

Visión Artificial



Título: Practica 6

Nombre: Mario Gerardo Casas Miramontes

Registro: 22310165

Objetivo

El objetivo de este tema es hacer la identificación de colores y que sean apartados de los elementos que no formen parte de ese color, siendo los filtros el rojo, azul y verde.

Funcionamiento

El funcionamiento del programa consiste en usar el hsv que es una forma en la que Python es capaz de distinguir los colores debido a que identifica el tono, saturación y valor, captando así la imagen de la cámara y comparando sus valores entre los que puede interpretarse los colores teniendo un mínimo y un máximo para la detección de cada color.

A continuación, se realiza la máscara de cada valor, con esto se busca identificar los colores que buscamos para después usar la función Bitwise en donde dependiendo de la máscara tomara ciertos puntos como blancos que al momento de compararse se mostrarán del color de la máscara, teniendo así las cámaras en la cual detectan los tres colores

Código Fuente

```
#Mario Gerardo Casas Miramontes 22310165
```

```
import cv2
```

```
import numpy as np
```

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

```
while(1):
```

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

```
    #Se lee el video de la cámara de forma HSV
```

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

```
    #Se define lo que es el tono, saturación y valor para cada color teniendo una parte baja y alta de los valores para color
```

```
    lower_red = np.array([150,150,50])
```

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

```
    lower_green = np.array([35, 50, 50])
```

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

```
    lower_blue = np.array([90, 50, 50])
```

```

upper_blue = np.array([130, 255, 255])

#Se crean mascaras para hacer una deteccion en binario en los que es blanco y negro
mask_red = cv2.inRange(hsv, lower_red, upper_red)
mask_green = cv2.inRange(hsv, lower_green, upper_green)
mask_blue = cv2.inRange(hsv, lower_blue, upper_blue)

#Son los resultados de la deteccion de blancos a traves de bitwise en donde idenitfica los colores
que se buscan.

res_red = cv2.bitwise_and(frame,frame, mask= mask_red)
res_green = cv2.bitwise_and(frame, frame, mask=mask_green)
res_blue = cv2.bitwise_and(frame, frame, mask=mask_blue)

#Se imprimen las imagenes
cv2.imshow('frame',frame)
cv2.imshow('mask',mask_red)
cv2.imshow('res',res_red)
cv2.imshow('Green Mask',res_green )
cv2.imshow('Blue Mask', res_blue)

k = cv2.waitKey(5) & 0xFF
if k == 27:
    break

cv2.destroyAllWindows()
cap.release()

```

Resultado



