

Computação Gráfica

Histograma de Imagem Digital

(Aula 6)

Leonardo Medeiros

Instituto Federal de Alagoas

24 de Maio de 2017

Roteiro

1 Histograma

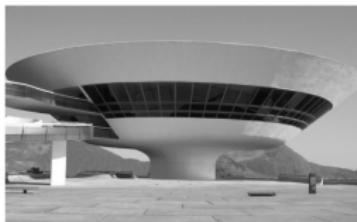
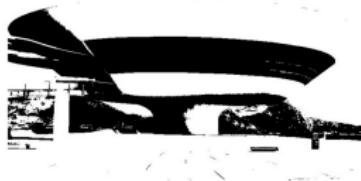
2 Plot

3 Máscara

4 Bibliografia

Imagens Monocromáticas

Imagens monocromáticas são imagens digitais onde cada *pixel* possui apenas uma **banda espectral**.



Exemplos de imagens monocromáticas

O número de tons entre os valores limites, **branco** e **preto**, que se pode representar em tons, depende de quantos **bits** são alocados na matriz de imagem para armazenar o tom de cada *pixel*.

Número de elementos na Escala de cinza	Tons de cinza limites	Números de Bits necessários para representação do <i>pixels</i>
2^1 2 valores	0,1	1
2^3 8 valores	0 a 7	3
2^4 16 valores	0 a 15	4
2^8 256 valores	0 a 255	8

Uma imagem **monocromática** pode ser representada geometricamente também por **valores reais** quanto à posição dos *pixels* como no gráfico $G(f)$ da função f :

$$G(f) = \{ (x,y,z); (x,y) \in R^2 ; z = f(x,y) \} \quad (3.4)$$

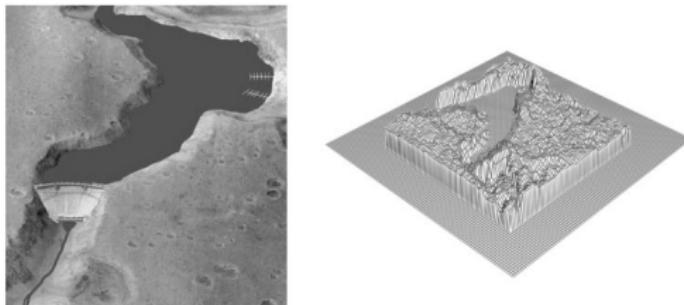


Imagen Monocromática de uma represa e seu gráfico 3D na forma de sua função $G(f)$ no R^3 .

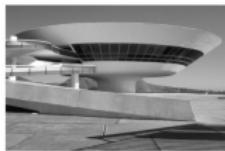
Imagens Coloridas

Imagens multibandas são imagens digitais onde cada *pixel* possui *n* bandas espectrais (*n*>1).

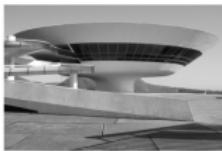
Quando uma imagem é representada pela composição de **três bandas visíveis (como as RGB)** tem-se uma **imagem colorida aos olhos humanos**.



(a) Imagem Colorida



(b) Banda Vermelha (Red)



(c) Banda Verde (Green)



(d) Banda Azul (Blue)

Imagen colorida e cada uma de suas bandas RGB.

Histograma de Imagem Digital

Uma imagem possui uma série de informações armazenadas, e uma das mais relevantes refere-se à cor ou ao tom de seus pixels. Para iniciar o processo de análise de uma imagem, pode-se observá-la a partir de uma ferramenta usual em Estatística: o **histograma**.

Histograma de Imagem Digital

O Histograma de uma imagem é simplesmente um conjunto de números indicando o percentual de pixels naquela imagem, que apresenta determinado nível de cinza ou cor.

Esses valores são normalmente representados por um gráfico de barras que fornece, para cada nível de cinza, número (ou percentual) de pixels correspondentes na imagem.

3.4. Histograma de imagem digital

O histograma de uma imagem indica o percentual de *pixels* que a imagem tem de determinado nível de cinza ou cor.

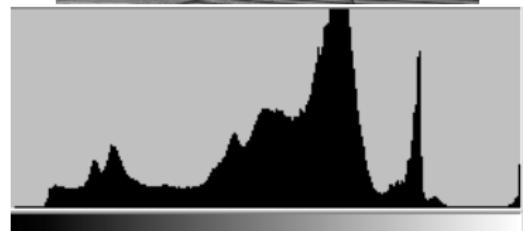


Figura 3.26 – Imagem em tons de cinza e o seu histograma.

Histograma

Indicação de Qualidade Quanto ao Nível de Contraste e Luminosidade Média

O histograma de uma imagem fornece uma indicação de sua qualidade quanto ao nível de contraste e quanto sua luminosidade média(ou seja, se a imagem é predominantemente clara ou escura).

O histograma fornece uma indicação da qualidade da imagem quanto ao contraste e intensidade luminosa.

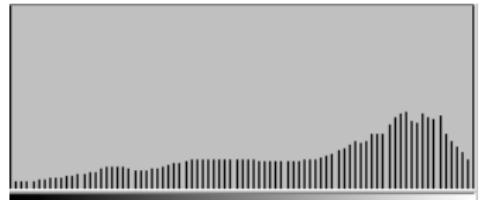
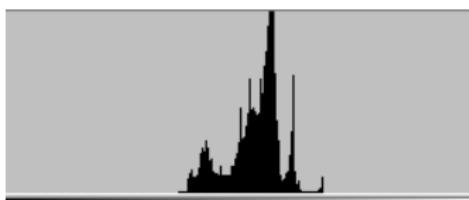


Figura 3.27 – Imagens em tons de cinza e seus respectivos histogramas: com pouco (direita) e muito contraste (esquerda).

Vamos Brincar um Pouco...

Converter uma Imagem Colorida Para Escala de Cinza

Primeiramente, vamos aprender como converter uma imagem colorida em Escala de Cinza.

Converte a Imagem Para *GrayScale*

```
import cv2
import numpy as np

image = cv2.imread('..../img/babuino2.jpg')
grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imwrite('..../img/babuino2gray.jpg', grayImage)
cv2.imshow('Imagen_Cinza', grayImage)
cv2.waitKey()
cv2.destroyAllWindows()
```

Converter Imagem Escala Cinza



Figura: em BGR



Figura: em GrayScale

Vamos Brincar um Pouco...

Criar o Histograma de Uma Imagem GrayScale

Agora que temos nossa imagem em cinza, vamos criar seu Histograma.

Função *calcHist()* do Open CV

Cálculo de Histograma no Open CV

A função *calcHist()* encontra o histograma de uma imagem. Para fazê-lo precisamos passar os devidos parâmetros
`(cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]]))`

Parâmetros da Função *calcHist()*

Descrição dos Parâmetros (PARTE1/2)

images : É a fonte da imagem de tipo *uint8* ou *float32*. Este parâmetro deve ser passado com uso de `[]`.

channels : É também repassado com uso de `[]`. Contém a quantidade de canais do histograma. Por exemplo, se passarmos uma imagem em *grayscale* seu valor será `[0]`. Para imagens coloridas você deve passar `[0], [1], [2]` para calcular o histograma de *blue*, *green* e *red* respectivamente.

Parâmetros da Função *calcHist()*

Descrição dos Parâmetros (PARTE2/2)

mask : É a máscara da imagem. Para encontrar o histograma da imagem completa, deve-se passar como parâmetro *None*. Mas, se quiser encontrar o histograma de uma região em particular, crie a máscara desta região (iremos mostrar posteriormente).[256]

histSize : Representa a discretização dos valores. Precisa ser repassado usando []. Para a escala completa de valores, passamos como parâmetro [256].

range : Este é o nosso RANGE ou intervalo de valores. Normalmente é utilizado [0, 256].

Vamos Chamar a Função *calcHist()*

Calcular o Histograma de Uma Imagem em *Grayscale*

```
cv2.calcHist([grayImage], [0], None, [256],[0, 256])
```

Calcular um Histograma

```
import cv2
import numpy as np

image = cv2.imread('..../img/babuino2.jpg')
grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow('Imagen_Cinza', grayImage)
histogram = cv2.calcHist([grayImage], [0], None, [256],
                        [0,256])
print(histogram)
cv2.waitKey()
cv2.destroyAllWindows()
```

Resultado do Histograma

histogram é um vetor 256×1 , onde cada valor corresponde a quantidade de pixels que aquela imagem possui.

Certo. E daí :p ?

Resultado do Histograma

Realmente. Imprimir este vetor não me ajudará em nada.
Os valores do histogramas serão estudados mais à frente quando quisermos comparar imagens. No momento, Vamos fazer o gráfico para compreendermos melhor esta ferramenta.

Vamos brincar um pouco...

Desta vez com Matplotlib

Matplotlib é uma biblioteca para Python 2D para plotar gráficos com qualidade. Esta biblioteca possui uma função específica que já calcula e exibe o histograma de uma imagem (`matplotlib.pyplot.hist()`).

Plot de um Histograma em *GrayScale*

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

image = cv2.imread('../img/lena.jpg')
grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2.imshow('Imagen_Cinza', grayImage)
plot.hist(grayImage.ravel(), 256, [0,256]);
plot.savefig('../img/histogramgray.jpg')
plot.show()
cv2.waitKey()
cv2.destroyAllWindows()
```

Histograma de uma Imagem Escala Cinza



Figura: em GrayScale

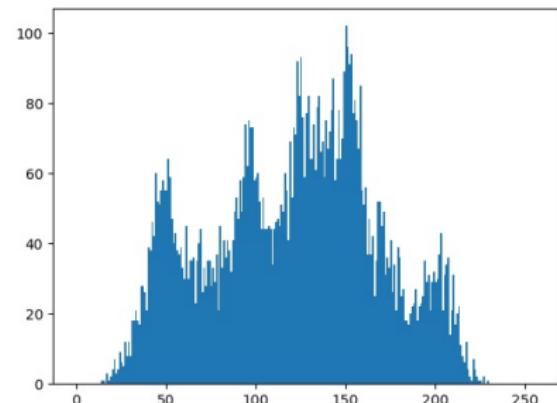


Figura: do Histograma em
GrayScale

Vamos criar um Histograma Colorido

Desta vez com Matplotlib

Matplotlib é uma biblioteca para Python 2D para plotar gráficos com qualidade. Esta biblioteca possui uma função específica que já calcula e exibe o histograma de uma imagem (`matplotlib.pyplot.hist()`).

Plot de um Histograma Colorido

```
import cv2
import numpy as np
from matplotlib import pyplot as plot
image = cv2.imread('../img/babuino2.jpg')
cv2.imshow('Imagen_Colorida', image)
color = ('b','g','r')
for i,col in enumerate(color):
    histr = cv2.calcHist([image],[i],None
                      ,[265],[0,256])
    plot.plot(histr,color = col)
    plot.xlim([0,256])
plot.savefig('../img/histogramcolor.jpg')
plot.show()
cv2.waitKey()
cv2.destroyAllWindows()
```

Histograma de uma Imagem Colorida



Figura: em GrayScale

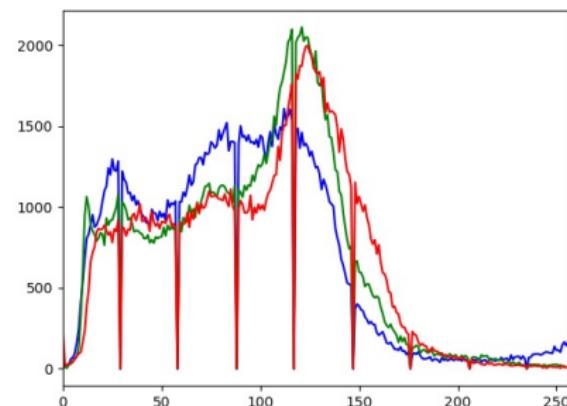


Figura: do Histograma em
GrayScale

Mask (Máscara)

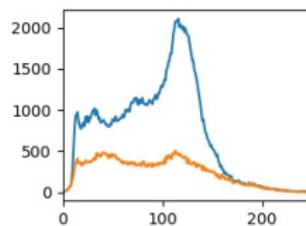
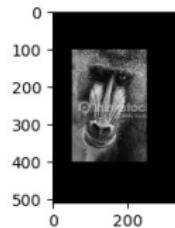
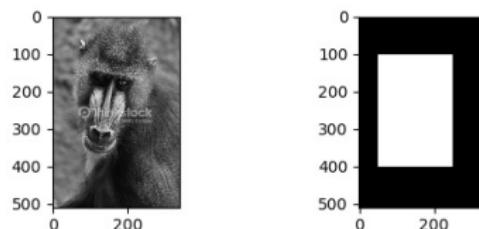
Como falamos anteriormente, existe o parâmetro *MASK* que delimita uma região de uma imagem. Podemos também atribuir como ROI (Region of Interest). E se quiséssemos encontrar o histograma de uma região e compará-lo com toda a região ? Logo, este é um dos principais usos de uma máscara. Neste contexto, eu posso trabalhar apenas com um subjconjunto da imagem e recuperar informações sobre a região e reduzir o custo computacional.

SubPlot de uma Máscara

```
# create a mask
mask = np.zeros(grayImage.shape[:2], np.uint8)
mask[100: 400, 50:250] = 255
masked_img = cv2.bitwise_and(grayImage, grayImage, mask = mask)

# Calculate histogram with mask and without mask
# Check third argument for mask
hist_full = cv2.calcHist([grayImage],[0],None,[256],[0,256])
hist_mask = cv2.calcHist([grayImage],[0],mask,[256],[0,256])
plot.subplot(221), plot.imshow(grayImage, 'gray')
plot.subplot(222), plot.imshow(mask, 'gray')
plot.subplot(223), plot.imshow(masked_img, 'gray')
plot.subplot(224), plot.plot(hist_full), plot.plot(
    hist_mask)
plot.xlim([0,256])
plot.show()
```

Histograma de uma Máscara



Bibliografia

- Solomon C., Breckon T. **Fundamentals of Digital Image Processing.. A Practical Approach with Examples in Matlab** (Wiley-Blackwell, 2011)(ISBN 9780470844724)(en)(355s)
- A. Conci, E. Azevedo e F.R. Leta - **Computação Gráfica: volume 2 (Processamento e Análise de Imagens Digitais)**, Campus/Elsevier. 2008 - ISBN 85-352-1253-3.
- Open CV Tutorials http://docs.opencv.org/3.1.0/d1/db7/tutorial_py_histogram_begins.html
- Histogram Calculation http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/histogram_calculation/histogram_calculation.html
- Matplotlib <http://matplotlib.org/>

DÚVIDAS ?