

**Reporte práctica 6**

Cortes de la Cruz Miguel Angel

**Registro:** 19110127

**Grupo:** 7E1

**Maestro:** Mauricio Alejandro Cabrera Arellano

Enlace GitHub:

<https://github.com/MiguelC240/Practica6_19110127.git>

15-5-2022

**Introducción**

Dibujar sobre la imagen y ROI

**Objetivo:** Crear una imagen totalmente oscura para Dibujar y Escribir sobre la imagen, en otra imagen segmentar la región de interés.

**Desarrollo**

import numpy as np

import cv2

from matplotlib import pyplot as plt

Primero importamos las librerías que utilizaremos.

cap = cv2.VideoCapture(0)

cap.set(3, 300)

Obtenemos los fotogramas con los que vamos a trabajar mediante la funcion VideoCapture y la guardamos en la variable cap, después establecemos el tamaño de la ventana y la resolución.

redBajo1 = np.array([0, 100, 20], np.uint8)

redAlto1 = np.array([8, 255, 255], np.uint8)

Establecemos los parámetros de los componentes [H, S, V] donde en H va de 0 a 8, en S de 100 a 255 y finalmente en S va de 20 a 255. Esto es para establecer el primer parámetro del color rojo.

redBajo2=np.array([175, 100, 20], np.uint8)

redAlto2=np.array([179, 255, 255], np.uint8)

Establecemos los parámetros de los componentes [H, S, V] donde en H va de 175 a 179, en S de 100 a 255 y finalmente en S va de 20 a 255. Esto es para establecer el segundo parámetro del color rojo.

blueBajo1 = np.array([95, 100, 20], np.uint8)

blueAlto1 = np.array([145, 255, 255], np.uint8)

Establecemos los parámetros de los componentes [H, S, V] donde en H va de 95 a 145, en S de 100 a 255 y finalmente en S va de 20 a 255. Esto es para poder detectar el color azul.

greenBajo1 = np.array([35, 100, 20], np.uint8)

greenAlto1 = np.array([75, 255, 255], np.uint8)

Establecemos los parámetros de los componentes [H, S, V] donde en H va de 35 a 75, en S de 100 a 255 y finalmente en S va de 20 a 255. Esto es para poder detectar el color verde.

while True:

ret,frame = cap.read()

if ret==True:

frameHSV = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

maskRed1 = cv2.inRange(frameHSV, redBajo1, redAlto1)

maskRed2 = cv2.inRange(frameHSV, redBajo2, redAlto2)

maskRed = cv2.add(maskRed1, maskRed2)

maskRedvis = cv2.bitwise\_and(frame, frame, mask= maskRed)

cv2.imshow('frame', frame)

cv2.imshow('maskRed', maskRed)

cv2.imshow('maskRedvis', maskRedvis)

if cv2.waitKey(30) & 0xFF == ord('m'):

break

cap.release()

cv2.destroyAllWindows()

Ahora trabajaremos dentro de un while para que nuestra cámara siga capturando hasta que se presione la letra m debido a estamos capturando un video. Dentro del while hacemos la transformación de colores BGR a HSV, donde además utilizamos nuestros parametros declarados antes para el color rojo y esto lo concatenamos. Finalmente visualizamos todas nuestras ventanas.

Y obtenemos lo siguiente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

cap = cv2.VideoCapture(0)

cap.set(3, 300)

while True:

ret,frame = cap.read()

if ret==True:

frameHSV = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

maskBlue = cv2.inRange(frameHSV, blueBajo1, blueAlto1)

maskBluevis = cv2.bitwise\_and(frame, frame, mask= maskBlue)

cv2.imshow('Original', frame)

cv2.imshow('maskBlue', maskBlue)

cv2.imshow('maskBluevis', maskBluevis)

if cv2.waitKey(30) & 0xFF == ord('m'):

break

cap.release()

cv2.destroyAllWindows()

Realizamos lo mismo, pero ahora para el color azul y obtenemos lo siguiente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

cap = cv2.VideoCapture(0)

cap.set(3, 300)

while True:

ret,frame = cap.read()

if ret==True:

frameHSV = cv2.cvtColor(frame, cv2.COLOR\_BGR2HSV)

maskGreen = cv2.inRange(frameHSV, greenBajo1, greenAlto1)

maskGreenvis = cv2.bitwise\_and(frame, frame, mask= maskGreen)

cv2.imshow('Original', frame)

cv2.imshow('maskGreen', maskGreen)

cv2.imshow('maskGreenvis', maskGreenvis)

if cv2.waitKey(30) & 0xFF == ord('m'):

break

cap.release()

cv2.destroyAllWindows()

Realizamos lo mismo, pero ahora para el color verde y obtenemos lo siguiente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente