

CIÊNCIAS DA COMPUTAÇÃO

Processamento de Imagens e Visão Computacional

Prof. César C. Xavier

Processamento de Imagens e Visão Computacional

ROTEIRO

Pré-Processamento

- Introdução
- Operações Básicas
 - Acessando e Modificando pixels de imagens
 - Dados das Imagens
- Histograma
 - Imagens Binárias
 - Imagens tons cinza
 - Imagens coloridas
- Equalização Histograma
- Práticas Python – Ao longo do Conteúdo

Prof. César C. Xavier

Introdução

Pré-Processamento

- Responsável por destacar (ressaltar) objetos de interesse em uma imagem.
- Visa facilitar o processo de segmentação a ser realizado em etapa posterior.
- Métodos:
 - Operações Aritméticas;
 - Operações Geométricas;
 - Ajuste Contraste; e
 - Redução de Ruído.

Prof. César C. Xavier

Operações Básicas

- Obtenção valores de cor de um pixel.

```
import cv2
```

```
#
```

```
# Extrai a cor do pixel na L,C=150
```

```
#
```

```
imagem = cv2.imread("frutas.jpg")
```

```
valorPixel = imagem[150,150]
```

```
print(valorPixel)
```

Prof. César C. Xavier

Operações Básicas

- Utilizando flags para a função imread.

Enumerations

```
enum cv::ImreadModes {  
    cv::IMREAD_UNCHANGED = -1,  
    cv::IMREAD_GRAYSCALE = 0,  
    cv::IMREAD_COLOR = 1,  
    cv::IMREAD_ANYDEPTH = 2,  
    cv::IMREAD_ANYCOLOR = 4,  
    cv::IMREAD_LOAD_GDAL = 8,  
    cv::IMREAD_REDUCED_GRAYSCALE_2 = 16,  
    cv::IMREAD_REDUCED_COLOR_2 = 17,  
    cv::IMREAD_REDUCED_GRAYSCALE_4 = 32,  
    cv::IMREAD_REDUCED_COLOR_4 = 33,  
    cv::IMREAD_REDUCED_GRAYSCALE_8 = 64,  
    cv::IMREAD_REDUCED_COLOR_8 = 65,  
    cv::IMREAD_IGNORE_ORIENTATION = 128  
}
```

Prof. César C. Xavier

Operações Básicas

- Exemplo imread com flag.

```
#  
# Importando Imagem Colorida diretamente  
# em Tons de Cinza e apresentando os valores para L,C=150  
#  
  
imagem1a = cv2.imread("frutas.jpg",cv2.IMREAD_GRAYSCALE)  
valorPixel1a = imagem1a[150,150]  
print("Imagem Importada em Tom Cinza:",valorPixel1a)
```

Prof. César C. Xavier

Operações Básicas

- Obtenção valores de tom cinza:

```
import cv2

#
# Extrai a cor do pixel na L,C=150
#

imagem = cv2.imread("frutas.jpg")
imagem = cv2.cvtColor(imagem, cv2.COLOR_BGR2GRAY)
valorPixel = imagem[150,150]
print(valorPixel)
```

Prof. César C. Xavier

Operações Básicas

- Obtenção valor de um canal de cor BGR de um pixel.

```
import cv2

#
# Extrai a cor do pixel na L,C=150
#

imagem = cv2.imread("frutas.jpg")
valorPixel = imagem[150,150,0]
print(valorPixel)
```

Prof. César C. Xavier

Operações Básicas

- Alteração do valor de cor BGR de um pixel.

```
import cv2

#
# Extrai a cor do pixel na L,C=150
#

imagem = cv2.imread("frutas.jpg")
valorPixel = imagem[150,150]
print("Valor Original Pixel (150, 150):"valorPixel)
Imagem[150,150] = [255, 255, 255] // intensidades B, G e R.
print("Valor Alterado Pixel (150, 150):"valorPixel)
```

Prof. César C. Xavier

Histograma

› Definição:

O *histograma* de uma imagem digital com níveis de intensidade no intervalo $[0, L - 1]$ é uma função discreta $h(r_k) = n_k$, onde r_k é o k -ésimo valor de intensidade e n_k é o número de pixels da imagem com intensidade r_k .

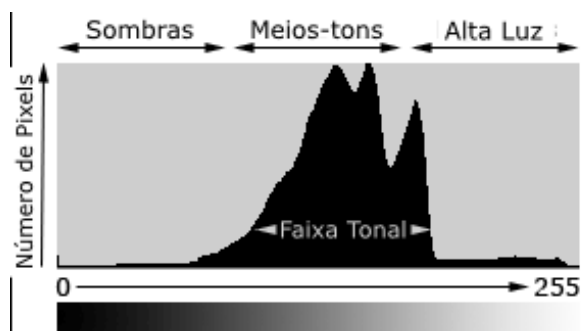


Figura – Exemplo típico de um histograma considerando seu espectro de 255 tons de intensidades de cinza.
<https://www.cambridgecolour.com/tutorials/histograms1.htm>

Prof. César C. Xavier

Histograma

> Definição:

O *histograma* de uma imagem digital com níveis de intensidade no intervalo $[0, L - 1]$ é uma função discreta $h(r_k) = n_k$, onde r_k é o k -ésimo valor de intensidade e n_k é o número de pixels da imagem com intensidade r_k .

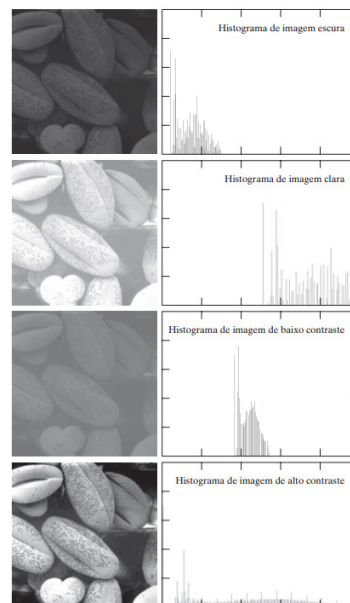


Figura 3.16 Quatro tipos básicos de imagem: escura, clara, baixo contraste, alto contraste e seus histogramas correspondentes.

Prof. César C. Xavier

Histograma

> Diferentes Espaços de Cores



(a)



(b)



(c)

Espaço de Cor (a) binária, (b) cinza e (c) RGB.

Prof. César C. Xavier

Histograma

- Imagem Binária

- Pixels da imagem são formados apenas pela cor preta ou branca.
- Total de pixels da imagem subtraído do total de pixels de uma determinada cor nos dá a quantidade de pixels branco ou pretos representados.

Prof. César C. Xavier

Imagem Binária

```
import cv2

#
# Conta o número de pixel nas cores 0 (preto) e 255 (branco)
#

imagem = cv2.imread("folha_binaria.bmp", 0)
print("altura:", imagem.shape[0], "largura:", imagem.shape[1])
totalPixelsPreto = 0;
totalPixelsBranco=0;
For y in range(0, imagem.shape[0]):
    For x in range(0, imagem.shape[1]):
        if imagem[y,x] == 255:
            totalPixelsBranco += 1;
        else:
            totalPixelsPreto += 1;
print(totalPixelsBranco)
print(totalPixelsPreto)
```

Prof. César C. Xavier

Processamento de Imagens e Visão Computacional

Histograma

- Imagem Binária: função *hist* da biblioteca Matplotlib

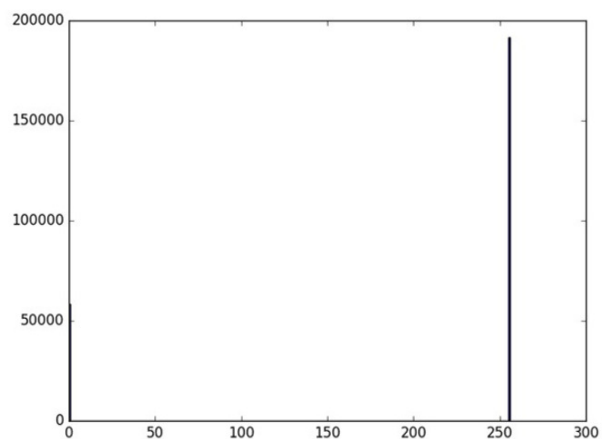
```
import cv2
import numpy as np
from matplotlib import pyplot as grafico
imagem = cv2.imread("folha_binaria.bmp", 0) # 0: escala cinza
grafico.hist(imagem.ravel(), 256, [0,256])
    # ravel(): transforma a matriz multidimensional da imagem em um array
    # unidimensional
    # 256: número de bins
    # [0, 256]: define o intervalo dos valores considerados
grafico.show()
```

Prof. César C. Xavier

Processamento de Imagens e Visão Computacional

Histograma

Imagem Binária



Prof. César C. Xavier

Histograma

- Imagem Tons de Cinza
 - Pixels da imagem são formados no intervalo [0, 255].
 - Pode conter no máximo 256 faixas.
 - Maior o número de faixas maior o número de informações que se pode extrair da imagem.

Prof. César C. Xavier

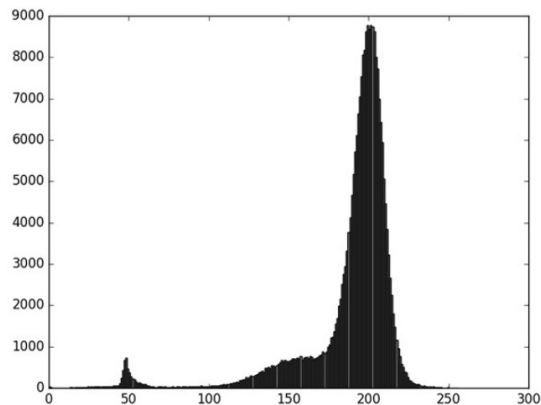
Imagem em Tons de Cinza:

```
import cv2
import numpy as np
import matplotlib.pyplot as grafico
#
# Conta o número de pixel nas cores 0 (preto) e 255 (branco)
#
imagem = cv2.imread("folha_tons_cinza.jpg",cv2.IMREAD_GRAYSCALE)
grafico.hist(imagem.ravel(), 256, [0,255])
grafico.title("Histograma Tons Cinza")
grafico.xlabel("Intensidade")
grafico.ylabel("QTD")
grafico.text(50, 25000, "folha_tons_cinza.jpg")
grafico.xlim(0,255)
grafico.grid(True)
cv2.imshow("Folha Tons Cinza", imagem)
grafico.show()
```

Prof. César C. Xavier

Histograma

Imagem em Tons de Cinza



Prof. César C. Xavier

Histograma

- Imagem Colorida
 - Possuirá um histograma para cada canal, individualmente.
 - Para imagem RGB de 8 bits, pode-se ter no máximo 256 faixas.
 - Maior o número de faixas maior o número de informações que se pode extrair da imagem.

Prof. César C. Xavier

Histograma

Imagem Colorida:

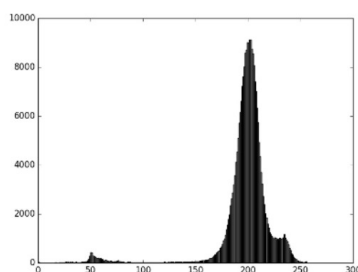
```
import cv2
import numpy as np
from matplotlib import pyplot as grafico

imagem = cv2.imread("folha_colorida.jpg")
azul, verde, vermelho = cv2.split(imagem)
grafico.hist(azul.ravel(), 256, [0,256])
grafico.figure();
grafico.hist(verde.ravel(), 256, [0,256])
grafico.figure();
grafico.hist(vermelho.ravel(), 256, [0,256])
grafico.show()
```

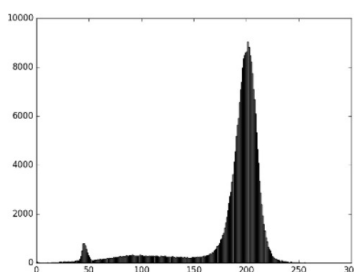
Prof. César C. Xavier

Histograma

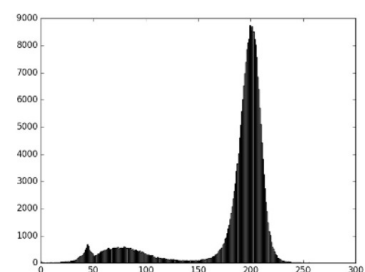
Imagem Colorida



(a) canal azul



(b) canal verde



(c) canal vermelho

Prof. César C. Xavier

Processamento de Imagens e Visão Computacional

Histograma

Imagem Colorida:

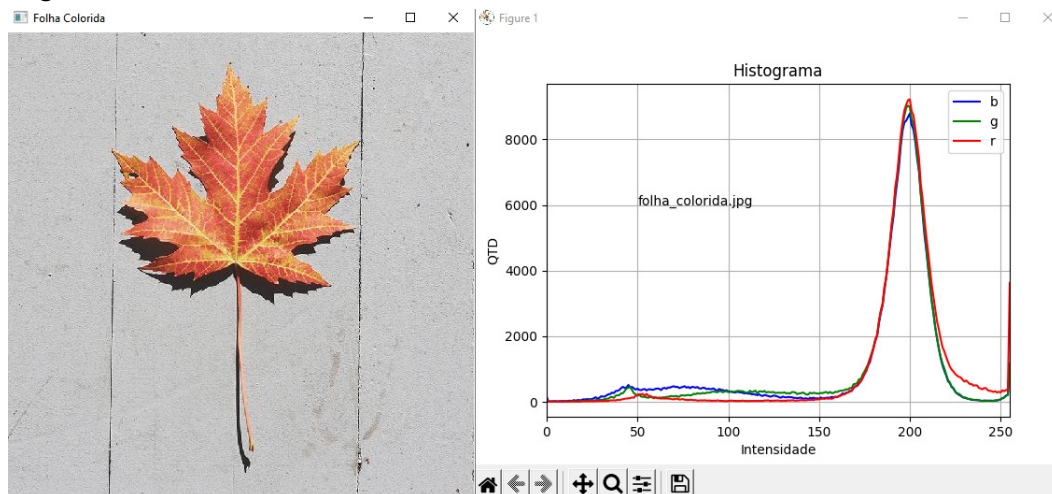
```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('folha_colorida.jpg', -1) # unchanged
cv2.imshow('Folha Colorida',img)
color = ('b','g','r')
for channel,col in enumerate(color):
    histr = cv2.calcHist([img], [channel], None, [256], [0,256])
    plt.plot(histr, color = col, label=col[channel])
    plt.xlim([0,256])
plt.title("Histograma")
plt.xlabel("Intensidade")
plt.ylabel("QTD")
plt.text(50, 6000, "folha_colorida.jpg")
plt.xlim(0,255)
plt.grid(True)
plt.legend()
plt.show()
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Prof. César C. Xavier

Processamento de Imagens e Visão Computacional

Histograma

Imagem Colorida



Prof. César C. Xavier

Equalização de Histograma

- Antes, há necessidade de interpretar o histograma.
- Níveis de exposição à luz e de contraste são informações que podem ser extraídos de um histograma.

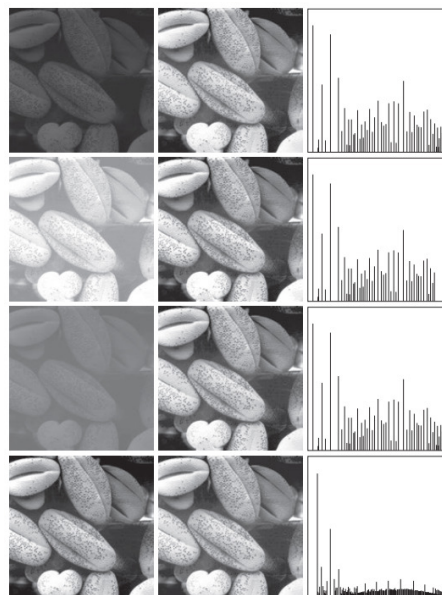


Figura 3.20 Coluna da esquerda: imagens da Figura 3.16. Coluna central: imagens que correspondem aos histogramas equalizados. Coluna da direita: histogramas das imagens da coluna central.

Prof. César C. Xavier

Equalização de Histograma

- Antes, há necessidade de interpretar o histograma.
- Imagens subexpostas x Imagens superexpostas.

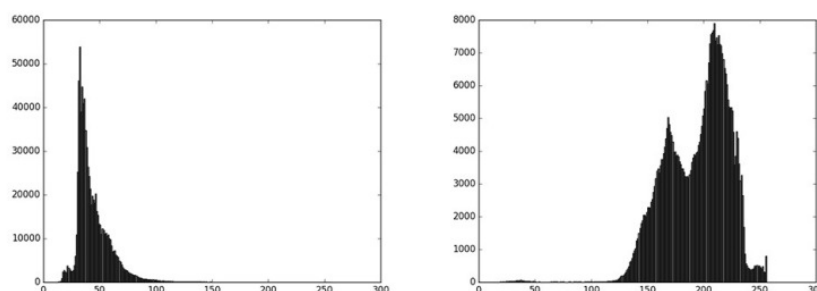


Figura 5.6: Análise de exposição em histograma

Prof. César C. Xavier

Equalização de Histograma

- Antes, há necessidade de interpretar o histograma.
- Imagens baixo nível contraste x Imagens alto nível contraste.

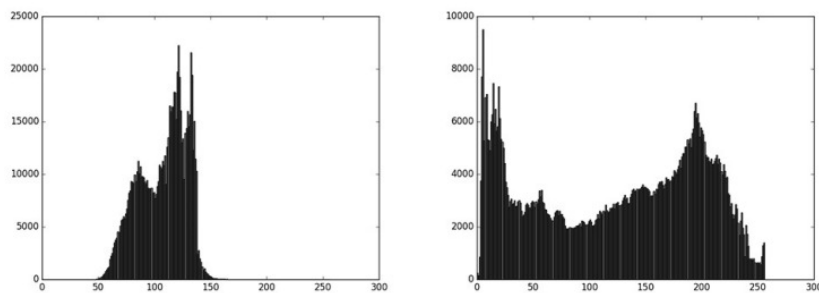
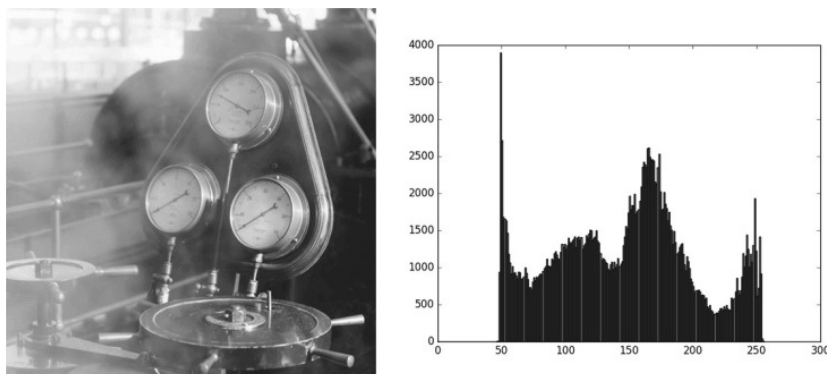


Figura 5.7: Análise de contraste em histograma

Prof. César C. Xavier

Equalização de Histograma

- Melhorando a nitidez



Prof. César C. Xavier

Equalização de Histograma

- Melhorando a nitidez: `equalizeHist()`

```
import cv2
import numpy as np
from matplotlib import pyplot as grafico

imagemOriginal = cv2.imread("maquina.jpg", 0)
imagemEqualizada = cv2.equalizeHist(imagemOriginal)

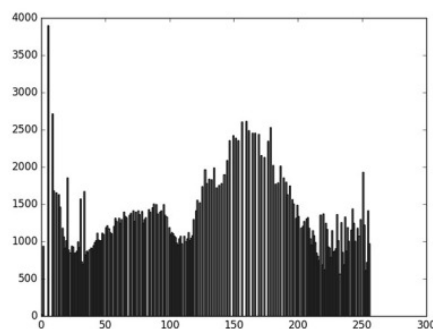
cv2.imshow("Imagem Original", imagemOriginal)
cv2.imshow("Imagem Equalizada", imagemEqualizada)
grafico.hist(imagemOriginal.ravel(), 256,[0,256])
grafico.figure();

grafico.hist(imagemEqualizada.ravel(), 256,[0,256])
grafico.show()
```

Prof. César C. Xavier

Equalização de Histograma

- Melhorando a nitidez: `equalizeHist()`



Prof. César C. Xavier

FIM