

CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL



Igualdades con rotación y reducción de fondo.

Practica 11

Victor Alejandro Dominguez Cruz

21310228

6°G

Visión Artificial

11 de junio de 2025

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

Código:

```
import cv2
```

```
img_template = cv2.imread('template.png', 0) # Plantilla
```

```
img_scene = cv2.imread('descarga.png', 0) # Principal
```

```
orb = cv2.ORB_create() # Detector ORB
```

```
kp_template, des_template = orb.detectAndCompute(img_template, None) #  
Detectar keypoints descriptores
```

```
kp_scene, des_scene = orb.detectAndCompute(img_scene, None)
```

```
bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True) # Compara  
descriptores Brute_Force matcher
```

```
matches = bf.match(des_template, des_scene)
```

```
matches = sorted(matches, key=lambda x: x.distance) # Ordena coincidencias
```

```
resultado = cv2.drawMatches(img_template, kp_template, img_scene, kp_scene,  
matches[:20], None, flags=2) # Dibuja mejores coincidencias
```

```
cv2.imshow('Similitudes detectadas', resultado)
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

Codigo 2:

```
import cv2

cap = cv2.VideoCapture(0) # o usa 'video.mp4'

# Leer primer frame como fondo inicial
ret, fondo = cap.read()
fondo_gray = cv2.cvtColor(fondo, cv2.COLOR_BGR2GRAY)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Diferencia absoluta entre fondo y frame actual
    diff = cv2.absdiff(fondo_gray, gray)

    # Umbral para mostrar sólo movimiento
    _, thresh = cv2.threshold(diff, 30, 255, cv2.THRESH_BINARY)

    cv2.imshow('Video Original', frame)
    cv2.imshow('Movimiento detectado', thresh)

    if cv2.waitKey(1) & 0xFF == 27: # Salir con ESC
```

```
break
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

Demostración:

