Universidade Estadual do Norte do Paraná

Centro de Ciências Tecnológicas

Curso: Ciência da Computação
Disciplina: Computação Gráfica
Prof. Bruno Miguel Nogueira de Souza

Notas de Aula – Computação Gráfica

Espaços de Cor

RGB
YCbCr e YUV

Exercício – Construa um algoritmo que converta os valores RGB de um pixel para o espaço de cor YCbCr de acordo com a matriz a seguir:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0{,}2568 & 0{,}5041 & 0{,}0979 \\ -0{,}1482 & -0{,}2910 & 0{,}4392 \\ 0{,}4392 & -0{,}3678 & -0{,}0714 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

Histogramas

O histograma armazena a frequência com que uma cor está presente na imagem.

Para os canais R,G e B é possível realizar o calculo do histograma para cada canal em separado (como pode ser visualizado na Figura 1 os histogramas de Vermelho, Verde e Azul de uma imagem. Cada ponto no eixo X representa a intensidade da cor (de 0 a 255) e o eixo Y representa a quantidade de pixels para cada uma das intensidades do eixo X.

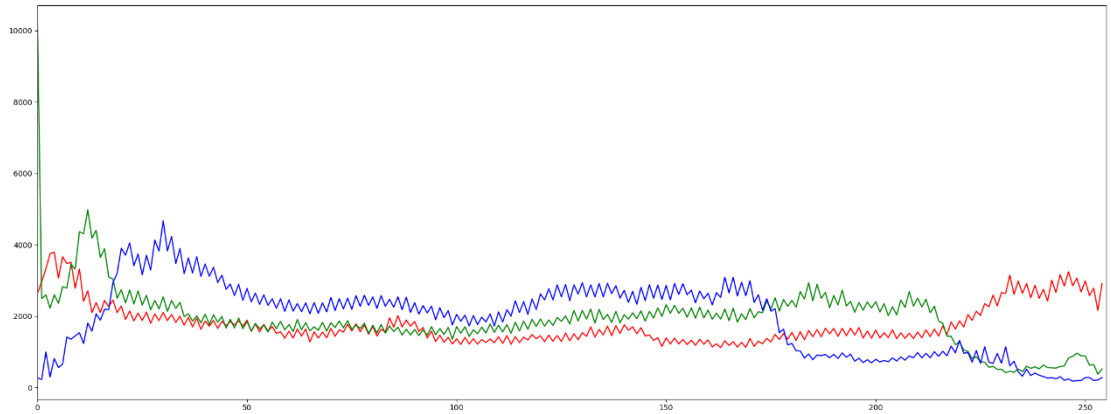

Figura 1 – Exemplo de Histograma

Exercício: Faça um algoritmo em python que leia uma imagem e calcule o histograma desta imagem para cada canal (RGB).

```
import cv2
from matplotlib import pyplot as plt
```

```
path = "D:\\imgsAula\\fire1.jpg"

img = cv2.imread(path)
print(img.shape)

h,w = img.shape[:2]
print("Tamanho da Imagem \n Largura: "+str(w)+ " Altura: " +str(h))
hnew = int(h/4)
wnew = int(w/4)

#resized = cv2.resize(img,(wnew,hnew), interpolation=cv2.INTER_AREA)
resized2 = cv2.resize(img,(wnew,hnew), interpolation=cv2.INTER_AREA)

imgR = resized2[:,:,2]
cv2.imshow("Vermelho",imgR)
imgG = resized2[:,:,1]
cv2.imshow("Verde",imgG)
imgB = resized2[:,:,0]
cv2.imshow("Azul",imgB)
#Calculando Histogramas pelo OpenCV

color = ('b','g','r')
histR = cv2.calcHist([imgR], [0], None, [255], [0, 255])
print(histR)
plt.plot(histR,color = color[2])
plt.xlim([0,255])
histG = cv2.calcHist([imgG], [0], None, [255], [0, 255])
plt.plot(histG,color = color[1])
plt.xlim([0,255])
histB = cv2.calcHist([imgB], [0], None, [255], [0, 255])
plt.plot(histB,color = color[0])
plt.xlim([0,255])
plt.show()

#cv2.imshow("Imagem ",resized)
cv2.imshow("Imagem2 ",resized2)
cv2.waitKey(0)
```

Calculando o Histograma

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

path = "D:\\imgsAula\\luke.jpg"

img = cv2.imread(path)

print(img.shape)
h,w,c = img.shape
hn = int(h/4)
wn = int(w/4)
resized = cv2.resize(img,(wn,hn))
histR = np.zeros(256, np.float)
histV = np.zeros(256, np.float)
histA = np.zeros(256, np.float)
# tem as seguinte caracteristica (altura, largura e canais)
for i in range(hn):
    for j in range(wn):
```

```
        pixelVermelho = resized[i,j,2]
        histR[resized[i,j,2]] = histR[resized[i,j,2]] + 1
        pixelVerde = resized[i, j, 1]
        histV[resized[i, j, 1]] = histV[resized[i, j, 1]] + 1
        pixelAzul = resized[i, j, 0]
        histA[resized[i, j, 0]] = histA[resized[i, j, 0]] + 1
print(histR)

#tabela de probabilidades...
pixeis = hn*wn
histR = histR/pixeis
histV = histV/pixeis
histA = histA/pixeis
print(histR)

color = ('r','g','b')
plt.plot(histR,color = color[0])
plt.xlim([0,255])
plt.plot(histV,color = color[1])
plt.xlim([0,255])
plt.plot(histA,color = color[2])
plt.xlim([0,255])
plt.show()
```

Normalização de Espaços de Cor

Normalização entre 0 e 1. Onde x é o valor do canal de cor C, Cmin é o valor mínimo que este canal pode possuir e Cmax é o valor máximo que C pode possuir:

$$Cx = \frac{x - Cmin}{Cmax - Cmin}$$

Demonstração de conversão de cores, transformação de cores de uma imagem utilizando a seguinte fórmula para obtenção das cores: R = Cr, B = Cb e G = (Y+Cb+Cr)/3.