

FUNDAMENTOS MATEMÁTICOS

DE LA

VISIÓN ROBÓTICA

TRABAJO PRACTICO Nº 2: COLORES

Vázquez Lareu, Román. Facultad de Ingeniería de la UBA

10/015/2020

Introducción

En el siguiente informe, a partir de una imagen dada, se buscará recuperar cada grupo de colores correspondientes a las arandelas. Esto se hará generando una máscara para cada color. Se realizará tanto en el espacio RGB como en el HSV.

Desarrollo

En primer lugar, se importarán las librerías necesarias para el análisis de la imagen: **numpy**, **cv2**, **matplotlib.pyplot**.

Captura de Notebook Nº 1: Importación de librerías

```
1 %matplotlib inline
2 import numpy as np
3 import cv2 as cv
4 import matplotlib.pyplot as plt
```

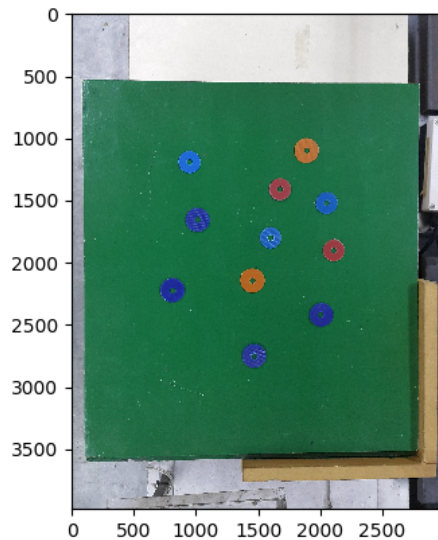
A continuación se cargará la imagen a color, y se la convertirá del formato BGR a RGB, para luego imprimirla por pantalla

Captura de Notebook Nº 2: Carga de imagen. Conversión a RGB e impresión por pantalla

```
1 img_color = cv.imread('arandelas.bmp')
2 imgRGB = cv.cvtColor(img_color, cv.COLOR_BGR2RGB)
3
4 plt.figure(0)
5 plt.imshow(imgRGB)
6 plt.show()
```

Se obtendrá el siguiente output:

Captura de Notebook № 3: Imagen a color RGB



1

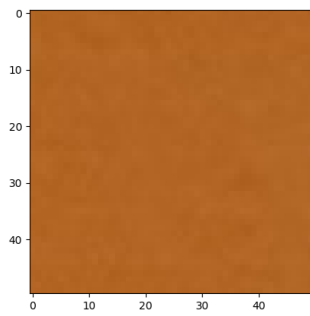
Dado que se busca generar máscaras para cada color de las arandelas, se toma una muestra de color representativa de cada arandela (una muestra por color de arandela).

Captura de Notebook № 4: Toma de muestra de color de arandelas e impresión por pantalla

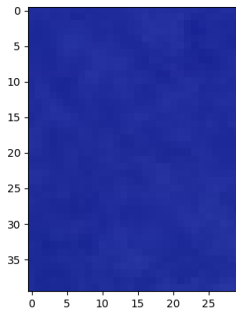
```
1 img_muestra_marron = imgRGB[2120:2170,1370:1420,:]  
2 img_muestra_azul = imgRGB[2230:2270,740:770,:]  
3 img_muestra_celeste = imgRGB[1180:1230,870:910,:]  
4 img_muestra_rojo = imgRGB[1400:1440,1610:1640,:]  
5  
6 lista_arandelas=[img_muestra_marron,img_muestra_azul,img_muestra_celeste,↵  
    img_muestra_rojo]  
7  
8 for i in range(1,5):  
9     plt.figure(i)  
10    plt.imshow(lista_arandelas[i-1])  
11 plt.show()
```

Obteniendo de esta manera el siguiente output:

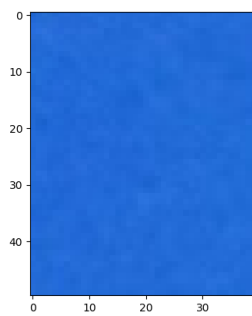
Captura de Notebook № 5: Muestras de color



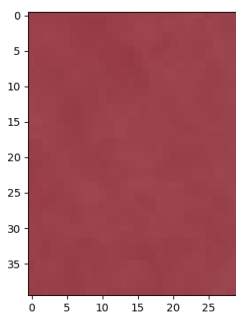
1



2



3



4

A continuación, se calculará un modelo estadístico para caracterizar el color de cada arandela en el espacio de color RGB

Captura de Notebook № 6: Cálculo del modelo estadístico

```
1 color_mean_marron, color_std_marron = cv.meanStdDev(img_muestra_marron)
2 color_mean_azul, color_std_azul = cv.meanStdDev(img_muestra_azul)
3 color_mean_celeste, color_std_celeste = cv.meanStdDev(img_muestra_celeste)
```

```
4 color_mean_rojo, color_std_rojo = cv.meanStdDev(img_muestra_rojo)
```

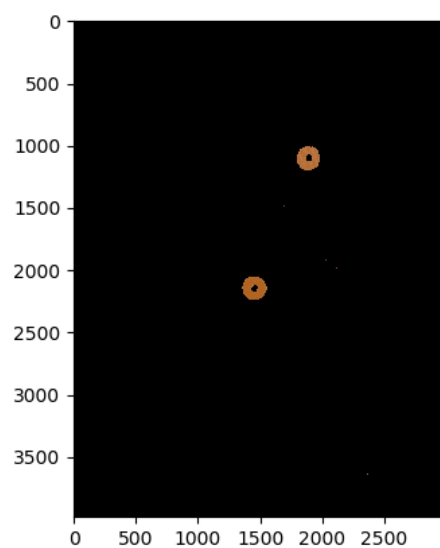
Finalmente se generan las máscaras y se aplican en la imagen original. En cada caso, se tomaron los **sigmas** que más favorecieran al objetivo de recuperar los colores.

Captura de Notebook № 7: Generación de máscaras y aplicación de las mismas

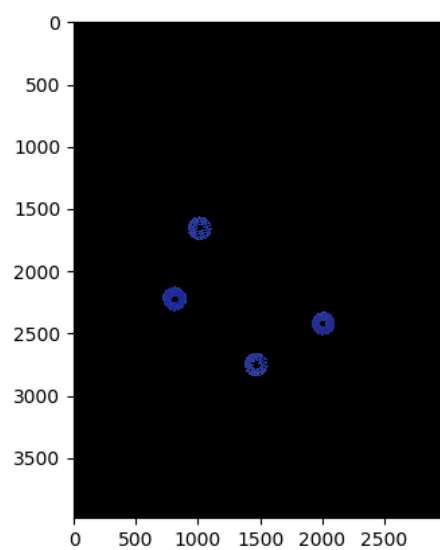
```
1 #mascara marron
2 mask = cv.inRange(imgRGB, color_mean_marron-color_std_marron*18, ↵
    color_mean_marron+color_std_marron*18)
3 img_segmentada_marron = cv.bitwise_and(imgRGB, imgRGB, mask=mask)
4
5 plt.figure(6)
6 plt.imshow(img_segmentada_marron)
7 plt.show()
8
9 #mascara azul
10 mask = cv.inRange(imgRGB, color_mean_azul-color_std_azul*12, ↵
    color_mean_azul+color_std_azul*12)
11 img_segmentada_azul = cv.bitwise_and(imgRGB, imgRGB, mask=mask)
12
13 plt.figure(7)
14 plt.imshow(img_segmentada_azul)
15 plt.show()
16
17 #mascara celeste
18 mask = cv.inRange(imgRGB, color_mean_celeste-color_std_celeste*13, ↵
    color_mean_celeste+color_std_celeste*13)
19 img_segmentada_celeste = cv.bitwise_and(imgRGB, imgRGB, mask=mask)
20
21 plt.figure(8)
22 plt.imshow(img_segmentada_celeste)
23 plt.show()
24
25 #mascara roja
26 mask = cv.inRange(imgRGB, color_mean_rojo-color_std_rojo*14, ↵
    color_mean_rojo+color_std_rojo*14)
27 img_segmentada_rojo = cv.bitwise_and(imgRGB, imgRGB, mask=mask)
28
29 plt.figure(9)
30 plt.imshow(img_segmentada_rojo)
31 plt.show()
```

Obteniendose el siguiente output:

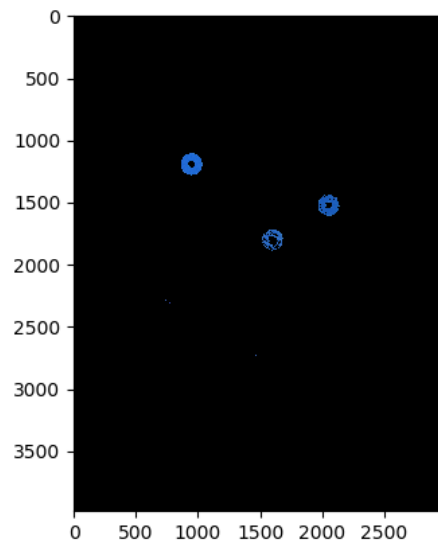
1



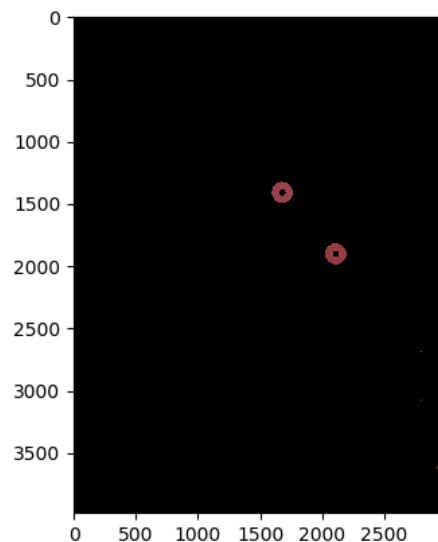
2



3



4



5

Puede observarse un resultado mucho más satisfactorio que en el caso de los morrones, esto se debe principalmente a la iluminación (ausencia de brillos) y a la uniformidad del color de las arandelas en contraposición con la variación de color que presentaba cada morrón.

Máscaras en espacio HSV

A continuación, se realizarán las misma máscaras pero a partir del análisis desde el espacio HSV. En principio convertimos la imagen original a color del espacio BGR al HSV

Captura de Notebook № 9: Conversión a HSV

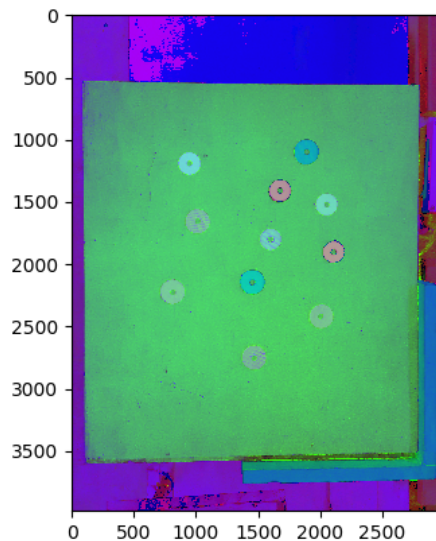
```
1 img_HSV = cv.cvtColor(img_color, cv.COLOR_BGR2HSV)
```

```
2 plt.figure(10)
3 plt.imshow(img_HSV)
```

obteniendo la siguiente imagen de salida:

Captura de Notebook № 10: Imagenes con máscaras HSV

1



2

Luego generamos las máscaras para cada color, pero esta vez ponemos como parámetro principal al matiz ("H:Hue") y como secundarios o de mayor amplitud a la saturación ("S:Saturation") y al valor ("V:Value").

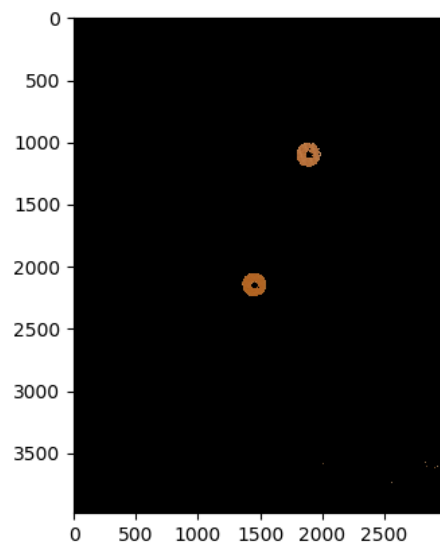
Captura de Notebook № 11: Generación de máscaras y aplicación de las mismas

```
1 #mascara marron HSV
2 color_l_marron = (13,100,100)
3 color_u_marron = (15,255,255)
4
5 mask_marron = cv.inRange(img_HSV, color_l_marron, color_u_marron)
6 img_segmentada_marron = cv.bitwise_and(imgRGB, imgRGB, mask=mask_marron)
7
8 plt.figure(11)
9 plt.imshow(img_segmentada_marron)
10 plt.show()
11
12 #mascara azul HSV
13 color_l_azul = (109,100,100)
```

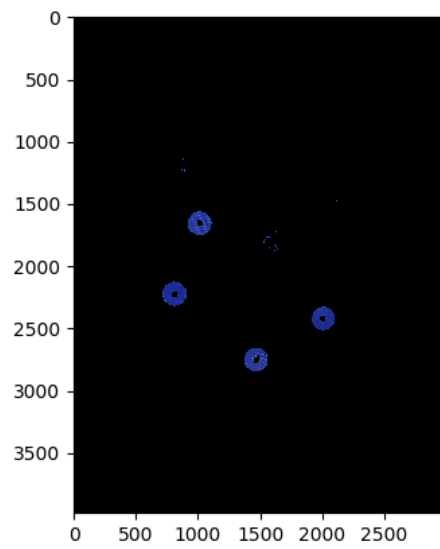
```
14 color_u_azul = (118,255,255)
15
16 mask_azul = cv.inRange(img_HSV, color_l_azul, color_u_azul)
17 img_segmentada_azul = cv.bitwise_and(imgRGB, imgRGB, mask=mask_azul)
18
19 plt.figure(12)
20 plt.imshow(img_segmentada_azul)
21 plt.show()
22
23 #mascara celeste HSV
24 color_l_celeste = (106,100,100)
25 color_u_celeste = (108,255,255)
26
27 mask_celeste = cv.inRange(img_HSV, color_l_celeste, color_u_celeste)
28 img_segmentada_celeste = cv.bitwise_and(imgRGB, imgRGB, mask=mask_celeste)
29
30 plt.figure(13)
31 plt.imshow(img_segmentada_celeste)
32 plt.show()
33
34 #mascara roja HSV
35 color_l_rojo = (176,100,100)
36 color_u_rojo = (182,255,255)
37
38 mask_rojo = cv.inRange(img_HSV, color_l_rojo, color_u_rojo)
39 img_segmentada_rojo = cv.bitwise_and(imgRGB, imgRGB, mask=mask_rojo)
40
41 plt.figure(14)
42 plt.imshow(img_segmentada_rojo)
43 plt.show()
```

Obteniendo las siguientes imágenes:

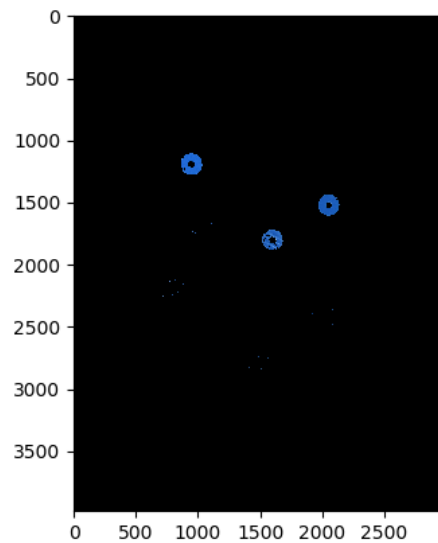
Captura de Notebook № 12: Máscaras HSV



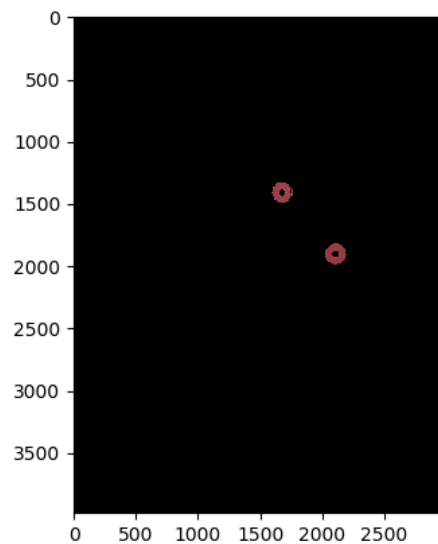
1



2



3



4

En el caso del marrón y rojo el resultado final fue similar tanto en el espacio RGB como en el HSV. Sin embargo, con el celeste y el azul, a partir del RGB se pudieron diferenciar entre ellos de mejor manera que en el caso HSV. Si quisieramos, en el HSV, discriminar las distintas tonalidades, deberíamos ajustar a su vez la saturación ("S") y el valor ("V")

Captura de Notebook № 13: Máscaras de azul y celeste ajustando los 3 valores de HSV

```

1 #mascara azul ajustando S y V
2 color_l_azul_ajustada = (109,150,130)
3 color_u_azul_ajustada = (118,210,180)
4
5 mask_azul_ajustada = cv.inRange(img_HSV, color_l_azul_ajustada, ↵
    color_u_azul_ajustada)

```

```

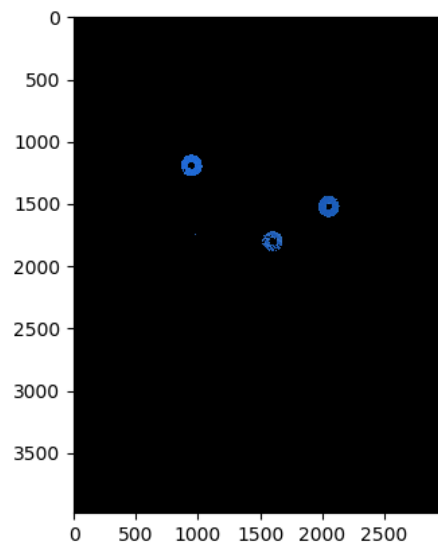
6  img_segmentada_azul_ajustada = cv.bitwise_and(imgRGB, imgRGB, mask=↵
    mask_azul_ajustada)
7
8  plt.figure(16)
9  plt.imshow(img_segmentada_azul_ajustada)
10 plt.show()
11
12 #mascara celeste ajustando S y V
13 color_l_celeste_ajustada = (106,180,165)
14 color_u_celeste_ajustada = (108,230,225)
15
16 mask_celeste_ajustada = cv.inRange(img_HSV, color_l_celeste_ajustada, ↵
    color_u_celeste_ajustada)
17 img_segmentada_celeste_ajustada = cv.bitwise_and(imgRGB, imgRGB, mask=↵
    mask_celeste_ajustada)
18
19 plt.figure(15)
20 plt.imshow(img_segmentada_celeste_ajustada)
21 plt.show()

```

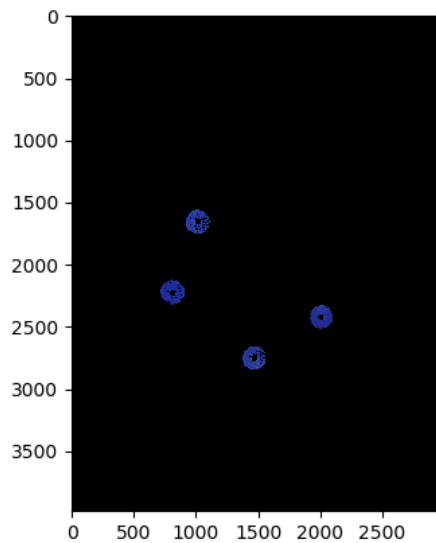
Ajustando los tres valores, en el caso del celeste y azul se obtuvieron las siguientes imágenes

Captura de Notebook № 14: Imágenes con máscaras de azul y celeste ajustadas

1



2



3

Conclusión

En conclusión, si la iluminación no es ideal, al momento de generar una máscara es conveniente trabajar en el espacio de HSV. Este sería el caso de la imagen de los morrones, cuyos colores no eran uniformes y la iluminación no era pareja en toda su superficie. Sin embargo, en la imagen de las arandelas, en general, la iluminación era pareja para todas, sumado a que no había presencia de brillos. De esta manera, utilizando el espacio RGB se pudo recuperar de buena manera cada color, incluso discriminando tonalidades. En el caso del HSV, para discriminar tonalidades se tuvo que ajustar no sólo el H (lógico), sino que también el S y V, a diferencia del caso de los morrones, donde los valores tomados eran amplios.