

Rodrigo Hernández Zavala

7 de junio de 2025

Ing. Mecatrónica

22310215

# Detección de Esquinas

Visión Artificial

¡Perfecto! A continuación tienes un código limpio y bien estructurado que cumple exactamente con el objetivo que mencionas:

---

### **Objetivo:**

- Seleccionar un **ROI (Región de Interés)** sobre una imagen.
  - **Extraer el fondo**, dejando **solamente el ROI visible** (todo lo demás negro).
  - **Detectar esquinas** únicamente dentro del ROI.
  - Mostrar el ROI con sus esquinas resaltadas.
- 

### **CÓDIGO COMPLETO**

```
python
CopiarEditar
import cv2
import numpy as np

# ----- CONFIGURACIÓN -----

# Cambia el nombre de la imagen por la que desees probar
nombre_imagen = "caja.jpg" # Puedes cambiar a: puerta.jpg, ventana.jpg,
arquitectura.jpg

# ----- CARGAR IMAGEN -----

img = cv2.imread(nombre_imagen)
if img is None:
    print(f"Error: No se pudo cargar la imagen '{nombre_imagen}'")
    exit()

# Copia para trabajo
img_copia = img.copy()

# ----- SELECCIÓN DE ROI -----

print("Selecciona el ROI (objeto a conservar) con el mouse y presiona
ENTER o ESPACIO.")
x, y, w, h = cv2.selectROI("Seleccionar ROI", img, False, False)
cv2.destroyWindow("Seleccionar ROI")

# Crear máscara negra del tamaño de la imagen
mascara = np.zeros_like(img)

# Copiar solo el ROI a la máscara (el resto queda negro)
mascara[y:y+h, x:x+w] = img_copia[y:y+h, x:x+w]

# Convertir ROI a escala de grises para detección de esquinas
roi_gris = cv2.cvtColor(mascara[y:y+h, x:x+w], cv2.COLOR_BGR2GRAY)
roi_gris = np.float32(roi_gris)
```

```
# ----- DETECCIÓN DE ESQUINAS -----

esquinas = cv2.goodFeaturesToTrack(roi_gris, maxCorners=100,
qualityLevel=0.01, minDistance=10)




# Dibujar esquinas sobre la imagen con fondo negro
if esquinas is not None:
    for esquina in esquinas:
        cx, cy = esquina.ravel()
        cv2.circle(mascara[y:y+h, x:x+w], (int(cx), int(cy)), 4, (0, 255,
0), -1)

# ----- MOSTRAR RESULTADO -----

cv2.imshow("Imagen con fondo removido y esquinas detectadas", mascara)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

---

## ¿Qué hace este código?

Paso	Explicación
<input type="checkbox"/> Selección de ROI	Usas el mouse para marcar el objeto que quieres conservar.
 Eliminación del fondo	Todo lo que no esté en el ROI se pone en negro.
 Detección de esquinas	Se aplica <b>Shi-Tomasi</b> solo dentro del ROI.
 Visualización	Se muestran puntos verdes sobre las <b>esquinas</b> del objeto.

---

## Imágenes que puedes usar:

Asegúrate de tener estas imágenes en la misma carpeta que el código:

- caja.jpg
- puerta.jpg
- ventana.jpg
- arquitectura.jpg

Solo cambia el nombre en esta línea:

```
python
CopiarEditar
nombre_imagen = "nombre.jpg"
```

**Código:**

```
import cv2

import numpy as np

# ----- CONFIGURACIÓN -----

# Cambia el nombre de la imagen por la que desees probar

nombre_imagen = "Mosaico.jpg" # Puedes cambiar a: puerta.jpg, ventana.jpg,
arquitectura.jpg

# ----- CARGAR IMAGEN -----

img = cv2.imread(nombre_imagen)

if img is None:

    print(f"Error: No se pudo cargar la imagen '{nombre_imagen}'")

    exit()

# Copia para trabajo

img_copia = img.copy()

# ----- SELECCIÓN DE ROI -----

print("Selecciona el ROI (objeto a conservar) con el mouse y presiona ENTER o
ESPACIO.")
```

```

x, y, w, h = cv2.selectROI("Seleccionar ROI", img, False, False)

cv2.destroyWindow("Seleccionar ROI")


# Crear máscara negra del tamaño de la imagen

mascara = np.zeros_like(img)


# Copiar solo el ROI a la máscara (el resto queda negro)

mascara[y:y+h, x:x+w] = img_copia[y:y+h, x:x+w]


# Convertir ROI a escala de grises para detección de esquinas

roi_gris = cv2.cvtColor(mascara[y:y+h, x:x+w], cv2.COLOR_BGR2GRAY)

roi_gris = np.float32(roi_gris)


# ----- DETECCIÓN DE ESQUINAS -----

esquinas = cv2.goodFeaturesToTrack(roi_gris, maxCorners=100, qualityLevel=0.01,
minDistance=10)


# Dibujar esquinas sobre la imagen con fondo negro

if esquinas is not None:

    for esquina in esquinas:

        cx, cy = esquina.ravel()

        cv2.circle(mascara[y:y+h, x:x+w], (int(cx), int(cy)), 4, (0, 255, 0), -1)

```

# ----- MOSTRAR RESULTADO -----

```
cv2.imshow("Imagen con fondo removido y esquinas detectadas", mascara)
```

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

```
cv2.destroyAllWindows()
```

**Funcionamiento:**



