

РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image_09_001

Детектори кутів. Алгоритм Моравека (Moravec)

[Підходи до побудови детекторів](#)

```
## Завантаження пакетів
import numpy as np
import matplotlib.pyplot as plt
import skimage.io as io
plt.rcParams['font.size'] = 10
```

Навчальний приклад

```
## -----
## Формуємо зображення 8 x 8 pix
## -----
win = np.ones ((3,3), dtype=np.int16)
v = np.zeros (8, dtype=np.int16)
ssd = np.zeros (8, dtype=np.float32)
displ_x = [-1, 0, 1,-1, 1,-1, 0, 1]
displ_y = [-1,-1,-1, 0, 0, 1, 1, 1]

rows_num = 10
clms_num = 10
test_im = np.zeros ((rows_num, clms_num), dtype=np.int16)

for i in range (5):
    for j in range (5) :
        test_im [i][j] = 200

print('Тестове зображення')
print(test_im)
```

```
Тестове зображення
[[200 200 200 200 200   0   0   0   0   0]
 [200 200 200 200 200   0   0   0   0   0]
 [200 200 200 200 200   0   0   0   0   0]
 [200 200 200 200 200   0   0   0   0   0]
 [200 200 200 200 200   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]]
```

Функція обчислення "Сили Кута"

```
def cornerness (i_pix, j_pix, img):  
    '''  
    !!! i_pix, j_pix - піксель інтересу  
    !!! розміри зображення (визначені глобально)  
    !!! displ_x, displ_y - напрями зміщення (визначені глобально)  
    !!! повернає мінімальне SSD для заданого пікселя  
    '''  
  
    #print (img)  
    #print (i_pix,i_pix)  
    ssd = np.zeros (8, dtype=np.float32)  
    s_pix = 0  
    for i in range (3):  
        for j in range (3):  
            s_pix += img [i_pix+i-1,j_pix+j-1]  
  
    for k in range (8):  
        s = 0.0  
        for i in range (3):  
            for j in range (3):  
                s += img[i_pix+displ_x[k]+i-1,j_pix+displ_y[k]+j-1]  
        ssd[k] = (s_pix-s)*(s_pix-s)  
    # print (ssd)  
    return np.min(ssd)
```

Тест функції

```
print (cornerness (4, 4, test_img))
```

40000.0

Завантажуємо зображення

```
%%script false  
## Загрузка файла  
test_img = np.zeros ((50, 50, 3), dtype=np.uint8)  
  
## Визначення структури та розміру зображення  
print ('IMAGE SHAPE', test_img.shape, 'IMAGE SIZE', test_img.size)  
  
irows_num = test_img.shape[0] ## кількість рядків  
iclms_num = test_img.shape[1] ## кількість колонок  
print ('ROWS NUMBER', irows_num, 'CLMS NUMBER', iclms_num)  
  
for i in range (20,30,1):  
    for j in range (20,30,1):  
        test_img[i, j, :] = [200,200,200]  
  
fig, ax = plt.subplots(figsize=(4, 4))  
plt.imshow(test_img )  
plt.title("Тестове зображення")
```

```
plt.show()
```

```
Couldn't find program: 'false'
```

```
## %%script false
# Завантаження зображення
path = './IMAGES/'
filename = 'Lenna.png'
# filename = 'Face_1_300_x_400.jpg'
test_img = io.imread(path+filename)
# Визначення структури та розміру зображення
print('---- IMAGE 1 ----')
print('IMAGE SHAPE', test_img.shape, 'IMAGE SIZE', test_img.size)
irows_num = test_img.shape[0] # кількість рядків
iclms_num = test_img.shape[1] # кількість колонок
print('ROWS NUMBER', irows_num, 'CLMS NUMBER', iclms_num)
fig, ax = plt.subplots(figsize=(4, 4))
plt.imshow(test_img)
plt.title("Тестове зображення")
plt.show()
```

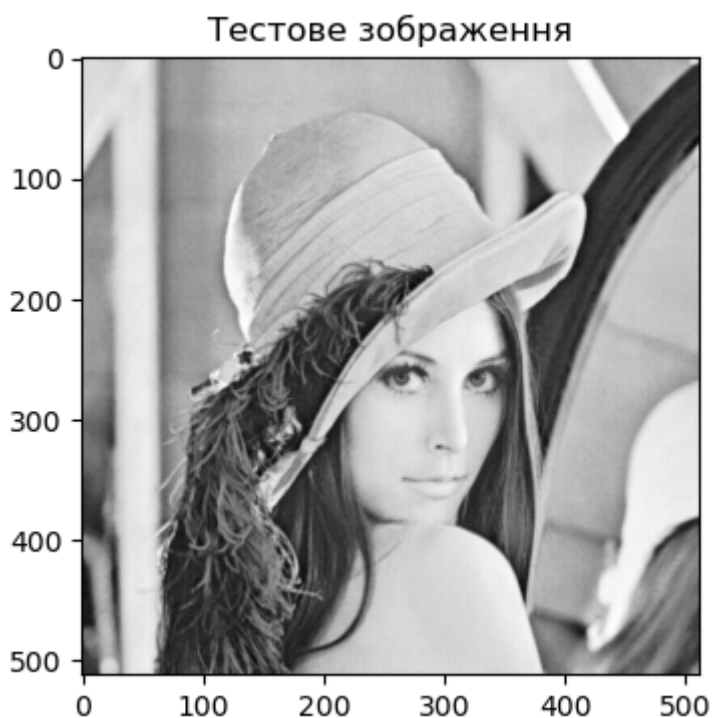
```
---- IMAGE 1 ----
IMAGE SHAPE (512, 512, 3) IMAGE SIZE 786432
ROWS NUMBER 512 CLMS NUMBER 512
```



Використовуємо тільки одну кольорову компоненту

```
test_image = np.zeros ((irows_num, iclms_num), dtype=np.uint8)
for i in range (irows_num):
    for j in range (iclms_num):
        test_image[i,j] = test_img [i,j,0]

#print ('Просто перевірка', cornerness (20, 20, test_image))
fig, ax = plt.subplots(figsize=(4, 4))
plt.imshow(test_image, 'gray' )
plt.title("Тестове зображення")
plt.show()
```



Обчислюємо SSD для всіх пікселів зображення

```
points_ = np.zeros ((irows_num, iclms_num), dtype=np.float32)
```

```
for i in range (2, irows_num-2, 1):
    for j in range (2, iclms_num-2, 1):
        points_ [i][j] = cornerness (i, j, test_image)
print ('-- END ---')
```

```
-- END ---
```

```
print ('Перевірка', points_ [20][20])
```

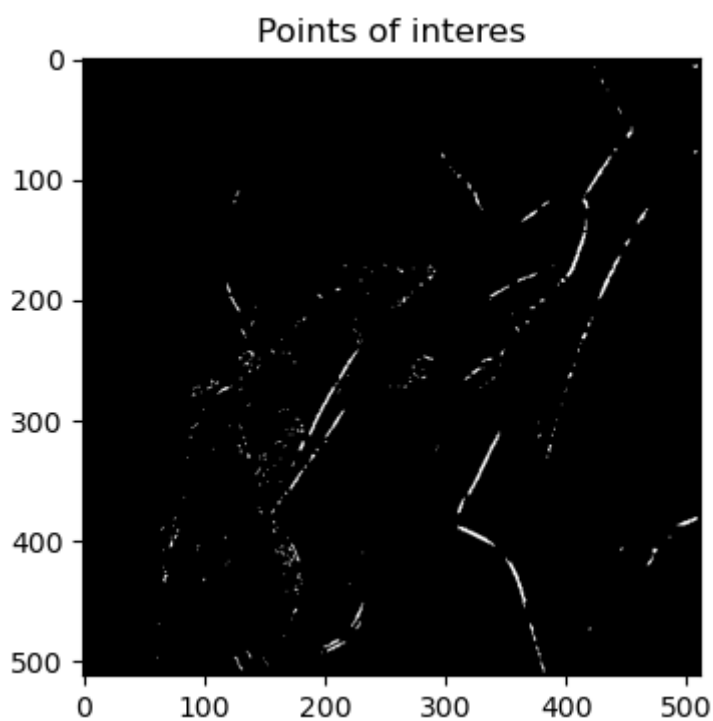
```
перевірка 0.0
```

```

# Формуємо зображення кутів
threshold = 5000 # поріг (зріз)
points_im = np.zeros ((irows_num, iclms_num, 3), dtype=np.uint8)
for i in range (irows_num):
    for j in range (iclms_num):
        if points_ [i,j] > threshold :
            points_im [i,j,:] = [255,255,255]

fig, ax = plt.subplots(figsize=(4, 4))
plt.imshow(points_im)
plt.title("Points of interes")
plt.show()

```



```

# Формуємо зображення кутів на відному зображення
test_im_points = test_img.copy()

print (test_im_points.shape)

for i in range (irows_num):
    for j in range (iclms_num):
        if points_ [i,j] > threshold :
            test_im_points [i,j,:] = [255,0,0]
            test_im_points [i+1,j,:] = [255,0,0]
            test_im_points [i-1,j,:] = [255,0,0]
            test_im_points [i,j+1,:] = [255,0,0]
            test_im_points [i,j-1,:] = [255,0,0]

```

```
fig, ax = plt.subplots(figsize=(4, 4))  
plt.imshow(test_im_points)  
plt.title("Points of interes")  
plt.show()
```

(512, 512, 3)

