

Práctica feb 26, 2021

Primera parte:

Usando los paquetes

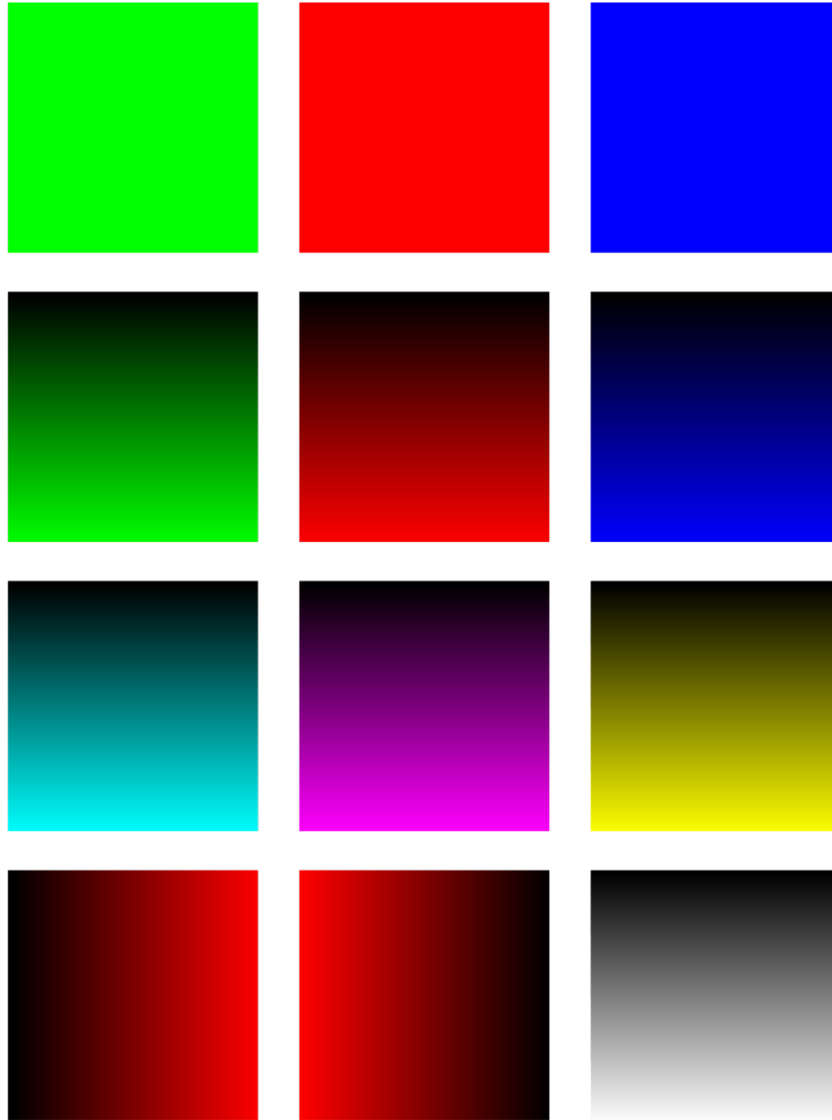
```
[2]: import matplotlib.pyplot as plt
import numpy as np
import math
```

y la forma “estándar” para recorrer matrices

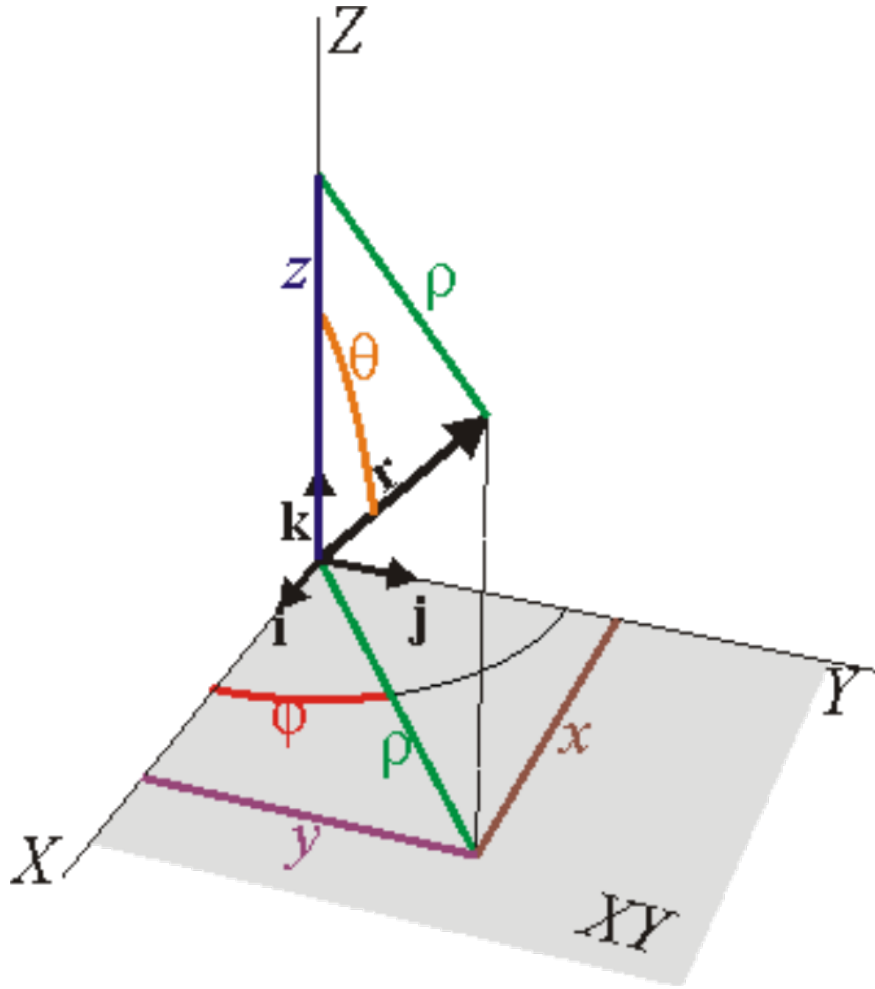
```
[4]: M = 64
N = 64
img = np.zeros((M,N,3))
img.shape
# populate array
for f in range(M):
    for c in range(N):
        img[f,c,:]=np.array((1.,0.,0.))

plt.axis('off')
plt.imshow(img)
```

generar las siguientes imágenes



Para obtener los distintos colores desplazándonos a través del “cubo RGB” (para obtener los gradientes de color en una determinada dirección), recordar que cada color se puede obtener como combinación lineal de vectores (colores) base $\vec{r} = [1, 0, 0]^T$, $\vec{g} = [0, 1, 0]^T$ y $\vec{b} = [0, 0, 1]^T$. Puede resultar útil también el siguiente esquema:



En donde los ejes X, Y, Z corresponden a los ejes del espacio de color R, G, B , respectivamente; mientras los vectores unitarios $\hat{i}, \hat{j}, \hat{k}$ corresponden a los vectores unitarios base para nuestro espacio de color $\vec{r} = [1, 0, 0]^T$, $\vec{g} = [0, 1, 0]^T$ y $\vec{b} = [0, 0, 1]^T$.

Segunda parte:

Genere círculos concéntricos en una imagen usando distintos colores (pueden ser generados de manera aleatoria).



¿Qué puede decir sobre la “visibilidad” cada uno de los círculos generados usando colores de forma alea-

toria? ¿cómo puede maximizar el contraste entre círculos contiguos?

Tercera parte:

Construya una función que permita ir entre los espacios de color RGB y HSI (recuerde que $R, G, B \in [0, 1]$).

Guía (verifique que las ecuaciones sean correctas):

- obtenga

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2} [(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\}$$

-

$$H = \begin{cases} \theta & \text{si } B \leq G \\ 360 - \theta & \text{si } B > G \end{cases}$$

-

$$S = 1 - 3 \min(R, G, B)$$

-

$$I = \frac{1}{3}(R + G + B)$$

Recuerde ser consistente con el uso de radianes y grados. Los rangos de las componentes H, S, I son escalados al rango $[0, 1]$ para presentarlos como imagen.

Para probar la función, leer una imagen a color (RGB - RGBA) y buscar su representación en el espacio HSI.

Sugerencia para leer imágenes

```
[89]: import matplotlib.cbook as cbook

with cbook.get_sample_data('ruta completa a la imagen') as image_file:
    img = plt.imread(image_file)

plt.imshow(img)
```