

Visión Artificial



Título: Practica 7

Nombre: Mario Gerardo Casas Miramontes

Registro: 22310165

Objetivo

Objetivo: El objetivo de esta practica es entender el funcionamiento de la eliminación de ruido dentro de un video donde se busca detectar con mayor detalle el ROI.

Funcionamiento

El funcionamiento de esta practica se basa en lo realizado en la practica número 6 en la cual se realizaron filtros de color con los cuales se buscaba destacar un color sobre los elementos de la cámara, para ello se limitaron parámetros de color con los cuales se pueda detectar la imagen, sin embargo, al momento de a detección se notaba cierto ruido en la imagen debido a la aparición de puntos negros en el área de la imagen, lo cual generaba problemas al momento de detectar un color.

Por lo que en esta practica se busco eliminar este ruido mediante la aplicación del kernel y métodos para eliminar este ruido, mediante varios métodos entre como el tophat y blackhat con los cuales se buscaba eliminar el ruido de la imagen mediante el relleno de los espacios en negro o marcando una mayor diferencia entre los métodos.

Código Fuente

#Mario Gerardo Casas Miramontes 22310165

```
import cv2
```

```
import numpy as np
```

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

```
while(1):
```

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

```
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)#Cambiamos de BGR a HSV
```

```
    #Rango de deteccion del color rojo
```

```
    lower_red = np.array([30,150,50])
```

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

```
    #Mascara tomando los rangos de color
```

```
    mask = cv2.inRange(hsv, lower_red, upper_red)
```

```
    #Aplicacion binaria
```

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

#Matriz que permite la eliminacion de ruido
kernel = np.ones((5,5), np.uint8)

#Resalta detalles
opening = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel)

#Elimina puntos blancos asilados
closing = cv2.morphologyEx(mask, cv2.MORPH_CLOSE, kernel)

#Rellena zonas negras
tophat = cv2.morphologyEx(mask, cv2.MORPH_TOPHAT, kernel)

#Resalta detalles brillantes
blackhat = cv2.morphologyEx(mask, cv2.MORPH_BLACKHAT, kernel)

#Oscurece zonas

res = cv2.bitwise_and(frame, frame, mask=closing)

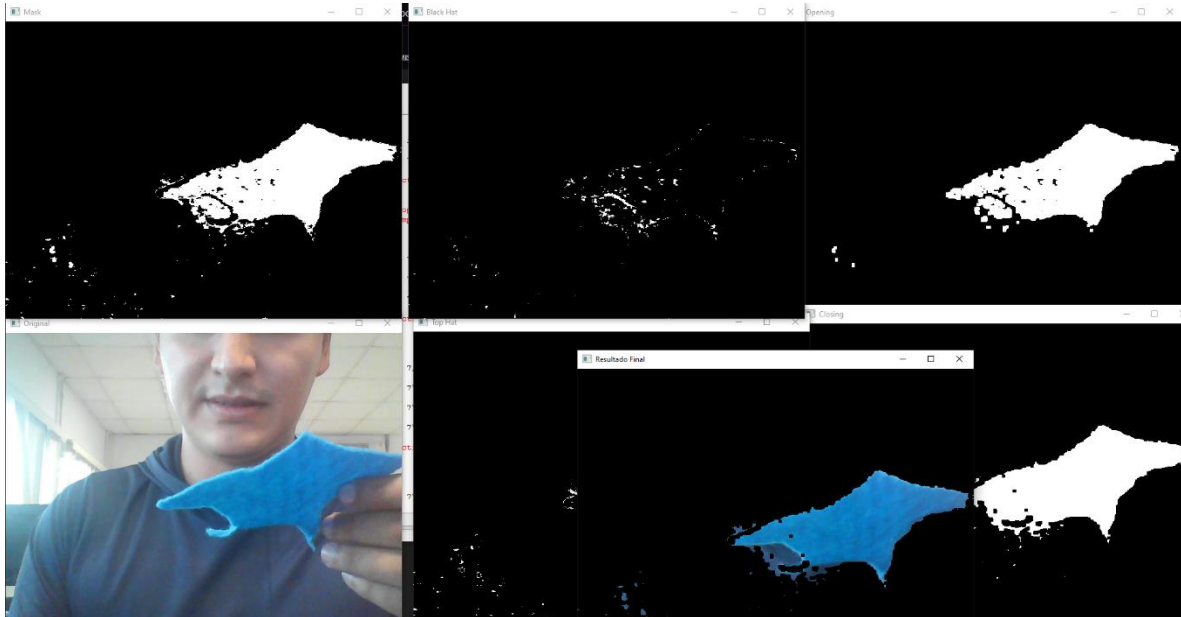

cv2.imshow('Original', frame)
cv2.imshow('Mask', mask)
cv2.imshow('Opening', opening)
cv2.imshow('Closing', closing)
cv2.imshow('Top Hat', tophat)
cv2.imshow('Black Hat', blackhat)
cv2.imshow('Resultado Final', res)


k = cv2.waitKey(5) & 0xFF
if k == 27: # Tecla ESC para salir
    break

cv2.destroyAllWindows()
```

cap.release()

Resultado



LINK de la Practica 7

<https://github.com/M-C117/VA/tree/main/Practica%207>