

# **Visión Artificial**



**Título: Practica 11**

**Nombre: Mario Gerardo Casas Miramontes**

**Registro: 22310165**

## Objetivo

Objetivo: De la imagen deseada encontrar las similitudes en otra imagen.

Objetivo 2: En VIDEO poder extraer el fondo de la imagen mediante la detección de movimiento.

## Funcionamiento

El funcionamiento de esta práctica se divide en dos, la primera parte trata de buscar coincidencias de las dos imágenes, muy a diferencia de lo realizado en la práctica 9 en donde solamente buscábamos parentescos con una imagen de referencia, en esta ocasión buscamos que la imagen pueda detectar coincidencias de puntos exactos, esto mediante el uso de los puntos clave de las imágenes, en donde llega un punto en el que se comparan y finalmente se dibujan en una imagen donde se unen los puntos clave de cada imagen siempre y cuando coincidan.

La otra parte del programa se basa en la extracción del fondo de una imagen a través de la detección de movimiento, en esta ocasión se usó la imagen de la cámara principal la cual al detectar movimiento mueve el fondo.

## Código Fuente

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

```
import numpy as np
```

```
import cv2
```

```
import matplotlib.pyplot as plt
```

```
#Analizamos las imagenes
```

```
img1 = cv2.imread('opencv-feature-matching-template.jpg',0)
```

```
img2 = cv2.imread('opencv-feature-matching-image.jpg',0)
```

```
#Usamos la funcion ORB la cual nos permite crear deteccion de puntos clave
```

```
orb = cv2.ORB_create()
```

```
#Usamos la funcion de detectandcompute para detectar los puntos clave y también para encontrar los descriptores
```

```
kp1, des1 = orb.detectAndCompute(img1,None)
```

```
kp2, des2 = orb.detectAndCompute(img2,None)
```

```
#La funcion de BFmatcher sirve para encontrar coincidencias entre cada imagen, a su vez que permite interpretar las relaciones de las imágenes con 0s y 1s
```

```
bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
```

```
#Se buscan las coincidencias entre los descriptores de las imagenes
```

```
matches = bf.match(des1,des2)
matches = sorted(matches, key = lambda x:x.distance)

#Pasamos a realizar el retiro del fondo a traves de la camara principal
cap = cv2.VideoCapture(0)
fgbg = cv2.createBackgroundSubtractorMOG2() #Con esta funcion realizacioin el retiro del fondo
#Creamos un bucle para realizar la actualizacion en el video de manera constante
while(1):
    ret, frame = cap.read()
    fgmask = fgbg.apply(frame)
    #Imprimimos los videos de la mascara con filtro retirado y el video de la camara
    cv2.imshow('fgmask',frame)
    cv2.imshow('frame',fgmask)

    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

#Finalmente usamos la funcion de draw matches con el fin de crear las coincidencias entre las dos
imagenes
#Esta funcion permite mostrar los puntos emparejados unicamente
img3 = cv2.drawMatches(img1,kp1,img2,kp2,matches[:10],None, flags=2)
plt.imshow(img3)
plt.show()

cap.release()
```

```
cv2.destroyAllWindows()
```

## Resultado

