

# Canny si contururi

December 7, 2020

Catalin Stoean

catalin.stoean@inf.ucv.ro

<http://inf.ucv.ro/~cstoean>

## 1 Canny pentru detectare de margini

```
[82]: import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('D:/poza.jpg', 0)
edges = cv2.Canny(img, 100, 200)

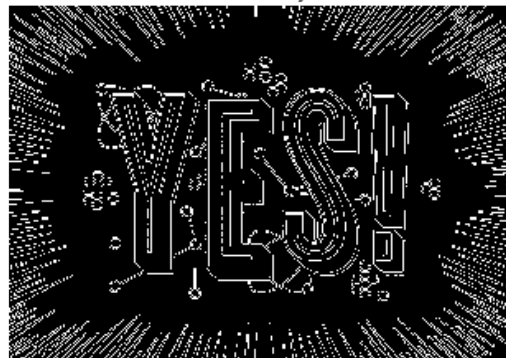
fig = plt.figure(figsize=(12, 5))
plt.subplot(121)
plt.imshow(img, cmap = 'gray')
plt.title('Imaginea grayscale')
plt.axis('off')
plt.subplot(122)
plt.imshow(edges, cmap = 'gray')
plt.title('Cu Canny')
plt.axis('off')

plt.show()
```

Imaginea grayscale



Cu Canny



## 2 Canny cu trackbar

```
[83]: import cv2

def applyCanny(_x):
    thresh1 = cv2.getTrackbarPos("thresh1", numeFereastra)
    thresh2 = cv2.getTrackbarPos("thresh2", numeFereastra)
    edges = cv2.Canny(img, thresh1, thresh2)
    cv2.imshow(numeFereastra, edges)

img = cv2.imread('D:/poza.jpg',0)
numeFereastra = 'Canny'
edges = cv2.Canny(img, 0, 0)
cv2.namedWindow(numeFereastra, cv2.WINDOW_NORMAL)
cv2.imshow(numeFereastra, edges)

cv2.createTrackbar('thresh1', numeFereastra, 0, 255, applyCanny)
cv2.createTrackbar('thresh2', numeFereastra, 0, 255, applyCanny)
cv2.waitKey(0)
```

[83]: 27

## 3 Gasire de contururi

```
[2]: import numpy as np
import cv2
from matplotlib import pyplot as plt

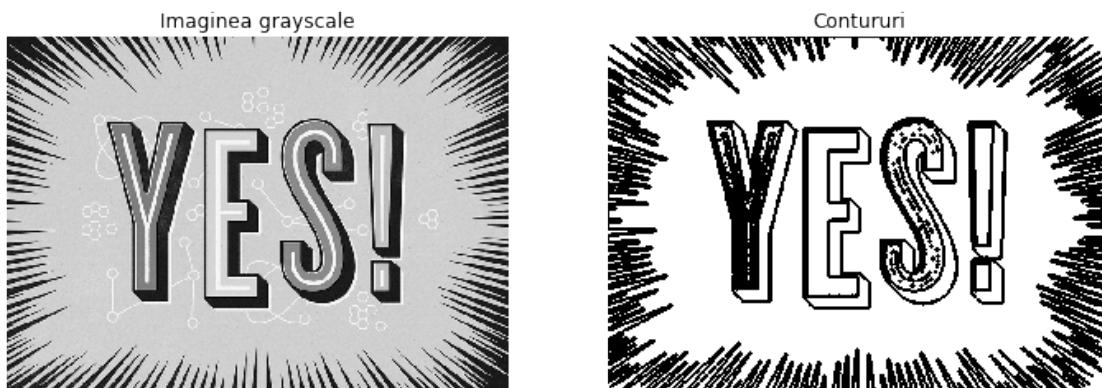
im = cv2.imread('D:/poza.jpg', 0)
_ret, thresh = cv2.threshold(im, 127, 255, cv2.THRESH_BINARY)
#plt.imshow(thresh, cmap = 'gray')
contours, hierarchy = cv2.findContours(thresh, cv2.RETR_TREE, cv2.
    ↳CHAIN_APPROX_SIMPLE)

imgContururi = np.zeros(im.shape, dtype=np.uint8)
imgContururi.fill(255) #facem toti pixelii albi
imgContururi = cv2.drawContours(imgContururi, contours, -1, 0, 3)

fig = plt.figure(figsize=(12, 5))
plt.subplot(121)
plt.imshow(im, cmap = 'gray')
plt.title('Imaginea grayscale')
```

```
plt.axis('off')
plt.subplot(122)
plt.imshow(imgContururi, cmap = 'gray')
plt.title('Contururi')
plt.axis('off')

plt.show()
```



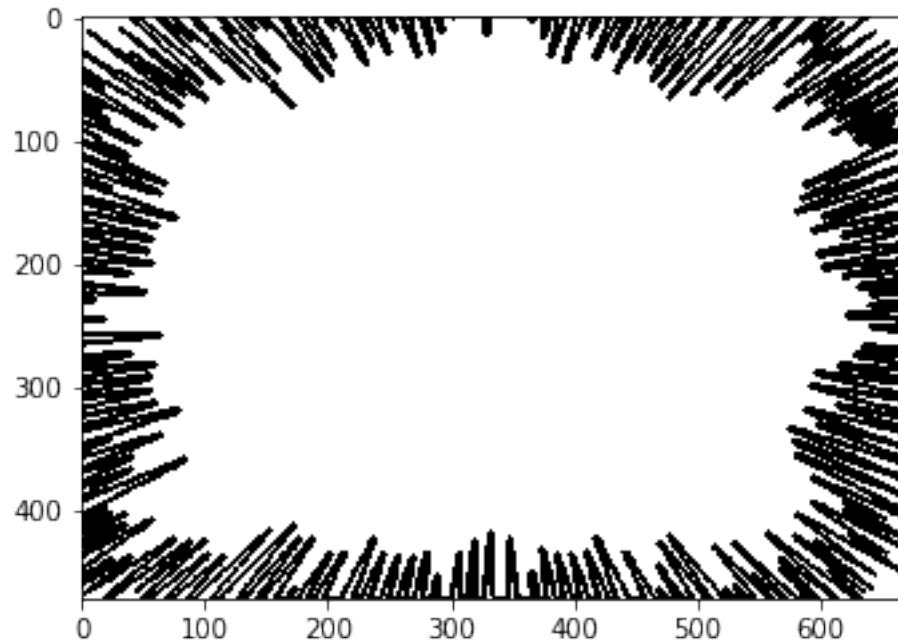
```
[2]: print('Numarul de contururi:', len(contours))
```

Numarul de contururi: 572

### 3.1 Primul contur

```
[3]: imgContururi = np.zeros(im.shape, dtype=np.uint8)
imgContururi.fill(255) #facem toti pixelii albi

#valoarea de dupa contours (0) se refera la numarul conturului de desenat
imgContururi = cv2.drawContours(imgContururi, contours, 0, 0, 3)
plt.imshow(imgContururi, cmap = 'gray')
plt.show()
```

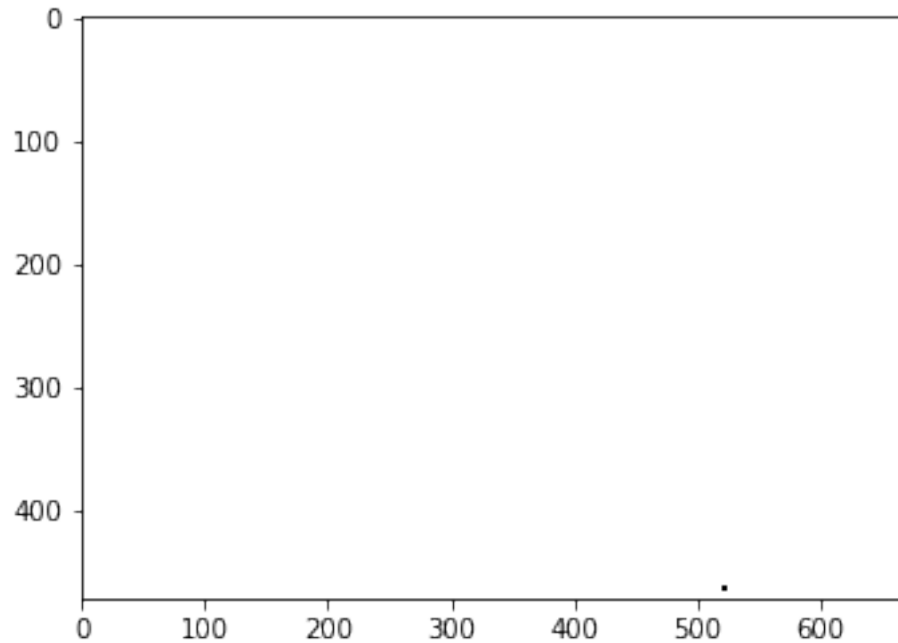


## 3.2 Al doilea contur

### 3.2.1 Vizualizare

```
[10]: imgContururi = np.zeros(im.shape, dtype=np.uint8)
imgContururi.fill(255) #facem toti pixelii albi

#valoarea de dupa contours (1) se refera la numarul conturului de desenat
imgContururi = cv2.drawContours(imgContururi, contours, 1, 0, 3)
plt.imshow(imgContururi, cmap = 'gray')
plt.show() #este o mica "pata" in dreapta jos, pe langa 500
```



### 3.2.2 Afisam punctele ce formeaza al doilea contur

```
[11]: print(contours[1])
```

```
[[[521 463]]  
 [[522 462]]  
 [[523 463]]  
 [[522 464]]]
```

### 3.2.3 Calcul arie pentru un contur

```
[19]: aria = cv2.contourArea(contours[1])  
      print(aria)
```

```
2.0
```

## 4 Gasim contururi cu arii mai mari

```
[16]: ariile = []  
      for c in contours:  
          ariile.append(cv2.contourArea(c))  
      print(ariile)
```

```
[267266.0, 2.0, 2.0, 2.0, 2.0, 2.0, 7.0, 2.0, 4.0, 2.0, 2.0, 2.0, 7.0, 2.0, 2.0, 4.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 4.0, 4.0, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 2.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 7.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 7.0, 2.0,
2.0, 2.0, 2.0, 4.0, 2.0, 7.0, 2.0, 7.0, 2.0, 10.0, 2.0, 2.0, 2.0, 7.0, 2.0, 2.0,
4.0, 4.0, 2.0, 4.0, 4.0, 4.0, 16.0, 2.0, 4.0, 2.0, 4.0, 6.0, 2.0, 2.0, 4.0,
2193.5, 720.0, 4.0, 4.0, 4.0, 16028.5, 8790.0, 7684.5, 3566.0, 4.0, 15857.0,
0.0, 0.0, 0.0, 0.5, 0.0, 1.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 2.5,
0.0, 3.0, 1.5, 0.0, 0.0, 0.0, 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 19.0, 4.0, 4.0, 0.0, 0.0, 1.0, 3.5, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.5, 0.5,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.5, 0.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.5, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 13.5, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0,
155.0, 2300.0, 2.0, 5.5, 6.0, 2.0, 2.0, 11.0, 2.0, 2.0, 8.5, 5.5, 2.0, 2.0, 2.0,
7.5, 4.0, 2.0, 4.0, 2.0, 8.5, 2.0, 4.0, 4.0, 2.0, 2.0, 2.0, 2.0, 4.0, 2.0, 4.0,
2.0, 2.0, 4.0, 2.0, 2.0, 29.5, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 4.0,
7.5, 2.0, 2.0, 4.0, 2.0, 2.0, 15.0, 5.5, 4.0, 17247.0, 9533.5, 2.0, 4.0, 2.0,
2.0, 2.0, 2.0, 4.0, 2.0, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0,
2.0, 2.0, 2.0, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 4.0, 2.0, 2.0, 2.0, 4.0, 2.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 6.0, 2.0, 2.0, 2.0, 8.0, 2.0, 4.0, 2.0, 4.0, 2.0,
2.0, 8.0, 2.0, 2.0, 2.0, 2.0, 4.0, 6.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0,
2.0, 2.0, 2.0, 2.0, 2.0, 4.0, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0,
4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 4.0, 2.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 4.0, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 4.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0,
2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0,
4.0, 4.0, 4.0, 2.0, 2.0, 2.0, 4.0, 2.0, 4.0, 4.0, 4.0, 2.0, 4.0, 2.0, 2.0, 7.0,
4.0, 2.0, 2.0, 4.0, 4.0, 7.0, 7.0, 2.0, 2.0, 2.0, 2.0, 7.0, 2.0, 2.0, 2.0,
2.0, 2.0, 2.0, 2.0, 10.0, 2.0, 2.0, 2.0, 7.0, 7.0, 2.0, 2.0, 2.0, 7.0, 7.0, 2.0,
2.0, 7.0, 2.0, 2.0, 2.0, 4.0, 2.0, 2.0, 2.0, 2.0, 5.5, 2.0, 2.0, 2.0, 2.0,
2.0, 2.0, 2.0, 2.0, 2.0, 2.0]
```

```
[17]: sortArii = sorted(ariile)
mari4 = sortArii[-4:]
print('mari4 = ', mari4)

#cautam pozitiile celor mai mari 4 contururi in ariile
pozitiiContururiMari = []
for a in mari4:
    pozitiiContururiMari.append(ariile.index(a))
print('pozitiiContururiMari = ', pozitiiContururiMari)
```

```
mari4 = [15857.0, 16028.5, 17247.0, 267266.0]
pozitiiContururiMari = [88, 83, 321, 0]
```

## 5 Afisam contururile din pozitiiContururiMari

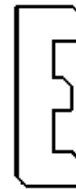
```
[46]: fig = plt.figure(figsize=(12, 10))
for i in range(0, 4):
    plt.subplot(2, 2, i + 1)
    imgContururi = np.zeros(im.shape, dtype=np.uint8)
    imgContururi.fill(255) #facem toti pixelii albi
    imgContururi = cv2.drawContours(imgContururi, contours,
    ↪pozitiiContururiMari[i], 0, 3)
    plt.imshow(imgContururi, cmap = 'gray')
    plt.title('Conturul ' + str(pozitiiContururiMari[i]))
    plt.axis('off')

plt.show()
```

Conturul 88



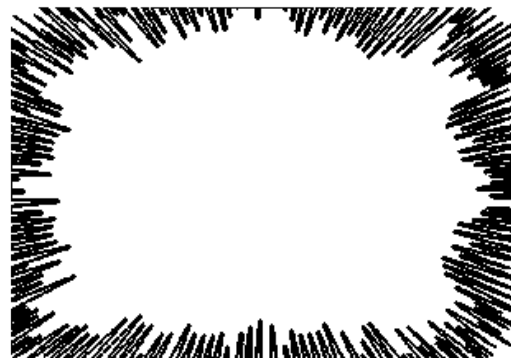
Conturul 83



Conturul 321



Conturul 0



## 6 Dreptunghiul care contine un contur

```
[21]: pozaColor = cv2.imread('D:/poza.jpg')
      pozaColor = cv2.cvtColor(pozaColor, cv2.COLOR_BGR2RGB)
      x, y, w, h = cv2.boundingRect(contours[88])
      cv2.rectangle(pozaColor, (x, y), (x + w, y + h), (255, 0, 0), 3)

      plt.imshow(pozaColor)
```

[21]: <matplotlib.image.AxesImage at 0x1726158c4c8>



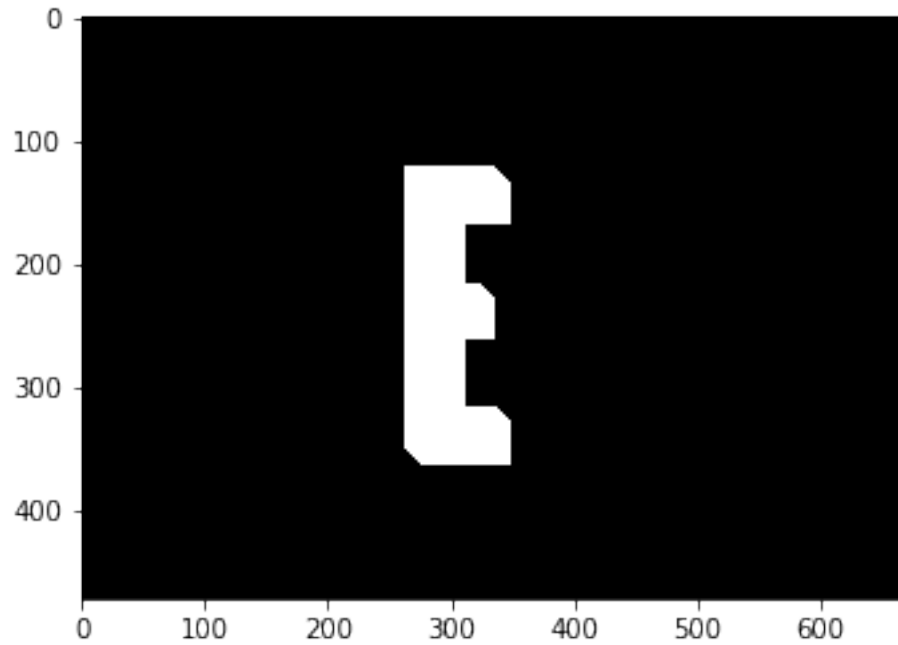
## 7 Putem face o masca cu un singur contur

```
[69]: mask = np.zeros(im.shape, np.uint8)

      cv2.drawContours(mask, contours, 83, 255, -1)
      plt.imshow(mask, cmap = 'gray')
```

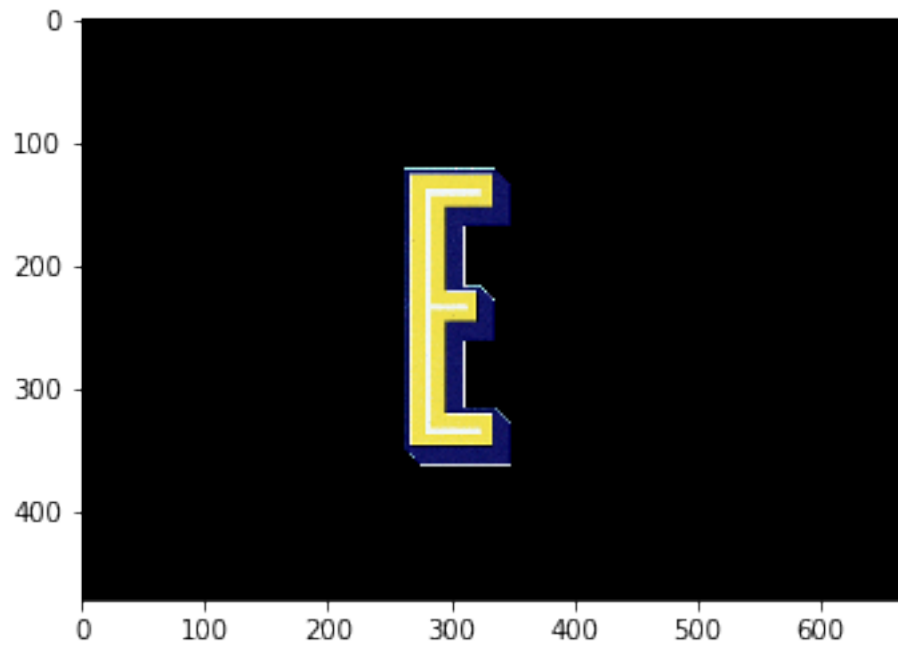
[69]: <matplotlib.image.AxesImage at 0x2445b6f3588>





```
[71]: doarE = cv2.bitwise_and(pozaColor, pozaColor, mask = mask)
plt.imshow(doarE)
```

```
[71]: <matplotlib.image.AxesImage at 0x24445b945288>
```



```
[80]: mask = np.zeros(im.shape, np.uint8)

cv2.drawContours(mask, contours, 83, 255, -1)
cv2.drawContours(mask, contours, 88, 255, -1)
cv2.drawContours(mask, contours, 321, 255, -1)

litere = cv2.bitwise_and(pozaColor, pozaColor, mask = mask)

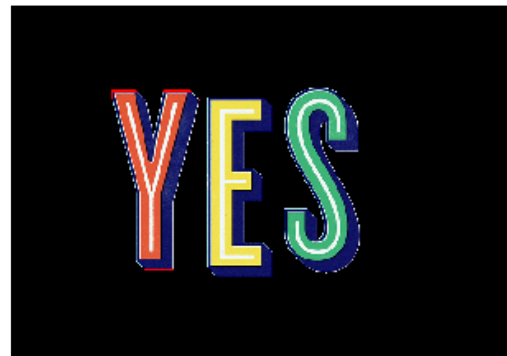
fig = plt.figure(figsize=(12, 5))
plt.subplot(121)
plt.imshow(mask, cmap = 'gray')
plt.title('Masca')
plt.axis('off')
plt.subplot(122)
plt.imshow(litere)
plt.title('Litere cu masca')
plt.axis('off')

plt.show()
```

Masca



Litere cu masca



[ ]: