incluída no gráfico só para uma abrangência completa.

Este trabalho foca na transformação logarítmica por ser esta a função aplicada na atividade prática.

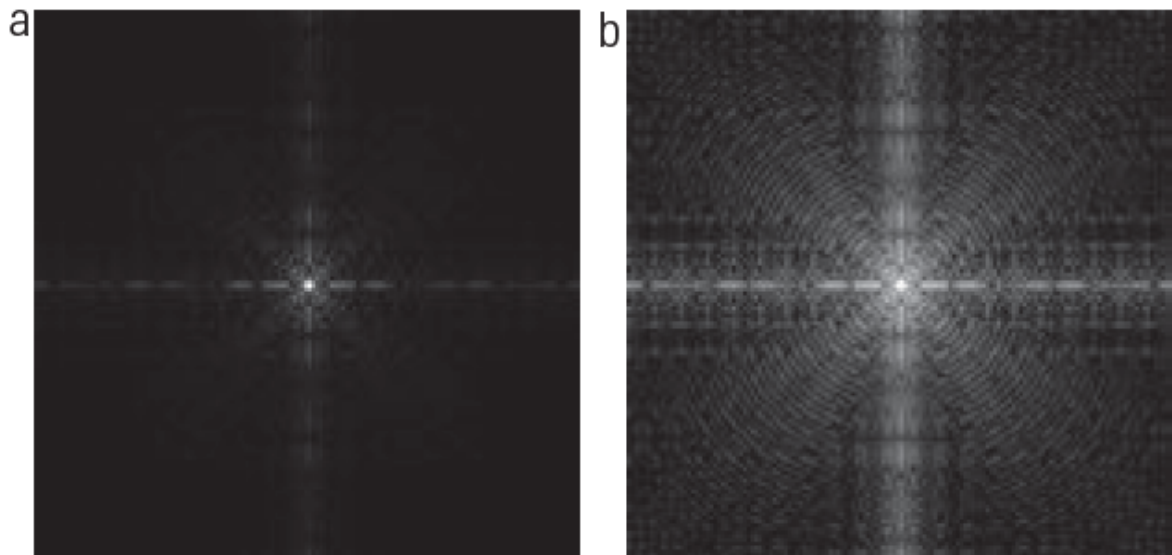
A forma geral da transformação logarítmica da Figura 7 é:

$$s = c \log(1 + r) \quad (2)$$

onde  $c$  é uma constante e considera-se que  $r \geq 0$ . O formato da curva logarítmica na Figura 7 mostra que essa transformação mapeia uma faixa estreita de baixos valores de intensidade de entrada em uma faixa mais ampla de níveis de saída. O oposto se aplica aos valores mais altos de níveis de intensidade de entrada. Utilizamos uma transformação desse tipo para expandir os valores de *pixels* mais escuros em uma imagem ao mesmo

tempo em que comprimimos os valores de nível mais alto. O oposto se aplica à transformação logarítmica inversa. Portanto, a função logarítmica tem a importante característica de comprimir a faixa dinâmica das imagens com grandes variações de valores de *pixels*.

Como uma ilustração das transformações logarítmicas, a Figura 8(a) mostra um espectro de *Fourier* com valores variando de 0 a  $1,5 \times 10^6$ . Quando esses valores são linearmente ajustados para serem exibidos em um sistema de 8 *bits*, os *pixels* mais claros dominam a exibição em detrimento dos valores mais baixos (e tão importantes quanto) do espectro. O efeito dessa dominância é ilustrado pela área relativamente pequena da imagem na Figura 8(a) que não é percebida como preta. Se, em vez de exibir os valores dessa maneira, aplicarmos inicialmente a Equação 2 (com  $c = 1$  neste caso) aos valores do espectro, a faixa de valores do resultado passa a ser de 0 a 6,2, que é a mais apropriada. A Figura 8(b) mostra o resultado de ajustar linearmente essa nova faixa e exibir o espectro no mesmo monitor de 8 *bits*. A riqueza de detalhes visíveis nessa imagem em comparação com uma exibição não modificada do espectro fica clara nessas imagens.



**Figura 8:** Exemplo de aplicação da função logarítmica.

## 2 OpenCV

OpenCV (Open Source Computer Vision) é uma biblioteca de visão computacional e *machine learning* de programação de código aberto com o objetivo de tornar o processamento de imagens mais acessível a desenvolvedores e hobistas e pode ser utilizada em diversas linguagens de programação (C++, Python, Java e MATLAB).

A biblioteca possui mais de 2500 algoritmos otimizados, dos quais incluem um compreensivo conjunto de algoritmos clássicos e de estado-da-arte de visão computacional e *machine learning*. Esses algoritmos podem ser usados para detectar e reconhecer rostos, identificar objetos, classificar ações humanas em vídeos, rastrear movimentos de objetos, juntar imagens para produzir uma imagem de alta resolução de uma cena inteira, encontrar imagens similares em um conjunto de imagens, etc. OpenCV tem mais de 47 mil de usuários ativos na comunidade e mais de 18 milhões de *downloads*. A biblioteca é utilizada extensivamente por *hobistas*, companhias, grupos de pesquisadores e governos.

OpenCV-Python é a biblioteca em questão compatível para Python. Ela faz uso da biblioteca de operações numéricas otimizadas do Python, a Numpy. É possível escrever códigos em C++ dentro de um código Python utilizando *wrappers*. Dessa maneira, essa biblioteca é um *wrapper* da implementação original da OpenCV-C++.

A atividade prática deste trabalho foi realizada utilizando as abstrações e comodidades oferecidas pela OpenCV-Python.

### 3 Atividade e Metodologia

A atividade consiste em escolher uma imagem ao gosto do aluno e realizar uma transformação logarítmica com intuito de realçar características da mesma. A imagem em questão deve ter baixa riqueza de detalhes para que sejam vistos os efeitos da transformação.

Primeiramente, a atividade foi dividida por integrante. O programa foi desenvolvido em máquinas relativamente modernas dos próprios alunos, em ambiente *Windows* ou *iOS*, e após instalação de pacotes necessários e solução de problemas do processo de instalação. Em seguida, cada um estudou individualmente uma ou mais maneiras de resolver o problema. Majoritariamente, o procedimento de desenvolvimento consistiu em leitura de artigos pela *web* e visualização de vídeo-aulas em plataformas *online*. Após solucionar o problema, a equipe se reuniu e discutiu os códigos-fontes.

Os códigos da equipe foram desenvolvidos todos em Python por questões de comodidade. Apesar da existência de rotinas prontas para transformações de intensidades na OpenCV, os integrantes desenvolveram suas próprias rotinas baseadas no conteúdo do livro-texto, conforme solicitado pelo professor.

Por fim, a equipe decidiu juntar as melhores partes dos códigos desenvolvidos para apresentar como código-fonte final.

### 4 Resultados

O programa escrito para realizar a atividade é apresentado no bloco de código abaixo. Como pode ser visto, a rotina `log_enhancement()` foi criada para realizar a processo de realce. A biblioteca Numpy é utilizada para usarmos a função `log()`. Há também uso de rotinas OpenCV para auxiliar na manipulação e renderização da imagem.

Primeramente, é feita a leitura da imagem original. Em seguida, convertemos a imagem para escala de cinza e, por fim, realizamos a operação alvo nela. O resultado do programa pode ser visto na Figura 10 abaixo.

---

```
import numpy as np
import cv2

#realce logaritmico
def log_enhancement(img):
    for i in range(len(img)):
        for j in range(len(img[0])):
            img[i][j] = int((0.15*(np.log(1 + img[i][j])))*255)
```

```

    return img

#leitura da imagem
image = cv2.imread('1 - paisagem.jpg')

#transformao da imagem para escala de cinza
image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
cv2.imshow('gray', image)
cv2.imwrite('2 - grayscale.jpg', image)

#realce logaritmico
image = log_enhancement(image)
cv2.imshow('log_enhancement', image)
cv2.imwrite('3 - log_enhancement.jpg', image)

cv2.waitKey(0)

```

---



**Figura 9:** Imagem original em escala de cinza.

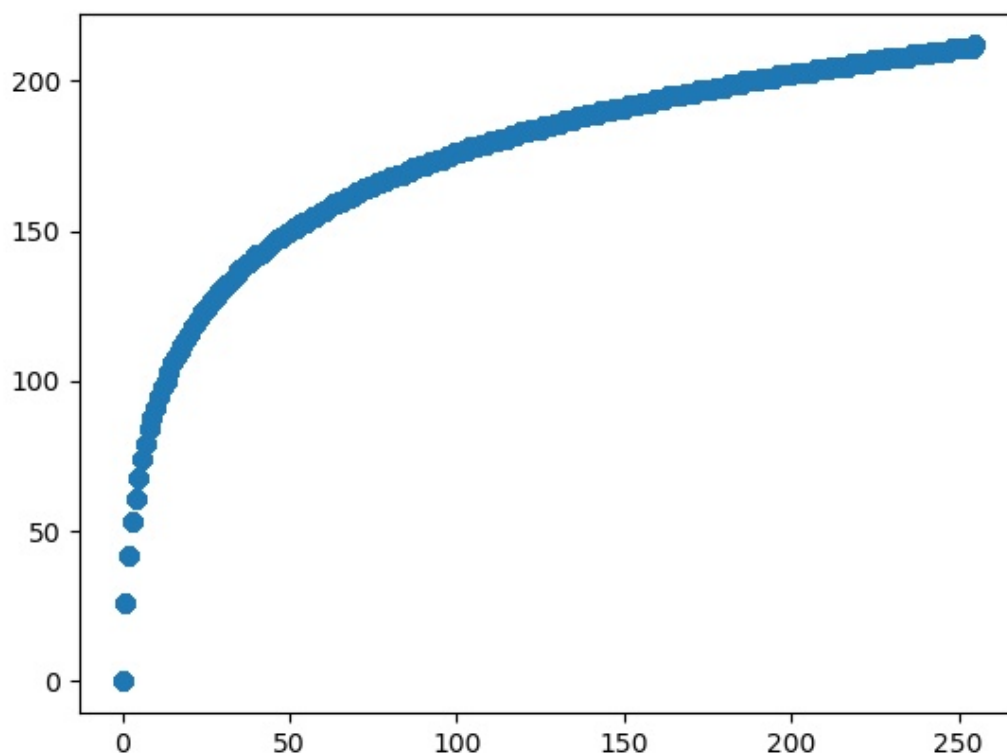


**Figura 10:** Imagem após a operação logarítmica.

A transformada utilizada é a função logarítmica abaixo, representada graficamente na Figura 11:

$$Y(i, j) = \lfloor 0.15 * (\log(1 + X(i, j))) * 255 \rfloor,$$

onde  $X$  é a imagem original e  $Y$  é a imagem realçada após a aplicação da transformada. A função leva cada *pixel*  $(i, j)$  de  $X$  para um intervalo normalizado  $[0, 1]$ . Esse valor corresponde ao fator de correção do *pixel*, por isso é feita a multiplicação por 255 para levar o valor normalizado à escala de intensidade de *pixels*,  $[0, 255]$ . Por fim, a função piso é aplicada para se obter a parte inteira do valor encontrado.



**Figura 11:** Gráfico função de realce logarítmico.

## Referências

- [1] GONZALEZ, R. C.; WOODS, R. E. *Processamento Digital de Imagens*. 3. ed. São Paulo - SP, Brasil: Pearson Education do Brasil, 2010.
- [2] Merlon de Alencar Rocha. *OpenCV - Uma breve introdução*. 2018. Disponível em <https://blog.cedrotech.com/opencv-uma-breve-introducao-visao-computacional-com-python/>. Acesso em 4 de dezembro de 2019.
- [3] OpenCV Team. *About OpenCV*. 2019. Disponível em <https://opencv.org/about/>. Acesso em 4 de dezembro de 2019.
- [4] Alexander Mordvintsev e Abid Rahman. *Introduction to OpenCV-Python Tutorials*. 2013. Disponível em [https://docs.opencv.org/master/d0/de3/tutorial\\_py\\_intro.html](https://docs.opencv.org/master/d0/de3/tutorial_py_intro.html). Acesso em 4 de dezembro de 2019.