

Centro de Enseñanza Técnica Industrial

Visión Artificial.

Ingeniería Mecatrónica.



Practica# 6: Filtros en los espacios de Color.

Platel: CETI Colomos.

Nombre: Ruiz Macías Luis Enrique - 21310196

Grado/Grupo: 6°G

Objetivo: El objetivo de la siguiente práctica es de hacer uso de los distintos filtros que nos proporciona cv2 para discriminar o buscar los diferentes colores en una imagen.

En la siguiente practica usamos CV2 con el afán de buscar colores en especifico utilizando los diferentes filtros con el objetivo de mostrar las funcionalidades y la utilidad de estos filtros, como lo son la detención de objetos en especifico por su tonalidad en lugar de sus formas o bordes, si no por su tonalidad.

Hacemos uso de filtros RGB(rojo, verde, Azul) HSV(Tono, Saturación, Valor) y YUV(Iluminación y Crominancia). Estos son sistemas de representación en espacios de color, por lo que los diferentes filtros mostraran diferentes imágenes finales.

A continuación, se presenta el siguiente código:

#Programa que muestra el histograma de cada una de las imágenes y antes y después de ecualizarla.

```
import cv2
```

```
import numpy as np
```

```
cap = cv2.VideoCapture(0)
```

```
modo = 'original'
```

```
while True:
```

```
    ret, frame = cap.read()
```

```
    if not ret:
```

```
        break
```

```
    salida = frame.copy()
```

```
    if modo == 'original':
```

```
        salida = frame.copy()
```

```
    elif modo.startswith('hsv'):
```

```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

```
#Eleccion de Modo de filtro
```

```
if modo == 'hsv_red':
```

```
    lower_red1 = np.array([0, 120, 70])
```

```
    upper_red1 = np.array([10, 255, 255])
```

```
    lower_red2 = np.array([170, 120, 70])
```

```
    upper_red2 = np.array([180, 255, 255])
```

```
    mask1 = cv2.inRange(hsv, lower_red1, upper_red1)
```

```
    mask2 = cv2.inRange(hsv, lower_red2, upper_red2)
```

```
    mask = cv2.bitwise_or(mask1, mask2)
```

```
    salida = cv2.bitwise_and(frame, frame, mask=mask)
```

```
elif modo == 'hsv_green':
```

```
    lower = np.array([35, 100, 100])
```

```
    upper = np.array([85, 255, 255])
```

```
    mask = cv2.inRange(hsv, lower, upper)
```

```
    salida = cv2.bitwise_and(frame, frame, mask=mask)
```

```
elif modo == 'hsv_blue':
```

```
    lower = np.array([100, 150, 0])
```

```
    upper = np.array([140, 255, 255])
```

```
    mask = cv2.inRange(hsv, lower, upper)
```

```
    salida = cv2.bitwise_and(frame, frame, mask=mask)
```

```
elif modo.startswith('yuv'):
```

```
    yuv = cv2.cvtColor(frame, cv2.COLOR_BGR2YUV)
```

```

if modo == 'yuv_red':
    lower = np.array([0,130,80])
    upper = np.array([255, 180, 130])
    mask = cv2.inRange(yuv, lower, upper)
    salida = cv2.bitwise_and(frame, frame, mask=mask)

elif modo.startswith('rgb'):
    b, g, r = cv2.split(frame)

    if modo == 'rgb_red':
        mask = cv2.inRange(r, 150, 255)
        salida = cv2.bitwise_and(frame, frame, mask=mask)

    elif modo == 'rgb_green':
        mask = cv2.inRange(g, 150, 255)
        salida = cv2.bitwise_and(frame, frame, mask=mask)

    elif modo == 'rgb_blue':
        mask = cv2.inRange(b, 150, 255)
        salida = cv2.bitwise_and(frame, frame, mask=mask)

# Mostrar el resultado
cv2.imshow('Filtro de Color', salida)

# Teclado para cambiar modos de filtro
key = cv2.waitKey(1) & 0xFF
if key == ord('q'):

```

```
        break
    elif key == ord('0'):
        modo = 'original'
    elif key == ord('1'):
        modo = 'hsv_red'
    elif key == ord('2'):
        modo = 'hsv_green'
    elif key == ord('3'):
        modo = 'hsv_blue'
    elif key == ord('4'):
        modo = 'rgb_red'
    elif key == ord('5'):
        modo = 'rgb_green'
    elif key == ord('6'):
        modo = 'rgb_blue'
    elif key == ord('7'):
        modo = 'yuv_red'

cap.release()
cv2.destroyAllWindows()
```



