

Aplicação de Paleta de Cores em uma Imagem

Diogo do Nascimento Paza¹

¹Universidade Estadual de Maringá(UEM)

Processamento Digital de Imagens

Professor: Franklin César Flores

diogopazacvel@gmail.com

Paleta de cores

Neste trabalho será adquirida uma imagem de entrada em tons de cinza. E posteriormente será aplicada nesta imagem uma paleta de cores HSI que varia de cor inicial até cor final e estas cores serão aplicadas na imagem original de entrada.

Para realizar este trabalho será utilizado os pacotes Numpy e OpenCV. O código gerado neste trabalho está todo comentado no final do artigo.

HSI

HSI(Hue, Saturation, Intensity), frequentemente utilizado pelos sistemas de processamento de imagem.

Separa a informação de cor(H+S) de sua intensidade.

O modelo HSI pode ser convertido diretamente para o RGB e vice-versa.

S pode ser interpretado com a pureza da cor(distância para a cor pura) e H pode ser interpretado como a distância para o vermelho.

Imagens de saída:



Figura 1 - Paleta de cores HSI gerada entre intervalo 0 até 60



Figura 1 - Imagem Original



Figura 2 - Imagem com cores da paleta HSI

Código escrito em Python do trabalho:

#Função para gerar uma paleta de cores com início e fim
def criar_paleta(inicio, fim):

```
#Gera vetor entre inicio e fim com 256 posições  
h=np.linspace(inicio, fim,256)  
s=np.tile(SV,256)  
i=np.tile(SV,256)
```

```
#Dando nova forma ao array  
novoH=np.tile(h.reshape((256,1)), PALETA)  
novoS=np.tile(s.reshape((256,1)), PALETA)  
novoI=np.tile(i.reshape((256,1)), PALETA)
```

```

#converte para 8bits
novoH=np.uint8(novoH)
novoS=np.uint8(novoS)
novoI=np.uint8(novoI)
#unindo os vetores para gerar a paleta
minhaPaleta = np.dstack((np.dstack((novoH,novoS)), novoI))
return minhaPaleta

#criando nova imagem com 3 canais
img2=np.zeros( (img.shape[0], img.shape[1],3 ))

#gera paleta HSI
paleta = criar_paleta(0,60)

#percorrendo cores
cores_paleta=paleta[:,0]
img=cores_paleta[img]

bgr_img=cv2.cvtColor(img, cv2.COLOR_HSV2BGR)
bgr_paleta=cv2.cvtColor(paleta, cv2.COLOR_HSV2BGR)

#imagens finais
cv2.imshow("original", minha_img)
cv2.imshow("paleta", bgr_paleta)
cv2.imshow("Imagem de Saida", bgr_img)

```