

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**«Бинарные изображения, основные характеристики бинарных  
изображений»**

**Отчет по лабораторной работе № 9**

**по дисциплине «Технологии распознавания образов»**

Выполнил студент группы ПИЖ-б-о-21-1

Гасанов Г. М. « » 2023г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2023

Цель работы: изучение методов цифровой обработки бинарных изображений, геометрических характеристик этих изображений, способов получения дополнительных параметров бинарных изображений. Изучение основных функций OpenCv, применяемых для цифровой обработки бинарных изображений.

Ход работы:

Задание 3.1. Вычислить площадь  $s$ , периметр  $p$ , ширину  $w$ , высоту  $h$ , отношение ширины к высоте  $w/h$ , отношение площади изображения к площади описывающего прямоугольника  $s/(wh)$ , эквивалентный диаметр, центр масс, моменты бинарного изображения.

## Лабораторная работа №9

### Бинарные изображения, основные характеристики бинарных изображений

#### Задание 3.1.

Вычислить площадь  $s$ , периметр  $p$ , ширину  $w$ , высоту  $h$ , отношение ширины к высоте  $w/h$ , отношение площади изображения к площади описывающего прямоугольника  $s/(wh)$ , эквивалентный диаметр, центр масс, моменты бинарного изображения.

```
In [18]: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
In [19]: img = cv2.imread('cat.jpg',0)
imgag = cv2.imread('cat.jpg',0)
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.axis("off")
```

```
Out[19]: (-0.5, 639.5, 461.5, -0.5)
```



```
In [20]: ret, thresh = cv2.threshold(img, 0, 255, 0)
contours, hierarchy = cv2.findContours(thresh, 5, 5)
plt.imshow(cv2.cvtColor(thresh, cv2.COLOR_BGR2RGB))
plt.axis("off")
```

```
Out[20]: (-0.5, 639.5, 461.5, -0.5)
```



Рисунок 1 – Результат работы

Индивидуальное задание.

Считать цветное изображение, найти его контуры, вывести:

1. Все контуры
2. Половину контуров

### 3. Четверть контуров

### 4. Самый длинный контур

#### Задание

Считать целое изображение, найти его контуры, вывести:

1. Все контуры
2. Половину контуров
3. Четверть контуров
4. Самый длинный контур

```
In [1]: import cv2
import numpy as np
import matplotlib.pyplot as plt

In [2]: img = cv2.imread('cat.jpg', 0)
ret,thresh = cv2.threshold(img,128,255, cv2.THRESH_BINARY)
contours, hierarchy = cv2.findContours(thresh, 5, 5)

plt.axis('off')
plt.title("Оригинал")
plt.imshow(img, cmap="gray");
```

Оригинал



```
In [3]: # Количество контуров
len_cont = len(contours)
print("Количество контуров: ", len_cont)

Количество контуров: 618
```

#### Найдем все контуры

```
In [4]: mask = np.zeros(img.shape,np.uint8)
all_cont = cv2.drawContours(mask,contours,-1,255,2)
```

#### Найдем половину контуров

```
In [5]: i = 0
cnt = []
while len(cnt) < len_cont/2:
    cnt.append(contours[i])
    i += 1
```

```
In [6]: mask = np.zeros(img.shape,np.uint8)
half_count = cv2.drawContours(mask,cnt,-1,255,2)
plt.imshow(mask, cmap="gray");
```

#### Найдем четверть контуров

```
In [7]: i = 0
cnt = []
while len(cnt) < len_cont/4:
    cnt.append(contours[i])
    i += 1
```

#### Найдем самый длинный контур

```
In [8]: mask = np.zeros(img.shape,np.uint8)
four_count = cv2.drawContours(mask,cnt,-1,255,2)
```

```
In [9]: mask = np.zeros(img.shape,np.uint8)
max=0
x1,countour=None
for countour in contours:
    if countour.shape[0]>max:
        x1,countour=countour
        max=countour.shape[0]
max_cont = cv2.drawContours(mask, [x1,countour], -1, (255,255,255), 2)
```

#### Организуем вывод изображений

```
In [10]: fig, ax = plt.subplots(2, 2, figsize=(10,7))
fig.tight_layout()
x1figure = ["Самый длинный контур", "Все контуры", "Половина контуров", "Четверть контуров"]
```

Рисунок 2 – Результат работы