



Maestro: Mauricio Alejandro Cabrera Arellano

Alumno: Nene Miranda José Said 22110319

Materia: Visión Artificial

Tarea: Practica 10

Fecha: 01-06-2025

Practica 10:

Objetivo:

De la imagen que deseen separar por medio de un ROI el fondo de la imagen dejando únicamente el ROI al cual se le buscarán todas las esquinas.

Codigo:

```
import cv2 # OpenCV para procesamiento de imágenes
import numpy as np # Librería para operaciones numéricas

# Cargar la imagen a color y convertirla a escala de grises
img_color = cv2.imread('Samus.png') # Imagen original en color
img_gray = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY) # Convertimos a escala de grises
para el análisis

# ----- Selección del ROI (Región de Interés) -----
x, y, w, h = 100, 80, 200, 200

# Coordenadas del ROI: punto (x, y), ancho (w) y alto (h)
roi_color = img_color[y:y+h, x:x+w] # Recortamos el ROI en color
roi_gray = img_gray[y:y+h, x:x+w] # Recortamos el ROI en escala de grises

# ----- Detección de esquinas con el detector de Harris -----
roi_gray = np.float32(roi_gray)

# Harris necesita que la imagen esté en formato float32
dst = cv2.cornerHarris(roi_gray, blockSize=2, ksize=3, k=0.04)

# blockSize: tamaño del vecindario considerado para detección
# ksize: tamaño del kernel de Sobel
# k: parámetro libre entre 0.04 y 0.06

# Dilatamos la imagen para que las esquinas se vean más claramente
```

```
dst = cv2.dilate(dst, None)
```

```
# Aplicamos un umbral para resaltar las esquinas (en rojo)
```

```
roi_color[dst > 0.01 * dst.max()] = [0, 0, 255]
```

```
# Las posiciones donde el valor de dst sea alto, se marcan en rojo
```

```
# ----- Mostrar solo el ROI con las esquinas detectadas -----
```

```
cv2.imshow('ROI con esquinas detectadas', roi_color)
```

```
cv2.waitKey(0) # Esperamos hasta que se presione una tecla
```

```
cv2.destroyAllWindows() # Cerramos la ventana
```

Demostración:

