contours

January 31, 2024

[ ]:
```
import cv2 as cv
import numpy as np
```

Leer la imagen original desde el archivo 'cat.jpg'

[ ]:
```
img = cv.imread('Images/cat.jpg')
cv.imshow('Cat', img)
```

Crear una imagen en blanco del mismo tamaño que la imagen original

[ ]:
```
blank = np.zeros(img.shape, dtype='uint8')
cv.imshow('Blank', blank)
```

Convertir la imagen original a escala de grises

[ ]:
```
gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
cv.imshow('Gray', gray)
```

Aplicar un filtro de desenfoque gaussiano a la imagen en escala de grises

[ ]:
```
blur = cv.GaussianBlur(gray, (7,7), cv.BORDER_DEFAULT)
cv.imshow('Blur', blur)
```

Aplicar el operador de Canny para detectar bordes en la imagen suavizada

[ ]:
```
canny = cv.Canny(blur, 125, 175)
cv.imshow('Canny Edges', canny)
```

Bloque de umbralización (comentado)

ret, thresh = cv.threshold(gray, 125, 255, cv.THRESH_BINARY) .imshow('Thresh', thresh)

Encontrar contornos en la imagen de bordes Canny

[ ]:
```
contours, hierarchies = cv.findContours(canny, cv.RETR_LIST, cv.
  ↪CHAIN_APPROX_NONE)
print(f'{len(contours)}' + ' contours found!')
```

Dibujar los contornos encontrados en la imagen en blanco

[ ]:
```
cv.drawContours(blank, contours, -1, (0,0,255), 1)
cv.imshow('Contours Drawn', blank)
```

Esperar hasta que se presione una tecla antes de cerrar las ventanas

[ ]:
```
cv.waitKey(0)
```