

# Anotações sobre a Iniciação Científica

Aluno            Heitor Barroso Cavalcante - 12566101

Professora    Nina S. T. Hirata

21 de Abril de 2022

## Conteúdo

<b>1</b>	<b>Processamento de Imagens Digitais</b>	<b>1</b>
1.1	Introdução . . . . .	1
1.2	Introdução à Sinais e Sistemas . . . . .	2
1.2.1	Sinais . . . . .	2
1.2.2	Sistemas . . . . .	2
<b>2</b>	<b>Aplicações do Processamento de Imagens Digitais</b>	<b>2</b>
<b>3</b>	<b>Dimensões das Imagens</b>	<b>3</b>
<b>4</b>	<b>Formação de Imagens em Câmeras</b>	<b>3</b>
4.1	Formação de Imagens em Câmeras Analógicas . . . . .	3
4.2	Formação de Imagens em Câmeras Digitais . . . . .	3
<b>5</b>	<b>Mecanismos das Câmeras</b>	<b>4</b>
5.1	Aperture - Abertura . . . . .	4
5.2	Shutter - Obturador . . . . .	4
5.3	ISO . . . . .	4
5.4	Conceito de Pixel . . . . .	5
5.4.1	Cálculo do número total de pixels . . . . .	5
5.4.2	Valores de pixels . . . . .	5
<b>6</b>	<b>Transformação de Perspectiva</b>	<b>5</b>
6.1	Referencial - Frame of Reference . . . . .	5
<b>7</b>	<b>Conceito de bits por pixel</b>	<b>6</b>
7.1	Cálculo do tamanho (em memória) de uma imagem . . . . .	6

<b>8</b>	<b>Tipos de imagens</b>	<b>7</b>
8.1	Imagem binária . . . . .	7
8.2	Formato 8 bits de cor . . . . .	7
8.3	Formato 16 bits de cor . . . . .	7
8.4	Formato 24 bits de cor . . . . .	8
<b>9</b>	<b>Códigos de cores</b>	<b>8</b>
9.1	Modelo CMYK . . . . .	9
9.2	Conversão RGB para Hexadecimal . . . . .	9
9.3	Conversão Hexadecimal para RGB . . . . .	10
<b>10</b>	<b>Conversão de RGB para Grayscale</b>	<b>10</b>
<b>11</b>	<b>Amostragem - Sampling</b>	<b>11</b>
11.1	Diferenças entre Zoom Analógico e Digital . . . . .	11
<b>12</b>	<b>Resolução de pixels</b>	<b>11</b>
12.1	Proporção da Tela . . . . .	12
<b>13</b>	<b>Mais sobre Zoom</b>	<b>12</b>
13.1	Nearest Neighbor - Pixel Replication . . . . .	12
13.2	Zero-Order Hold - Bloqueio de Ordem Zero . . . . .	13
13.3	K-times Zooming . . . . .	14
<b>14</b>	<b>Resolução Espacial - Spatial Resolution</b>	<b>16</b>
14.1	Pixels por Polegada . . . . .	16
14.2	Pontos por Polegada . . . . .	17
<b>15</b>	<b>Resolução Grayscale</b>	<b>17</b>
<b>16</b>	<b>Conceito de Quantização</b>	<b>17</b>
<b>17</b>	<b>Curvas de Preferência ISO - ISO Preference curves</b>	<b>18</b>
17.1	Curvas de Isopreferência . . . . .	19
<b>18</b>	<b>Conceito de Dithering</b> <b>ACHEI ESSA SEÇÃO CONFUSA</b>	<b>19</b>
<b>19</b>	<b>Introdução a Histogramas</b>	<b>19</b>
19.1	Histograma de uma Imagem . . . . .	19
19.2	Aplicações dos Histogramas . . . . .	19
<b>20</b>	<b>Brilho e Contraste</b>	<b>20</b>
20.1	Contraste . . . . .	20

<b>21</b>	<b>Transformação em imagens</b>	<b>21</b>
<b>22</b>	<b>Deslizamento de Histogramas - Histogram Sliding</b>	<b>21</b>
<b>23</b>	<b>Alongamento de Histogramas - Histogram Stretching</b>	<b>22</b>
<b>24</b>	<b>Probabilidade para Transformações de Histogramas</b>	<b>24</b>
<b>25</b>	<b>Equalização de Histogramas - Histogram Equalization</b>	<b>27</b>
<b>26</b>	<b>Transformações de Níveis de Cinza</b>	<b>29</b>
26.1	Aprimoramento de imagens . . . . .	29
26.2	Transformação de Nível de Cinza . . . . .	29
26.2.1	Transformação Linear . . . . .	29
26.2.2	Transformação Logarítmica . . . . .	30
26.2.3	Transformação Gamma - Power Law - Exponencial . . .	31
<b>27</b>	<b>Conceito de Convolução</b>	<b>32</b>
27.1	Máscara . . . . .	32
27.1.1	Como efetuar convolução utilizando a máscara . . . . .	32
27.1.2	Vantagens da Convolução . . . . .	34
<b>28</b>	<b>Conceito de Máscara</b>	<b>34</b>
<b>29</b>	<b>Conceito de Blurring</b>	<b>34</b>
<b>30</b>	<b>Conceito de bordas - Edges</b>	<b>36</b>
30.1	Prewitt Operator . . . . .	37
30.2	Sobel Operator . . . . .	39
30.3	Robinson Compass Masks . . . . .	40
30.4	Krisch Compass Masks . . . . .	41
30.5	Laplacian Operator . . . . .	42
<b>31</b>	<b>Análise do Freqüência de Domínio - Frequency Domain Analysis</b>	<b>43</b>
<b>32</b>	<b>Series e Transformadas de Fourier</b>	<b>44</b>
<b>33</b>	<b>Series de Fourier</b>	<b>44</b>
<b>34</b>	<b>Teorema da Convolução</b>	<b>45</b>
<b>35</b>	<b>Filtros Passa Alta e Filtros Passa Baixa em Imagens</b>	<b>45</b>

<b>36</b>	<b>Introdução aos Espaços de Cor</b>	<b>45</b>
<b>37</b>	<b>Introdução à Compressão de JPEG</b>	<b>47</b>
<b>38</b>	<b>Reconhecimento Ótico de Caracteres - OCR</b>	<b>47</b>
<b>39</b>	<b>Visão Computacional</b>	<b>47</b>
<b>40</b>	<b>Computação Gráfica</b>	<b>48</b>

# 1 Processamento de Imagens Digitais

## 1.1 Introdução

O processamento de imagens é uma subárea da disciplina de “Processamento de Sinais”, e essa subárea pode ser dividida em processamento digital ou analógico de imagens.

- **Processamento Analógico de Imagens:**  
As imagens são manipuladas através da variação de sinais elétricos, o maior exemplo deste tipo de processamento é a imagem de televisão.
- **Processamento Digital de Imagens:**  
Consiste em desenvolver sistemas digitais para manipular imagens.

Nesse sentido, imagens podem ser compreendidas como um sinal bidimensional e pode ser definida como uma função  $f(x, y)$  tal que  $(x, y)$  são as coordenadas de um pixel e  $f(x, y)$  é o valor deste pixel. Assim, figuras visualizadas em um computador, por exemplo são matrizes de inteiros que variam de 0 a 255. As dimensões da imagem são as dimensões da matriz.

Um sinal, pode ser definido como qualquer quantidade física medida através do tempo, do espaço ou qualquer outra dimensão. Logo, uma imagem digital pode ser caracterizada como um sinal de duas dimensões.

A formação de uma imagem digital parte de um processo físico. A luz refletida pelos objetos fotografados é medida por diversos sensores e uma tensão contínua é gerada de acordo com a quantidade de luz captada pelos sensores. Agora, esse sinal analógico deve ser convertido para um digital. Para isso, amostragem e quantização são utilizadas para gerar a matriz de números que forma a imagem digital.

Visão Computacional consiste no desenvolvimento de um sistema que consiga, a partir de uma imagem, gerar uma saída contendo informações sobre a imagem. Por exemplo, sistemas com reconhecimento facial.

Por fim, resumindo, o processamento de sinais é muito importante para o processamento de imagens. Sensores captam a luz do mundo físico, resultando em um sinal bidimensional que, após ser processado, forma uma imagem digital. Aí então esta imagem é manipulada através do processamento digital de imagens.

## 1.2 Introdução à Sinais e Sistemas

Primeiramente, definindo os termos chaves dessa seção:

### 1.2.1 Sinais

Em Engenharia elétrica, a medida fundamental para representação de informações é chamada sinal. Um sinal poderia ter quaisquer dimensões e ser de qualquer forma.

- Sinais Analógicos:

Um sinal analógico, medido através do tempo é um sinal contínuo. Eles são difíceis de analisar. Para representá-los, precisaríamos de memória infinita. Isso ocorre pois a amostra é “infinita”. Um exemplo é a voz humana ou  $y = \sin(x)$ .

- Sinais Digitais:

São a apropriação dos sinais analógicos de uma maneira descontínua, valores discretos são utilizados para representar informações. Logicamente são menos precisos que sinais analógicos.

### 1.2.2 Sistemas

Um sistema é definido através dos tipos de sinais de entrada e dos tipos de sinais de saída em relação aos quais o sistema funciona. Nesse caso, um sistema pode ser interpretado como o processo de conversão de um sinal analógico para um sinal digital.

Em tal tipo de conversão, há dois conceitos chave. Amostragem e Quantização. Amostragem pode ser definido como o ato de tirar amostras do sinal isso ocorre no domínio da função que representa o sinal analógico. Já Quantização pode ser definido como o ato de particionar os valores que a função pode assumir.

## 2 Aplicações do Processamento de Imagens Digitais

- Restauração de imagens.
- Análise de imagens médicas.
- Sensoriamento remoto.

- Visão computacional.
- Reconhecimento de padrões.
- Processamento de vídeos.
- Análise de imagens microscópicas.

### 3 Dimensões das Imagens

Primeiramente, o conceito de dimensão define o número mínimo de pontos necessários para saber a posição de um objeto particular no espaço. No caso das imagens, sabemos que são bidimensionais. Altura e Largura. No caso de vídeos, podemos considerar o tempo como uma terceira dimensão e, por isso, podemos classificar esse tipo de imagens (vídeos) como 3D. Desso modo, podemos definir a imagem como uma função de duas variáveis.

$$f(x, y) = \textit{Imagem}$$

É interessante ressaltar que um filme pode ser considerado 3D. E, portanto, na realidade, um filme 3D seria 4D.

### 4 Formação de Imagens em Câmeras

Primeiro, é bom comentar sobre a formação de imagens no olho humano. A luz é refletida pelos objetos e é captada pelos olhos. Ao entrar no olho a imagem fica invertida na retina. Então, o cérebro interpreta a imagem invertida e a “desinverte”.

#### 4.1 Formação de Imagens em Câmeras Analógicas

Nesse tipo de câmera, o filme — uma tira plástica — é revestida com halogeneto de prata. Essa substância reage com os fótons e forma o negativo das imagens fotografadas.

#### 4.2 Formação de Imagens em Câmeras Digitais

Nesse tipo de câmera, a formação das imagens se dá através do uso de um vetor de sensores CCD. CCD significa Charged Couple Devices. Ou seja, dispositivo de carga acoplada, que é um sensor semicondutor que é sensível à imagem e a converte em um sinal elétrico. Há uma matriz retangular

desses sensores, em que cada um dos sensores corresponde à um pixel da imagem formada. Depois de cada sensor na matriz armazenar uma carga a partir da incidência de fótons, tal carga é transformada em uma tensão que é convertida em informação digital. Isso ocorre em um sensor por vez.

## **5 Mecanismos das Câmeras**

### **5.1 Aperture - Abertura**

Como o próprio nome diz, é um orifício que permite a passagem da luz para dentro da câmera. Há lâminas que regulam o tamanho da abertura fazendo com que mais ou menos luz tenha acesso ao interior da câmera. O efeito da abertura da câmera corresponde ao brilho da imagem fotografada

### **5.2 Shutter - Obturador**

Podemos pensar no obturador como uma espécie de cortina que fica depois da abertura da câmera. Por trás do obturador está a matriz de dispositivos de carga acoplada. Assim que o obturador se abre a imagem é formada nessa matriz. Os efeitos do obturador dependem do tempo que a ele permanece aberto. Quanto mais tempo mais clareza haverá na imagem. Nesse sentido, dois conceitos-chaves sobre esse mecanismo são o tempo e a velocidade. “Shutter speed” e “Shutter time”. A velocidade representa quantas vezes o obturador se abriu ou se fechou, já o tempo representa o intervalo de tempo em que ele permanece aberto. É interessante ressaltar que há uma relação inversamente proporcional entre a velocidade e o tempo. Quanto mais velocidade, menor tempo e vice-versa. Uma aplicação importante da velocidade e do tempo do obturador é o ato de capturar imagens que se movimentam em alta velocidade. Assim, quanto menor for o tempo do obturador, melhor conseguiremos capturar imagens que se movem rapidamente.

### **5.3 ISO**

O fator ISO representa a sensibilidade de uma câmera à luz. Quanto maior o número ISO de uma câmera, mais sensível ela é à luz. Contudo, há um sintoma colateral que se pronuncia quando o número ISO é elevado. Nesses casos, há mais ruído na imagem capturada.



## 5.4 Conceito de Pixel

O pixel é a unidade da imagem. Em uma imagem em gray-scale de 8 bits, cada pixel possui um valor de 0 a 255, que corresponde à intensidade da luz naquele ponto. Novamente, cada dispositivo de carga acoplada corresponde à um pixel.

### 5.4.1 Cálculo do número total de pixels

Para saber a quantidade de pixels em uma imagem, basta multiplicar as dimensões da matriz

$$\#pixels = \#colunas \cdot \#linhas$$

### 5.4.2 Valores de pixels

Sabemos que cada pixel possui somente um valor. Além disso, por definição, o valor 0 significa ausência de luz.

## 6 Transformação de Perspectiva

De modo geral, um exemplo claro do que é perspectiva, é o fato de, através dos olhos humanos, ao visualizar objetos menos distantes, esses aparentam ser maiores. Nesse sentido, a transformação de perspectiva é sobre a conversão de uma realidade tridimensional (altura, largura e profundidade) em uma imagem bidimensional (altura e largura).

### 6.1 Referencial - Frame of Reference

#### ACHEI ESSA PARTE CONFUSA

Para analisar o mundo/imagem/cena tridimensional, precisamos de 5 referenciais — “frame of references” — sendo esses:

- Objeto
- Mundo
- Camera
- Imagem
- Pixel

De qualquer maneira, essa transformação pode ser compreendida ao utilizarmos simples conceitos de ótica e trigonometria. Observe: Podemos usar

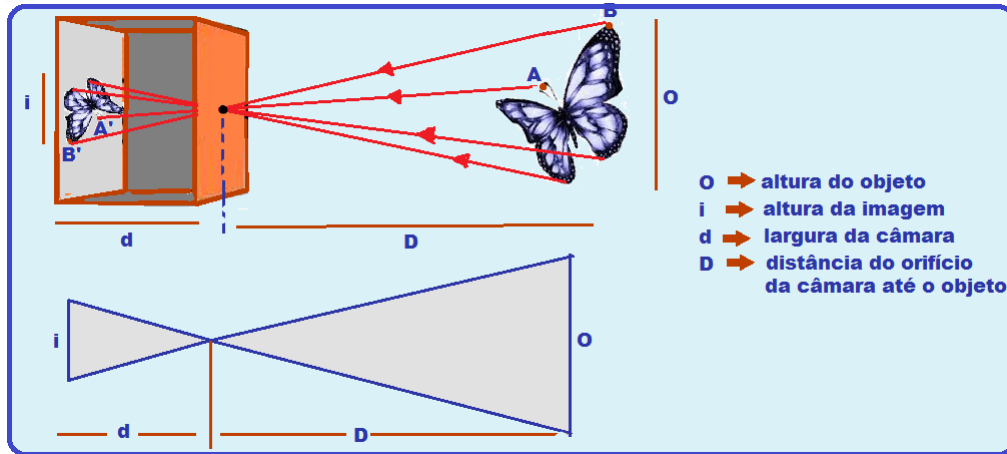


Figura 1: Imagem para exemplificar o uso da trigonometria na conversão do mundo 3D para o 2D

semelhança de triângulos para descobrir o tamanho da imagem formada  $i$ .

$$\frac{i}{d} = \frac{O}{D} \Rightarrow i = d \frac{O}{D}$$

## 7 Conceito de bits por pixel

O número de cores que podem ser representadas em uma imagem depende da quantidade de bits por pixel presente nessa imagem. De maneira geral, podemos dizer que a maioria das imagens em gray-scale consiste de 8 bits por pixel. Equanto as coloridas tem, geralmente 24 *bpp*. Isso pode ser compreendido ao imaginar que cada canal R, G, e B são possuem 8 bits.

Como dito anteriormente, o valor 0 é, por padrão, o valor que representa a cor preta, intensidade mínima em um canal de cor. Agora, o valor que representa o valor máximo não é padronizado, depende do *bpp* dessa imagem. Então, teremos que esse valor pode ser representado por  $2^{bpp} - 1$ .

### 7.1 Cálculo do tamanho (em memória) de uma imagem

Para saber o tamanho de uma imagem, devemos levar em consideração as suas dimensões bidimensionais e o número de *bpp*. Por exemplo, digamos que uma imagem possua 1024 linhas, 1024 colunas e 8*bpp*, teremos:

$$\begin{aligned}
tamanho &= 1024 \cdot 1024 \cdot 8 = 8388608 \text{ bits} \Rightarrow \\
&\Rightarrow tamanho = \frac{8388608}{8} = 1048576 \text{ bytes} \Rightarrow \\
&\Rightarrow tamanho = \frac{1048576}{1024} = 1024 \text{ kb} \Rightarrow tamanho = \frac{1024}{1024} \text{ Mb}
\end{aligned}$$

## 8 Tipos de imagens

### 8.1 Imagem binária

Nesse caso, há somente 2 valores possíveis para os pixels dessa imagem 0 ou 1. Onde 0 representa o preto e 1 o branco.

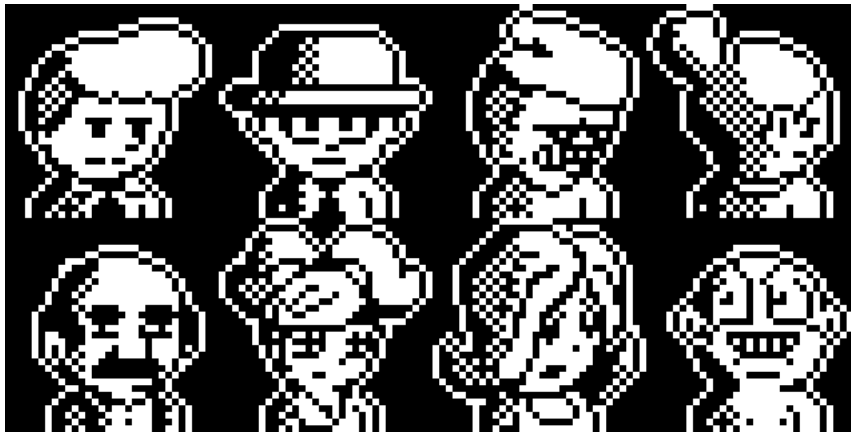


Figura 2: Imagem binária

### 8.2 Formato 8 bits de cor

Comumente conhecida como imagens Grayscale.

### 8.3 Formato 16 bits de cor

Aqui já temos imagens coloridas. Esses 16 bits são distribuídos nos 3 canais R, G e B. Da seguinte maneira:

- 5 bits para o vermelho (R).
- 6 bits para o azul (G).
- 5 bits para o verde (B).



Figura 3: Imagem Grayscale

## 8.4 Formato 24 bits de cor

Por fim, temos o formato de imagens mais comuns hoje em dia. Em que temos esses 24 bits distribuídos igualmente entre os canais R, G e B.



Figura 4: Exemplo de imagem de 24 bits

## 9 Códigos de cores

Consideramos aqui que todas as cores estão no formato de 24bits. ou seja, uma cor “ $C$ ” é tal que  $(R, G, B) = (C_R, C_G, C_B)$  Assim, no formato RGB, temos:

- Branco = (255, 255, 255)

- Preto = (0, 0, 0)
- Vermelho = (255, 0, 0)
- Verde = (0, 255, 0)
- Azul = (0, 0, 255)
- Cinza = (128, 128, 128)

Nesse caso, a cor cinza é sempre o ponto médio entre a intensidade máxima e mínima. Logo, poderíamos escolher o valor de 127 ou de 128. Nesse caso, escolhemos 128 (em todos os canais).

## 9.1 Modelo CMYK

Agora, vale ressaltar o modelo de cores CMYK — comum em impressoras —, aqui, CMYK significa “cyan, magenta, yellow, black”. No conceito de impressoras, geralmente há um cartucho de cores, (CMY) e outro preto. Assim, as cores ciano, magenta e amarelo podem ser formadas ao se variar as porções de vermelho, verde e azul também. Os códigos RGB dessas cores são como segue.

- Magenta = (255, 0, 255)  
Combinação de vermelho e azul.
- Ciano = (0, 255, 255)  
Combinação de verde e azul.
- Amarelo = (255, 255, 0)  
Uma combinação de vermelho e verde.

Além disso, também há o formato hexadecimal para representação de cores. Logo, para converter de um formato para outro, de RGB para hexadecimal ou vice versa, fazemos:

## 9.2 Conversão RGB para Hexadecimal

O formato de uma cor em hexadecimal é tal que #FFFFFF00 (amarelo). para obter esses valores hexadecimais, deve-se, canal a canal RGB de uma dada cor, realizar a divisão do valor do canal por 16. Então, o quociente da divisão do canal R é o primeiro caracter do código, o segundo caracter é o resto dessa divisão. O terceiro valor é o quociente da divisão do valor do canal B por 16, o quarto é o resto dessa divisão. E assim por diante.

Observe:

Sabemos que o código do **amarelo** é (255, 255, 0). Então, sobre R:  $255/16$  tem quociente 15 e resto 15. FF. O mesmo ocorre com G. Agora, sobre B, teremos 00. Formando, assim o código #FFFF00.

### 9.3 Conversão Hexadecimal para RGB

Para fazer o caminho inverso, devemos dividir o código hexadecimal em três porções. Sobre o exemplo do amarelo, temos FF, FF, 00. Agora, em um desses pares por vez, convertemos cada um de seus caracteres para binário. Concatenamos os dois números binários obtidos, convertemos para decimal e, assim, obtemos os valores dos canais. Observe:

Primeiramente, tratando de FF. convertendo para binário, temos 1111 e 1111. Concatenando: 11111111. Convertendo para decimal: 255 (valor do canal R) Repetindo esses passos para os outros dois canais, obtemos o código da cor em (R, G, B).

## 10 Conversão de RGB para Grayscale

Primeiramente a maneira mais simples de se converter uma imagem para Grayscale é fazer a média entre os valores dos canais R, G e B. Observe que



Figura 5: Imagem original.



Figura 6: Imagem Grayscale feita com a média

essa figura não fica com um aspecto muito bom, fica “empretecida”. Isso ocorre pois o olho humano não capta os níveis de vermelho, verde e azul na mesma proporção. Na realidade, o olho humano é muito mais sensível ao verde. Então, há o método de utilizar pesos para corrigir essas imperfeições. Então, se usarmos 30% para o vermelho, 59% para o verde e 11% para o azul, teremos:



Figura 7: Imagem original.



Figura 8: Imagem Grayscale feita utilizando pesos

## 11 Amostragem - Sampling

Como dito anteriormente, o conceito de amostragem se relaciona ao processo de conversão de um sinal analógico para o digital. Assim, tiramos amostras do domínio da função que queremos converter. Logo, podemos correlacionar esse conceito com os pixels da imagem que será formada. Imagine que queremos formar uma imagem de dimensões  $5 \times 5$ . Então, teremos que fazer uma amostragem da altura e largura do mundo que queremos representar em 5 valores diferentes cada. Ou seja, se a imagem é  $f(x, y)$ ,  $x$  assumirá 5 valores e  $y$  assumirá 5 valores também. De maneira análoga, podemos relacionar esse conceito à matriz de dispositivos de carga acoplada “CCDs”.

### 11.1 Diferenças entre Zoom Analógico e Digital

Apesar de ambos os conceitos se relacionarem ao fato de aumentarmos a amostragem do sinal, o zoom analógico aumenta essa amostragem através do movimento das lentes da câmera. Já o zoom digital funciona como se a imagem já houvesse sido capturada. Zoom analógico é feito nos sinais e o digital em uma imagem digital.

## 12 Resolução de pixels

De maneira simples, a resolução de uma imagem pode ser definida como a resolução a partir da quantidade de pixels em uma imagem. Quando sabemos que a resolução de uma imagem é tal que  $m \times n$  isso quer dizer que há  $m$  colunas de pixel (largura) e  $n$  linhas de pixel (altura). Nesse sentido, podemos calcular os MegaPixels de uma imagem fazendo  $\frac{m \cdot n}{1 \text{ mi}}$  além disso, podemos saber o tamanho da imagem a partir de sua resolução também:  $\text{tamanho} = \text{pixelResolution} \cdot \text{bpp}$

## 12.1 Proporção da Tela

Essa proporção nada mais é que a razão entre a largura da imagem e sua altura. Isso é extremamente útil ao mudar o tamanho das imagens, pois, se mantermos essa razão, então a imagem com tamanho modificado terá o mesmo aspecto que a imagem original.

## 13 Mais sobre Zoom

Como introduzimos os conceitos de zoom digital e analógico, podemos compará-los melhor. De fato, como o zoom analógico, ou óptico, nos proporciona um resultado melhor que o zoom digital. Isso ocorre pois, de fato estamos mudando a maneira como a fotografia será tirada, através do movimento das lentes. Agora, o zoom digital, é simplesmente, aumentar uma certa área de visualização de uma imagem, fazendo os pixels ficarem “maiores”. Por isso, a qualidade da imagem é comprometida. Para fazer isso, há alguns métodos possíveis. “Nearest neighbor interpolation”, “Zero order hold method”, “Zooming K times”.

Um adendo interessante, é que o Exercício Programa 2 da disciplina de Laboratório de Métodos Numéricos, ministrada no primeiro semestre de 2022, tratou exatamente deste tópico. Podemos considerar o ato de dar zoom em uma imagem com uma maneira de interpolar pontos, formando novos pixels a partir de pixels dados. Nesse sentido, estudamos os métodos de interpolação polinomial bilinear e bicúbica. Agora, continuemos a tratar dessas outras maneiras de dar zoom, que são abordadas no material.

### 13.1 Nearest Neighbor - Pixel Replication

Para efetuar esse tipo de interpolação, devemos pensar, separadamente nas linhas e nas colunas da matriz que representa a imagem. Digamos que queremos aumentar a imagem por um fator de zoom  $x$ . Assim, assumindo que a imagem inicial tenha dimensões  $3 \times 3$ , por exemplo, e  $x = 2$ , a imagem final terá dimensões  $6 \times 6$ . Agora, seja essa imagem exemplificada pela matriz:

$$\begin{bmatrix} 10 & 4 & 22 \\ 2 & 18 & 7 \\ 9 & 14 & 25 \end{bmatrix}$$



Primeiramente, devemos normalizar a posição dos pixels da nova imagem em relação à imagem antiga, fazemos isso sobre as linhas e sobre as colunas, obtendo uma razão entre o número de linhas da imagem original e o número de linhas da nova imagem. Análogo, para colunas. Então, obtemos duas proporções sobre as dimensões das duas imagens. Agora, multiplicamos os índices de cada elemento da nova imagem por essa proporção. Desse modo, atribuiremos o valor do elemento (na imagem original) que possua índice mais próximo ao que foi calculado. Observe no exemplo, onde queremos interpolar os pontos  $P_1$  e  $P_2$ .

$$\begin{bmatrix} 10 & 4 & 22 \\ 2 & 18 & 7 \\ 9 & 14 & 25 \end{bmatrix} \Rightarrow \begin{bmatrix} P_1 & x & x & x & x & x \\ x & x & x & P_2 & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \\ x & x & x & x & x & x \end{bmatrix} \quad \text{Sabemos que a razão entre as}$$

linhas é:  $\frac{3}{6} = \frac{1}{2}$  e as colunas a mesma. Então, o ponto  $P_1$  tem coordenadas  $(0,0)$ . Fazendo um produto escalar de  $(0,0)$  por  $\left(\frac{1}{2}, \frac{1}{2}\right)$  teremos  $(0,0)$ . O ponto  $P_1$  tem “vizinho mais próximo” o pixel de coordenadas  $(0,0)$  na imagem original.  $P_1 = 10$ . Já em relação à  $P_2$ , teremos:  $(1,3) \cdot \left(\frac{1}{2}, \frac{1}{2}\right) = (0.5, 1.5)$ . Tomando o teto:  $(1,2)$ . O elemento  $P_2$  tem vizinho mais próximo na imagem original o elemento de índice  $(1,2)$  e recebe seu valor.  $P_2 = 7$ . Fazendo o mesmo processo para todos os elementos, teremos a imagem com “zoom”, ou seja, a imagem interpolada.

- Vantagem:  
É um processo extremamente simples.
- Desvantagem:  
A imagem fica com um aspecto borrado.

Além disso, a imagem final terá tamanho tal que  

$$tamanhoFinal = \#linhasDaImagemOriginal \cdot fatorZoom \times \#colunasDaImagemOriginal \cdot fatorZoom$$

## 13.2 Zero-Order Hold - Bloqueio de Ordem Zero

Esse algoritmo de zoom funciona de maneira muito simples. Para obter a imagem com zoom, devemos, colocar, entre os pixels da matriz da imagem

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

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

original, pixels que tenham valores correspondentes à média dos seus vizinhos. Tome o exemplo:

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

Inserindo uma linha entre as linhas dessa matriz, teremos:

Observe que os valores inseridos são o piso das médias obtidas. Agora, inserindo linhas entre as linhas:

Note que ocorreu o mesmo processo. A interpolação, “zoom” está feita.

- Vantagem:  
A imagem não fica com um aspecto borrado.
- Desvantagem:  
Teremos que efetuar zoom nos baseando em potências de 2.

### 13.3 K-times Zooming

Esse algoritmo sana problemas dos que foram discutidos anteriormente.  $K$  significa o fator de zoom. Para explicar esse algoritmo, usaremos um exemplo. Tome a seguinte matriz como representação de uma imagem. Pri-

$$\begin{bmatrix} 15 & 30 & 15 \\ 30 & 15 & 30 \end{bmatrix}$$

meiramente, supohemos  $K = 3$ . O número de valores que devem ser inseridos entre cada coluna e cada linha é  $K - 1 = 3 - 1 = 2$ . Assim, iniciando pelas linhas (inserindo elementos entre as colunas) da matriz, temos que tomar os dois primeiros pixels adjacentes. (15 e 30). Subtraímos 15 de 30 (15). Dividimos o resultado da subtração por  $K$  ( $15/3 = 5$ ). Esse resultado é chamado de OP. Agora devemos adicionar OP ao menor dos dois pixels escolhidos inicialmente ( $5 + 15 = 20$ ) esse será o primeiro valor a ser inserido na matriz. Somamos OP a esse último resultado ( $OP + 20 = 5 + 20 = 25$ ) e esse será o segundo elemento a ser inserido na matriz, note que é o elemento de número

$k - 1 = 2$  que foi inserido entre os dois primeiros elementos da primeira linha da matriz. Devemos fazer isso para todas as colunas. Ao fim desse processo, teremos a matriz:

$$\begin{bmatrix} 15 & 20 & 25 & 30 & 20 & 25 & 15 \\ 30 & 20 & 25 & 15 & 20 & 25 & 30 \end{bmatrix}$$

Depois de fazer isso, devemos ordenar os pixels inseridos de acordo com a ordem estabelecida na matriz. Por exemplo, no trecho:  $[30 \text{ } 20 \text{ } 25 \text{ } 15]$  Tivemos os elementos em azul inseridos. Note que a ordem da matriz era decrescente, da esquerda para a direita, indo de 30 indo até 15. Então, ordenaremos 20 e 25 em ordem decrescente também. Desse modo, Obteremos a seguinte matriz:

$$\begin{bmatrix} 15 & 20 & 25 & 30 & 25 & 20 & 15 \\ 30 & 25 & 20 & 15 & 20 & 25 & 30 \end{bmatrix}$$

Agora, precisamos efetuar o mesmo processo entre as linhas da matriz. Teremos: Observe que usamos o piso da divisão de 5 por 3, e obtemos OP

$$\begin{bmatrix} 15 & 20 & 25 & 30 & 25 & 20 & 15 \\ 20 & 21 & 21 & 20 & 21 & 21 & 20 \\ 25 & 22 & 22 & 25 & 22 & 22 & 25 \\ 30 & 25 & 20 & 15 & 20 & 25 & 30 \end{bmatrix}$$

igual a 1.

**FIQUEI CONFUSO AQUI. O TUTORIAL DIZ QUE A IMAGEM FINAL FICA ASSIM, MAS EU NÃO TERIA QUE ORDENAR OS ELEMENTOS QUE FORAM INSERIDOS NOVAMENTE?**

As dimensões da imagem final serão:  $(K(\#linhas - 1) + 1) \times (K(\#colunas - 1) + 1)$

- Vantagens:

Esse algoritmo combina as qualidades dos dois msotrados anteriormente. Não temos restrições quanto às dimensões da imagem final e a imagem não fica muito borrada.

- Desvantagens:  
Devido à necessidade de ordenação, esse algoritmo fica computacionalmente mais custoso que os demais.

## 14 Resolução Espacial - Spatial Resolution

Esse tipo de resolução se relaciona ao menor detalhe que pode ser identificado em uma imagem. Podemos definir essa resolução também como o número de pixels independentes por polegada. Em suma, esse conceito se refere à comparação de imagens de uma mesma resolução (mesmas dimensões) para saber qual possui mais nitidez. Observe: Podemos medir esse tipo de resolução

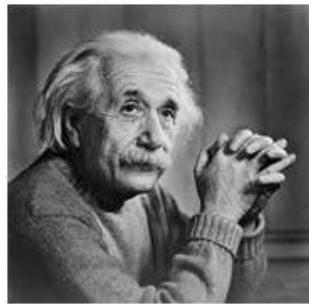


Figura 9: Imagem com maior resolução espacial

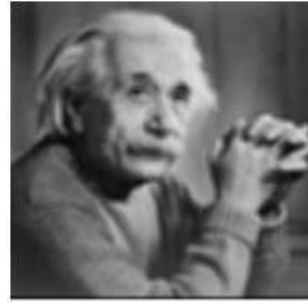


Figura 10: Imagem com menor resolução espacial

das seguintes maneiras:

- Pontos por Polegada.
- Pixels por Polegada.

### 14.1 Pixels por Polegada

Geralmente usada para medir qualidade de telas, sabemos que, para obter essa unidade, devemos, primeiramente, obter

a diagonal em pixels do visor calculada, a partir de sua resolução em pixels.  $c = \sqrt{altura^2 + largura^2}$  Depois disso, dividiremos essa diagonal em pixels pelo tamanho da diagonal.  $PPi = c/diagonalPolegadas$ .

## 14.2 Pontos por Polegada

Essa unidade é, geralmente, utilizada para representar a qualidade de impressoras. Isto é, mede quantos pontos de tinta são impressos por polegada.

## 15 Resolução Grayscale

Resgatando conceitos já tratados anteriormente, a resolução em Grayscale tem íntima relação com a medida de *bpp* (bits por pixel). Ou seja, quanto mais nuances de cinza puderem ser representadas, maior a resolução dessa imagem em termos de grayscale. Ainda, o total de nuances de cinza que podem ser exibidas nessa imagem com  $bpp = k$  é tal que  $2^k$ , mas podemos também representar essa resolução puramente pelo número de *bpp* e não pela quantidade de nuances de cinza que esse número produz.

## 16 Conceito de Quantização

Já sabemos que, no processo de conversão de um sinal analógico para um digital, há os conceitos de amostragem e quantização. Amostragem é feita no domínio do sinal, ou função. Já a quantização, é realizada nos valores que a função pode assumir. Ou seja, na imagem dessa função, feita no eixo *y*. Na realidade, ao quantizar um sinal, estamos digitalizando as amplitudes desse sinal, as particionando. Nesse sentido, podemos interpretar a

resolução grayscale de uma imagem como um exemplo de quantização. Ao limitar os valores de cor que a imagem pode assumir, estamos quantizando as amplitudes do sinal. Então, observe as diferenças entre Imagens com diferentes níveis de resolução grayscale (quantizações distintas).

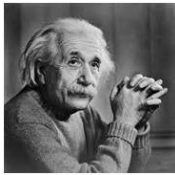


Figura 11:  
256 níveis de  
cinza.

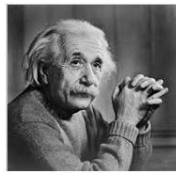


Figura 12:  
128 níveis de  
cinza.

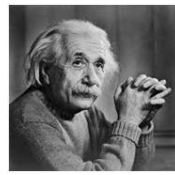


Figura 13:  
64 níveis de  
cinza.

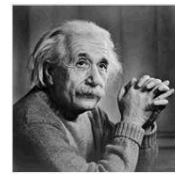


Figura 14:  
32 níveis de  
cinza.

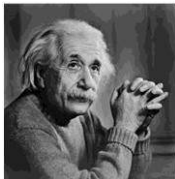


Figura 15:  
16 níveis de  
cinza.

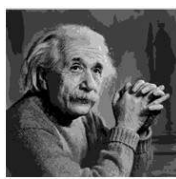


Figura 16:  
8 níveis de  
cinza.

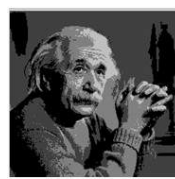


Figura 17:  
4 níveis de  
cinza.

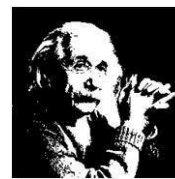


Figura 18:  
níveis de  
cinza.

## 17 Curvas de Preferência ISO - ISO Preference curves

Como podemos observar nas imagens acima, a partir de uma certa quantidade de níveis de cinza, há o aparecimento de linhas na imagem. Tais linhas são chamadas de “countouring”. Note que quanto mais discretizarmos (maior quantização) a imagem, mais aparecem essas linhas. Contudo, esse não é o único fator que influencia no “countouring”.

## 17.1 Curvas de Isopreferência

Em suma, estudos comprovaram que o aparecimento das linhas características do “countouring” está também relacionado à quantidade de detalhes presentes na imagem. Isso quer dizer que quanto mais detalhes uma imagem tiver, mais demorará para o efeito do “countouring” se tornar evidente.

## 18 Conceito de Dithering **ACHEI ESSA SEÇÃO CONFUSA**

Dithering é uma forma de ruído digital aplicado intencionalmente em um sinal. Nesse sentido, podemos “permutar” os pixels de uma imagem com baixos níveis de grayscale utilizando dithering para tornar mais detalhes visíveis nessa imagem.

Primeiro, devemos trabalhar com limiares — thresholding —

## 19 Introdução a Histogramas

### 19.1 Histograma de uma Imagem

O histograma de uma imagem mostra as frequências das intensidades dos pixels. Tomando, novamente, o exemplo de uma imagem grayscale, no eixo  $x$ , teremos as intensidades (níveis) de cinza presentes na imagem, enquanto no eixo  $y$ , teremos a frequência de ocorrência dessas intensidades.

### 19.2 Aplicações dos Histogramas

Uma das principais utilidades dos histogramas é a análise geral de imagens, podemos examinar uma imagem a partir de seu

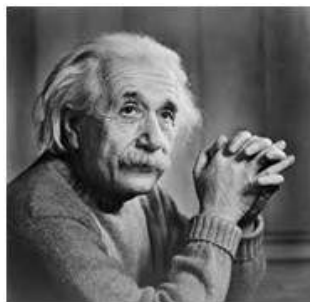


Figura 19: Imagem

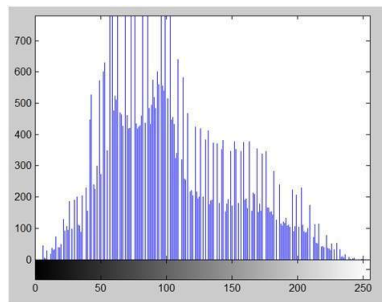


Figura 20: Histograma da Imagem

histograma. Além disso, estudar o brilho e contraste da imagem pode ser feito através do histograma. Por fim, há amplos usos em limiar — *thresholding* —, que são utilizados em visão computacional.

## 20 Brilho e Contraste

Brilho é um conceito relativo. Logo, podemos definir brilho como uma relação entre a quantidade de luz presente na saída (imagem) e a quantidade de luz presente na entrada (mundo real). Para tornar uma imagem mais “clara” (no sentido de possuir mais brilho ou luminosidade), basta adicionar valores à matriz que representa a imagem. Assim — supondo uma imagem grayscale — a cada vez que adicionarmos valores positivos à matriz, estaremos aproximando os valores de seus pixels à 255 (branco).

### 20.1 Contraste

Para definir contraste, simplesmente, podemos dizer que conceito corresponde à diferença máxima (em módulo) entre os valores dos pixels da matriz da imagem. Então, se o valor máximo de um pixel em uma matriz em grayscale for, por exemplo, 200



e o valor mínimo for 0, teremos um contraste de 200 nessa imagem.

## 21 Transformação em imagens

Inicialmente, é válido ressaltar que uma transformação é uma função que leva elementos de um conjunto à outro conjunto. Agora, sendo  $f(x, y)$ , a imagem de entrada,  $G(x, y)$  a imagem de saída e  $T(r)$  a função de transformação que ocorrerá na imagem, temos:  $G(x, y) = T(f(x, y))$ .

Exemplo:

Considerando uma função de binarização da imagem, podemos considerar que, se um pixel ( $f(x, y)$ ) possuir valor maior que 127, deve se tornar branco e, caso contrário, deve se tornar preto. Então, podemos denotar que a função de transformação é tal que:

$$T(f(x, y)) = \begin{cases} 1 & \text{se } f(x, y) > 127 \\ 0 & \text{se } f(x, y) \leq 127 \end{cases}$$

## 22 Deslizamento de Histogramas - Histogram Sliding

O conceito de “delizar” um histograma é, basicamente, mover o histograma de uma imagem para a direita ou para a esquerda. Desse modo, podemos usar esse ato para manipular o brilho de uma imagem. Intuitivamente, se quisermos aumetar o brilho de uma imagem, ao “deslizar” seu histograma para a direita, digamos, em 50 valores, teremos, simplesmente adicionado o valor 50 a todos os pixels da imagem, o que significa, exatmamente, aumentar o brilho da imagem. De maneira análoga, fazemos o

oposto para diminuir o brilho de uma imagem (deslizamos o seu histograma para a esquerda).

## 23 Alongamento de Histogramas - Histogram Stretching

Outra operação que podemos fazer com histogramas, produzem alterações interessantes na imagem é o “alongamento”. Ao fazer isso, alteramos o contraste da imagem. Há diversas abordagens para realizar essa tarefa. Abordaremos com maior detalhe, uma transformação que se chama “Min-Max Stretching”, ou seja esticamento mínimo-máximo.

Nesse tipo de transformação, o valor mínimo de um pixel na imagem é mapeado para o valor 0 e o valor máximo é mapeado para 255. Os valores intermediários serão transformados de acordo com a fórmula:

$$f(x, y)_{novo} = \frac{f(x, y) - f(x, y)_{min}}{f(x, y)_{max} - f(x, y)_{min}} \cdot 255$$

Observe um exemplo dessa transformação:

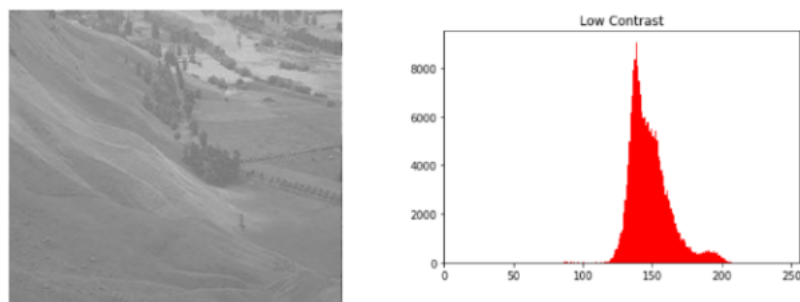


Figura 21: Imagem original

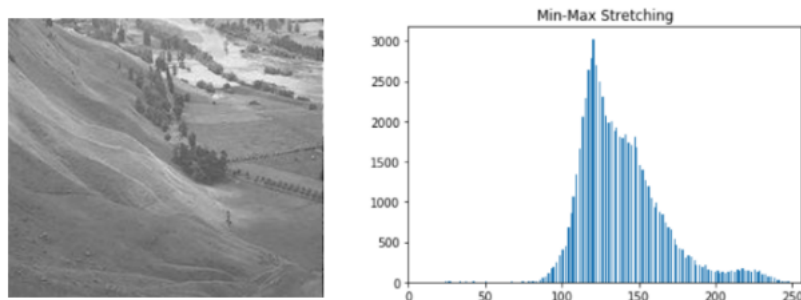


Figura 22: Imagem original

Agora, como nessa maneira de “esticar” o histograma, levamos em consideração, simplesmente o maior e o menor valor, há a possibilidade de que valores outliers modifiquem o resultado final da imagem. Então, se retirarmos as bordas do histograma, ou seja, desconsiderarmos pixels que saiam do padrão de valores presentes na imagem, podemos melhorar o resultado final. Note que, nesse exemplo, há poucos pixels com valores menores que 100, que, certamente atuaram para deixar a imagem transformada mais “esbranquiçada”. Logo, se desconsiderarmos esses pixels, “cortando” o histograma, com certeza teremos um resultado final melhor. Essa técnica é chamada de “Percentile - Stretching”.

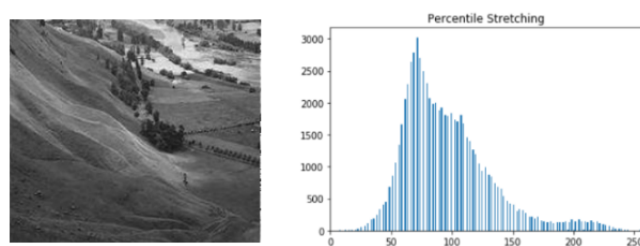


Figura 23: Percentil histogram stretching

## 24 Probabilidade para Transformações de Histogramas

Uma operação muito importante que é efetuada em histogramas de imagens se chama equalização — Histogram Equalization. Dessa maneira, para realizar tal operação, é necessário saber calcular a função de massa de probabilidade e a distribuição de probabilidade acumulada do histograma — ou da própria imagem — em questão. Logo, vamos comentar sobre esses dois conceitos de probabilidade.

### 1. Função de massa de probabilidade:

De maneira geral, essa função associa um valor de probabilidade à uma variável aleatória discreta (nesse caso, os valores dos pixels). Então, supondo a seguinte representação de uma imagem:

$$\begin{bmatrix} 0 & 1 & 50 \\ 2 & 30 & 0 \\ 2 & 1 & 1 \end{bmatrix}$$

Podemos montar a tabela de função de massa de probabilidade:

Valor do pixel	Frequência de aparecimento	$f(x)$
0	2	$\frac{2}{9}$
1	3	$\frac{3}{9}$
2	2	$\frac{2}{9}$
30	1	$\frac{1}{9}$
50	1	$\frac{1}{9}$

De maneira análoga, podemos calcular essa função de probabilidade a partir do histograma de uma imagem. Considere o seguinte histograma:

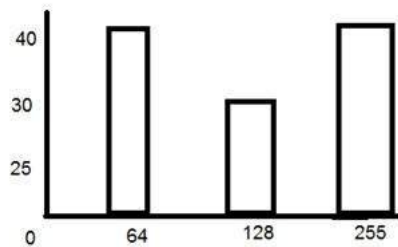


Figura 24: Exemplo de histograma para FMP

Note que o eixo  $y$  desse histograma mostra as frequências de ocorrência dos valores dos pixels. Então, para cada valor que ocorre (64, 128 e 255), basta calcular a razão entre suas ocorrências (frequência) e o total de valores presentes na imagem. Então, para este histograma, teremos a se-

guinte função de massa de probabilidade:

Valor do pixel	Frequência de aparecimento	$f(x)$
64	40	$\frac{40}{110}$
128	30	$\frac{30}{110}$
255	40	$\frac{40}{110}$

## 2. Função distribuição acumulada de probabilidade:

Esse conceito corresponde à soma acumulada das funções massa e probabilidade para cada valor do histograma, ou da imagem. Para exemplificar, é válido observar um gráfico que represente essa função:

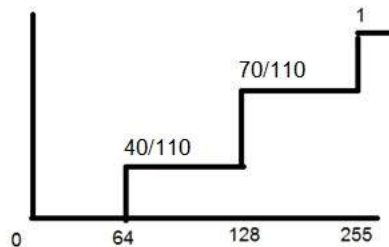


Figura 25: Exemplo de histograma para FMP

Note, que, como a definição deixa claro, essa função corresponde à soma das funções de massa de probabilidade dos valores anteriores ao valor analisado no momento. Ou seja,

$$F(x) = \sum_{xi < x} f(x_i)$$

## 25 Equalização de Histogramas - Histogram Equalization

Usaremos essa técnica de equalização de histogramas para aumentar o contraste de uma imagem. Contudo, em alguns casos, ao efetuar esse processo, o contraste da imagem pode acabar diminuindo.

Desse modo, vamos explicar a partir de um exemplo. Considere que nossa imagem possua os seguintes valores de cinza e as respectivas funções acumuladas de probabilidade:

Valor de cinza	CDF
0	0.11
1	0.22
2	0.55
3	0.66
4	0.77
5	0.88
6	0.99
7	1

Agora, efetuamos a seguinte operação:  $\lfloor CDF \cdot (2^{bpp} - 1) \rfloor$  e adicionar à uma nova coluna, (nesse caso, temos  $bpp = 3 \therefore 2^{bpp} = 8 \therefore 2^{bpp} - 1 = 7$ ) obtendo:

Valor de cinza	CDF	$\lfloor CDF \cdot (2^{bpp} - 1) \rfloor$
0	0.11	0
1	0.22	1
2	0.55	3
3	0.66	4
4	0.77	5
5	0.88	6
6	0.99	6
7	1	7

Esse valor que calculamos será o novo valor que os pixels que possuíam os antigos valores de cinza assumirão. Então, basta

trocar os valores e obteremos uma imagem que teve seu histograma equalizado. Observe as mudanças:

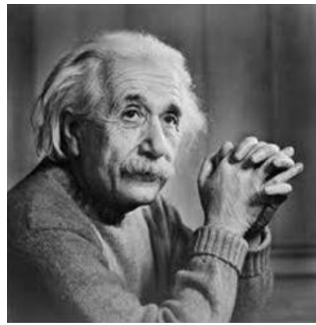


Figura 26: Imagem original

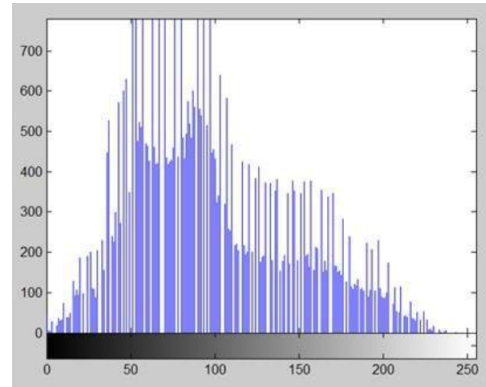


Figura 27: Histograma da Imagem original

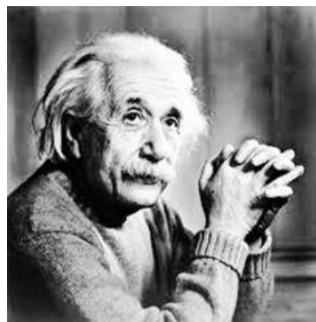


Figura 28: Nova imagem

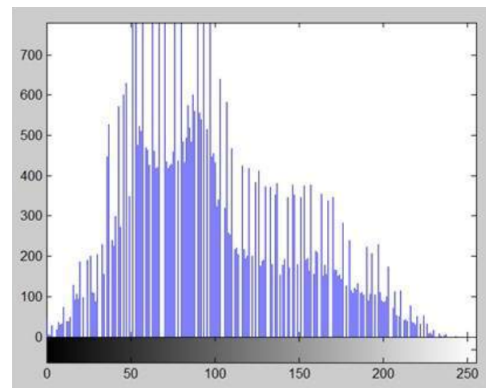


Figura 29: Histograma da nova imagem

Podemos notar que o contraste aumentou na imagem produzida. Além disso, é válido mencionar que, com essa operação, o formato do histograma é distorcido, enquanto os esticamentos e deslizamentos de histograma mantêm a forma original do histograma.



## 26 Transformações de Níveis de Cinza

### 26.1 Aprimoramento de imagens

Nesse contexto, o aprimoramento de imagens assegura um melhor contraste e maior nível de detalhes se comparado à uma imagem “crua”. Isso pode ser feito por meio de transformações dos níveis de cinza dos pixels de uma imagem (grayscale).

### 26.2 Transformação de Nível de Cinza

Há três transformações básicas de níveis de cinza:

- Linear
- Logarítmica
- Power - Law — (Exponencial?)

#### 26.2.1 Transformação Linear

Como o nome diz, os valores de cada pixel na saída da transformação terão uma relação linear com os valores dos pixels de entrada. Nesse sentido, podemos evidenciar duas transformações lineares:

1. Transformação Identidade:

Nesse tipo de transformação, sendo  $g(f(x, y))$  o valor de saída do pixel e  $f(x, y)$  o valor de entrada, teremos:

$$g(f(x, y)) = f(x, y)$$

Ou seja, a imagem de saída é exatamente igual à de entrada.

2. Transformação Negativa:

Nesse tipo de transformação, seguindo os padrões estabelecidos no item imediatamente anterior, teremos:

$$g(f(x, y)) = 2^{bpp} - f(x, y)$$

Note que, se  $f(x, y)$  tem o valor do maior nível de cinza,  $g(x, y)$  será zero e vice-versa. Estaremos produzindo a imagem negativa.

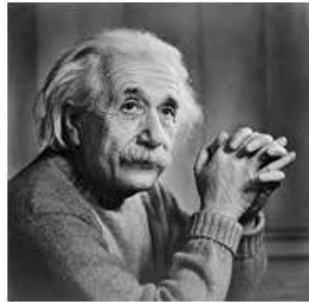


Figura 30: Imagem de entrada



Figura 31: Imagem de saída

### 26.2.2 Transformação Logarítmica

Há duas transformações logarítmicas principais, as logarítmicas de fato e as logarítmicas inversas. Explicaremos sobre as transformações logarítmicas e o inverso valerá para as inversas. Primeiramente, poderemos esperar uma padronização no aspecto da imagem. Isso ocorre devido à forma geral das funções logarítmicas (de acordo com o crescimento dos valores de  $x$ , os valores de  $f(x, y)$  vão se aproximando).

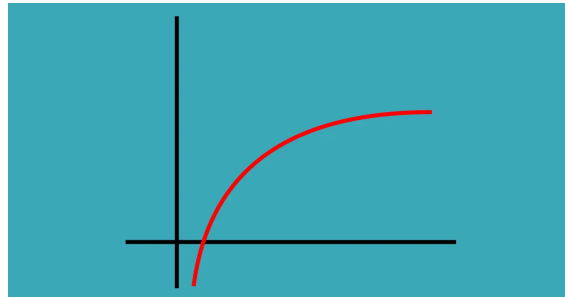


Figura 32: Exemplo de gráfico de função log

Portanto, sendo a transformação logarítmica definida por:

$$g(f(x, y)) = c \log (f(x, y) + 1)$$

Os pixels com valores mais altos terão seus valores aproximados entre si e os pixels que sofrerão desse efeito podem ser controlados ao manipularmos a constante  $c$ . Observe que somamos 1, para não termos problemas em relação ao log dos valores 0 de pixels.

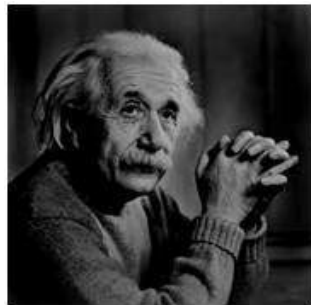


Figura 33: Imagem de entrada



Figura 34: Imagem de saída

### 26.2.3 Transformação Gamma - Power Law - Exponencial

Nesse caso, também há a transformações de exponenciação e de radiciação. Essa transformação é representada a partir da fórmula:

**AQUI A TRANSFORMAÇÃO GAMMA É**

$$g(f(x, y)) = c \cdot f(x, y)^\gamma \text{ OU } g(f(x, y)) = c \cdot f(x, y)^{\frac{1}{\gamma}} ???$$

$$g(f(x, y)) = c \cdot f(x, y)^{\frac{1}{\gamma}}$$

Observe exemplos dessas transformações:

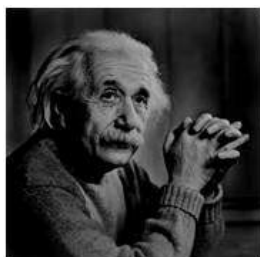


Figura 35:  $\gamma = 10$

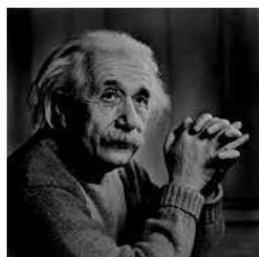


Figura 36:  $\gamma = 8$



Figura 37:  $\gamma = 6$

## 27 Conceito de Convolução

Convolução é uma terceira maneira de manipular imagens. Já abordamos transformações em histogramas e funções de transformação grayscale. Matematicamente, convolução pode ser representado como:

$$g(x, y) = h(x, y) * f(x, y)$$

Podemos dizer que a “máscara” envolveu a imagem. Por outro, lado, devido à comutatividade dessa operação, podemos escrever:

$$g(x, y) = f(x, y) * h(x, y)$$

Que pode se explicado dizendo que a imagem foi envolvida pela “máscara”.  $h(x, y)$  é a máscara, ou filtro.

### 27.1 Máscara

A máscara pode ser interpretada, também, como um sinal. Representada através de uma matriz bidimensional, a máscara deve ser quadrada e ter dimensões ímpares.

#### 27.1.1 Como efetuar convolução utilizando a máscara

Seja a seguinte matriz uma representação de nossa máscara:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Primeiro, devemos espelhar a matriz horizontalmente:

$$\begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{bmatrix}$$

Segundo, devemos espelhar a matriz verticalmente:

$$\begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix}$$

Agora, vamos considerar que nossa imagem seja representada pela seguinte matriz:

$$\begin{bmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \\ 14 & 16 & 18 \end{bmatrix}$$

Por fim, para efetuar a convolução, devemos pegar a máscara, guiada pelo seu elemento central, e “colocá-la” sobre cada elemento da imagem. Ao fazer isso, devemos multiplicar os elementos que se sobrepõem e efetuar a soma dessas multiplicações. Depois disso, o resultado obtido deve ser colocado na posição que estamos analisando. Por exemplo: Ao colocarmos a máscara sobre o elemento de índice  $(0, 0)$  da imagem — nesse exemplo —, os elementos que se sobreporiam seriam os seguintes: 5 sobre o 2, 4 sobre 4, 1 sobre o 10 e 2 sobre o 8. Efetuando as multiplicações e as somas teríamos:  $5 \cdot 2 + 4 \cdot 4 + 1 \cdot 10 + 2 \cdot 8 = 10 + 16 + 10 + 16 = 52$ . O valor 52 deve ser colocado na posição do índice em questão.

### 27.1.2 Vantagens da Convolução

Com convolução podemos realizar feitos que não são possíveis com os demais métodos comentados. Por exemplo:

- Redução de ruídos
- Detecção de bordas
- blurring
- Sharpening

## 28 Conceito de Máscara

Uma máscara nada mais é que um filtro que é aplicado na imagem, de ponto a ponto, como mostrado na seção anterior. A cada ponto, a resposta do filtro é calculada a partir de uma relação pré-definida. De forma geral, há dois tipos principais de filtros: filtros lineares e filtros de domínio de frequência — frequency domain filters. Os usos mais comuns dos filtros são: blurring, sharpening, detecção de bordas e redução de ruídos.

Redução de ruído e blurring estão relacionados. Para “borrar” uma imagem, usamos máscaras para fazer com que, haja uma transição suave entre pixels de valores diferentes. Desse modo, a redução de ruído é feita utilizando blurring. Além disso, as máscaras são usadas para detectar e deixar as bordas mais “afiadas” (sharpening). Observe uma imagem após a aplicação desse efeito:

## 29 Conceito de Blurring

Para borrar ou esfumacar uma imagem, comumente utilizamos os seguintes tipos de filtros: filtro de média, filtro de média

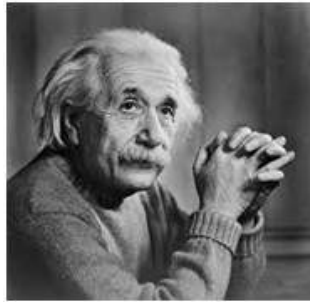


Figura 38: Imagem de entrada



Figura 39: Imagem de saída

ponderada e filtro gaussiano. Focaremos nos dois primeiros.

#### 1. Filtro de Média:

Aqui, a ideia é realizar fazer com que o elemento sobre o qual aplicaremos o filtro tenha o valor da média aritmética de sua vizinhança. para fazer isso, produzimos uma máscara de dimensões  $n \times n$  com  $n$  ímpar tal que, todos os elementos sejam iguais e sua soma seja igual à 1. Dessa maneira, ao efetuarmos a convolução, digamos em um pixel no meio da imagem, esse pixel será equivalente ao valor da média aritmética dos  $n^2$  elementos mais próximos a ele (incluindo ele próprio). Observe algumas imagens que foram submetidas à esse processo:

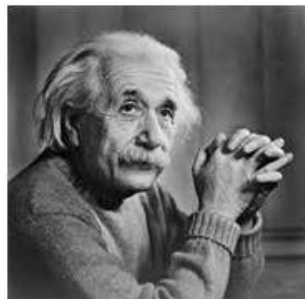


Figura 40: Imagem original

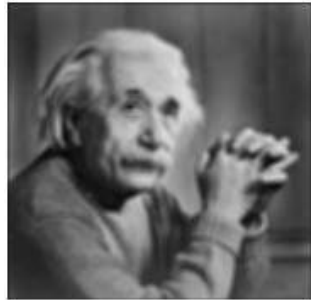


Figura 41: Filtro  $5 \times 5$

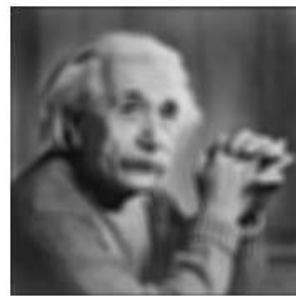


Figura 42: Filtro  $7 \times 7$



Figura 43: Filtro  $9 \times 9$



Figura 44: Filtro  $11 \times 11$

## 2. Filtro de Média Ponderada:

Nesse caso, a ideia é parecida com a mostrada no item anterior, contudo, teremos um peso maior associado aos elementos centrais, certificando que as propriedades da soma dos elementos na matriz ser 1,  $n$  deve ser ímpar e o peso do elemento central deve ser o maior de todos.

## 30 Conceito de bordas - Edges

Podemos definir “bordas” como mudanças bruscas em valores de pixels adjacentes em uma imagem. É interessante ressaltar que estudar como detectar e aprimorar as bordas de uma imagem é importante, pois a maior parte da informação das imagens está presente em suas bordas. Então, fazendo isso, podemos fazer a imagem ficar mais nítida. Observe um exemplo dessa



transformação:

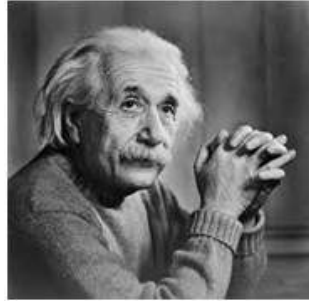


Figura 45: Imagem original

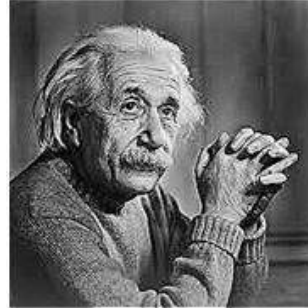


Figura 46: Imagem com o filtro

Nesse sentido, podemos ressaltar os seguintes tipos de bordas: horizontais, verticais e diagonais.

Para fazer isso, são utilizados alguns tipos de filtro:

- Prewitt Operator
- Sobel Operator
- Robinson Compass Masks
- Krisch Compass Masks
- Laplacian Operator

### 30.1 Prewitt Operator

Essa maneira de encontrar bordas em uma imagem é capaz de detectar bordas horizontais e verticais. Essa, assim como as outras máscaras utilizadas para encontrar bordas em imagens, tem seu funcionamento baseado em diferenciação. Nesse sentido, as “máscaras derivativas” devem seguir as seguintes propriedades: Valores de sinais opostos devem estar presentes na máscara, a soma dos valores deve ser igual a zero e quanto mais peso, mais

detecção.

Primeiramente, vamos observar as máscaras que esse método nos fornece. Para detecção na direção vertical, a máscara é a seguinte:

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

Para entender o funcionamento dessa máscara, pensemos em um exemplo em que a máscara será aplicada no ponto  $(x_i, y_j)$  da seguinte imagem/matriz:

$$\begin{bmatrix} f(x_{i-1}, y_{j-1}) & f(x_{i-1}, y_j) & f(x_{i-1}, y_{j+1}) \\ f(x_i, y_{j-1}) & f(x_i, y_j) & f(x_i, y_{j+1}) \\ f(x_{i+1}, y_{j-1}) & f(x_{i+1}, y_j) & f(x_{i+1}, y_{j+1}) \end{bmatrix}$$

Ao aplicarmos essa máscara no ponto referido,  $f(x_i, y_j)$  assumirá o valor de:

$$(f(x_i, y_{j+1}) - f(x_i, y_{j-1})) + (f(x_{i-1}, y_{j+1}) - f(x_{i-1}, y_{j-1})) + (f(x_{i+1}, y_{j+1}) - f(x_{i+1}, y_{j-1}))$$

Desse modo, como o valor em um pixel será a soma das diferenças (em relação à vertical) dos pixels ao seu redor, se essas diferenças forem grandes, o valor do pixel será alto, caso contrário, esse valor será baixo. Cumprindo assim, o papel de aumentar o contraste nas bordas verticais.

O mesmo princípio é utilizado na máscara para detectar as bordas horizontais. A máscara utilizada é:

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

Observe esse método funcionando:

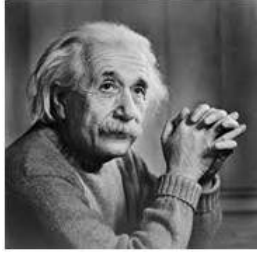


Figura 47: Imagem original



Figura 48: Detecção de bordas verticais



Figura 49: Detecção de bordas horizontais

## 30.2 Sobel Operator

Esse método de realizar detecção de bordas é extremamente similar ao de Prewitt. De maneira análoga, ele é capaz de identificar bordas verticais e horizontais. A diferença entre eles é a atribuição de pesos diferentes nesse caso. Contanto que a máscara formada não viole as propriedades das máscaras derivativas. Um exemplo desse tipo de máscara é:

$$\text{Vertical: } \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}; \text{ Horizontal: } \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Observe esse método em funcionamento:

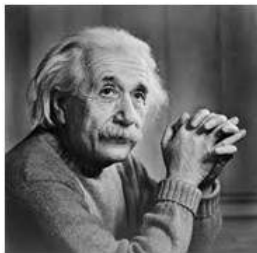


Figura 50: Imagem original



Figura 51: Detecção de bordas verticais



Figura 52: Detecção de bordas horizontais

### 30.3 Robinson Compass Masks

Tratando de mais um tipo de máscara derivativa para detectar bordas de imagens, podemos identificar esta como máscara de direção. Isso ocorre pois a ideia desta máscara é utilizar uma máscara como as mostradas anteriormente e, simplesmente, rotacioná-la nas 8 direções de uma bússola. Para fazer isso, podemos utilizar qualquer máscara derivativa, observe:

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

Máscara de  
direção Norte

Máscara de  
direção Noroeste

Máscara de  
direção Oeste

Máscara de  
direção Sudoeste

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

Máscara de  
direção Sul

Máscara de  
direção Sudeste

Máscara de  
direção Leste

Máscara de  
direção Nordeste

Disso, temos os seguintes resultados:



Figura 53:  
Direção  
Norte



Figura 54:  
Direção No-  
roeste



Figura 55:  
Direção  
Oeste



Figura 56:  
Direção Su-  
doeste



Figura 57:  
Direção Sul



Figura 58:  
Direção Su-  
deste



Figura 59:  
Direção  
Leste

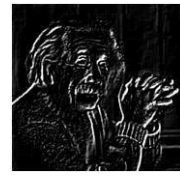


Figura 60:  
Direção Nor-  
deste

### 30.4 Krisch Compass Masks

Esse método é muito similar ao de Robinson. A diferença, é que já possuímos valores da máscara que será rotacionada:

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

Máscara de  
direção Norte

Máscara de  
direção Noroeste

Máscara de  
direção Oeste

Máscara de  
direção Sudoeste

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

Máscara de  
direção Sul

Máscara de  
direção Sudeste

Máscara de  
direção Leste

Máscara de  
direção Nordeste

Observe os resultados em uma imagem real:



Figura 61:  
Direção  
Norte



Figura 62:  
Direção No-  
roeste



Figura 63:  
Direção  
Oeste



Figura 64:  
Direção Su-  
doeste



Figura 65:  
Direção Sul



Figura 66:  
Direção Su-  
deste



Figura 67:  
Direção  
Leste



Figura 68:  
Direção Nor-  
deste

### 30.5 Laplacian Operator

Por fim, temos o método de Laplace para encontrar bordas de imagens. Diferentemente dos outros, que eram métodos que se baseavam em primeiras derivadas, este utiliza uma máscara derivativa de segunda ordem. Nesse sentido, temos somente dois tipos de máscaras de Laplace: Positiva (detecta bordas externas) e Negativa (detecta bordas internas). Primeiro, observe essas duas máscaras de Laplace:

$$\text{Positiva: } \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}; \text{ Negativa: } \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Um detalhe importante, é que, para obter o resultado esperado, não basta realizar a convolução na imagem utilizando essa máscara. Na realidade, se aplicarmos a máscara positiva em uma imagem, teremos de subtrair a imagem resultante da imagem original. Já se aplicarmos a máscara negativa, teremos de adicionar a imagem resultante à imagem original.

Observe os resultados:

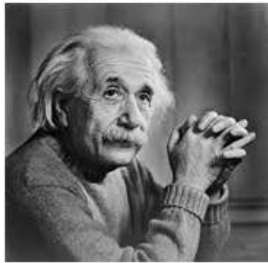


Figura 69: Imagem original



Figura 70: Detecção de bordas externas



Figura 71: Detecção de bordas internas

## 31 Análise do Frequência de Domínio - Frequency Domain Analysis

Até agora, analisamos os sinais obtidos em respeito do tempo ou do espaço. Ou seja, geralmente, o tempo ou a posição são as variáveis independentes. Nesse caso, a frequência se torna o domínio da função. Agora lidaremos com a medida de mudança de valores dos pixels. Nesse sentido, o processo de transformação na imagem segue os seguintes passos:

- Recebemos uma imagem como entrada.
- Convertemos essa imagem para sua distribuição de frequências.
- Processamos essa distribuição.
- Convertemos a nova distribuição em uma nova imagem.
- Entregamos a imagem de saída.

**DIFÍCIL NÃO ENTENDI AS APLICAÇÕES E ACHEI MAL EXPLICADO**

## 32 Series e Transformadas de Fourier

### 33 Series de Fourier

A ideia das series de Fourier é simplesmente que um sinal pode ser decomposto como uma soma infinita de senos e cossenos multiplicados por constantes. Assim, ao tratarmos sobre Transformadas de Fourier desejamos decompôr o sinal (imagem) em senos e cossenos, e a saída dessa transformada é a representação da imagem cujo domínio são as frequências. Faremos uma breve explicação sobre transformadas de Fourier discretas (DFT) — a aplicação em imagens digitais, devido à quantização e amostragem.

Assim, o número de frequências corresponderá ao número de pixels. Desse modo, para uma imagem quadrada  $N \times N$ , a transformada discreta de Fourier, que converte uma imagem para a sua representação que possui frequência como domínio, é dada por:

$$F(u, v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi \left( \frac{ux}{N} + \frac{vy}{N} \right)}$$

A transformada inversa de Fourier, que converte a imagem à sua forma original é dada por:

$$f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} F(u, v) e^{j2\pi \left( \frac{ux}{N} + \frac{vy}{N} \right)}$$

ACHEI MAL EXPLICADO



## 34 Teorema da Convulsão

O teorema da convulsão serve para relacionar uma imagem à sua representação no domínio das frequências. Observe a representação gráfica dessas duas interpretações de uma imagem:

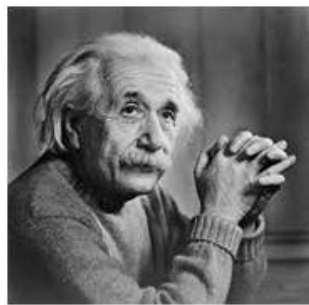


Figura 72: Imagem original

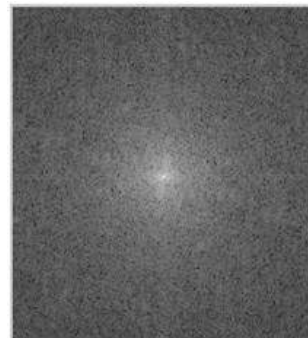


Figura 73: Imagem com o filtro

O Teorema da Convulsão pode ser representado como:

$$\begin{aligned}f(x, y) * h(x, y) &\Leftrightarrow F(u, v)H(u, v) \\f(x, y)h(x, y) &\Leftrightarrow F(u, v) * H(u, v) \\h(x, y) &\Leftrightarrow H(u, v)\end{aligned}$$

Ou seja, realizar convulsão tendo o espaço como domínio é equivalente à filtrar tendo as frequências como domínio, e vice-versa.

## 35 Filtros Passa Alta e Filtros Passa Baixa em Imagens

## 36 Introdução aos Espaços de Cor

Alguns dos padrões de cores utilizados atualmente são:

- RGB

- CMYK
- YUV
- YIQ
- YCbCr

Sobre RGB, sabemos que é um modelo de cor extremamente utilizado. A representação das imagens utilizando esse formato passa a consistir em uma matriz 3D, em que, cada pixel possui 3 valores — R, G e B —, ou, também podemos interpretar como 3 matrizes distintas.

No modelo CMYK, já foi abordado que podemos tratar, separadamente CMY e K, e que esse é um padrão comumente utilizado em impressoras. Para converter RGB para CMY, fazemos:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 2^{bpp} - 1 \\ 2^{bpp} - 1 \\ 2^{bpp} - 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Sobre YUV, podemos dizer que é utilizado nos seguintes padrões de cor:

- NTSC (National Television System Committee)
- PAL (Phase Alternating Line)
- SECAM (Sequential couleur a amemoire, French for “sequential color with memory”)

Já em relação ao YCbCr, é interessante ressaltar que é utilizado, principalmente, na compressão de JPEG e MPEG.

## **37 Introdução à Compressão de JPEG**

## **38 Reconhecimento Ótico de Caracteres - OCR**

OCR consiste na conversão de textos (manuscritos, escaneados, fotografados) para textos compreensíveis para o computador. OCR revolucionou os mais diversos tipos de indústrias ao redor do mundo, principalmente na questão de documentos. Entre diversas aplicações importantes dessa tecnologia, podemos ressaltar as seguintes:

- Bancos (principalmente em relação à cheques)
- Auxílio à pessoas com deficiências visuais  
Com OCR, existe a possibilidade de que o computador efetue leitura convencional (sem ser braile) para essas pessoas.
- Vendas — Reconhecimento de caracteres em códigos de barra

## **39 Visão Computacional**

Visão Computacional busca replicar a visão humana utilizando computadores. Há uma hierarquia no estudo de visão computacional:

- Baixo nível:  
Inclui processamento da imagem para futura extração
- Nível Intermediário:  
Inclui reconhecimento de objetos e interpretação de cenas
- Nível alto:  
Inclui descrição conceitual de cenas.

Alguns dos incontáveis exemplos de aplicação dessa área de estudo são:

- Robótica:  
Carros inteligentes que desviam de objetos, interação entre robôs e humanos, movimentação de robôs utilizando visão computacional e inteligência artificial.
- Medicina:  
Classificação de doenças usando imagens médicas, cirurgias guiadas por robôs, construção de representação de órgãos para realizar exames.
- Indústria:  
Reconhecimento de padrões como OCR, ordenação de objetos utilizando visão computacional, interpretação de documentos.

## 40 Computação Gráfica

Esse tipo de computação foca na criação, manipulação e renderização de objetos e suas imagens relacionadas. Depois do crescimento da indústria dos vídeo games, essa área se expandiu consideravelmente. Nesse sentido, as tarefas relacionadas à ela são realizadas, geralmente, com o auxílio de hardware e software especializados. Um exemplo claro disso são as poderosíssimas placas de vídeo que surgem ano após ano. Alguns exemplo de atuação dessas tecnologias são:

- Design  
Utilizado para fazer modelos 3D e realistas de edifícios, carros, aviões, etc.
- Animação 3D e Vídeo games
- Educação  
Utilizado para gerar modelos de corpos para auxiliar no estudo da medicina, por exemplo. Ou simulações para pilotos de avião.