

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Нахождение и обработка контуров»

ОТЧЕТ
по лабораторной работе №13
дисциплины
«Технологии распознавания образов»

Выполнил:
Мизин Глеб Егорович
2 курс, группа ПИЖ-б-о-21-1,
011.03.04 «Программная инженерия»,
направленность (профиль) «Разработка
и сопровождение программного
обеспечения», очная форма обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

```
In [48]: image_blur = cv2.medianBlur(gray, 5)
        thresh = cv2.adaptiveThreshold(image_blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)
```

Определим контуры

```
In [49]: contours, hierarchy = cv2.findContours(thresh.copy(),cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)
```

Нарисуем все найденные контуры

```
In [50]: img_with_contours = image_rgb.copy()
        for contour in contours:
            cv2.drawContours(img_with_contours, contour, -1, (0,255,0), 2)
```

Выведем

```
In [51]: res1 = [gray, thresh]
        title1 = ['Оригинал Ч/Б', 'Контуры']

        img_print(res1, title1, hard_type='gray')

        res2 = [image_rgb, img_with_contours]
        title2 = ['Оригинал RGB', 'Контуры']

        img_print(res2, title2)
```



Задание 7.2. Протестировать функцию поиска контура cv2.findContours с аргументом cv2.CHAIN_APPROX_SIMPLE, который экономит память.

```
In [52]: image_rgb = img_input('pic/jojo3.jpg')
        gray = cv2.cvtColor(image_rgb, cv2.COLOR_BGR2GRAY)

        image_blur = cv2.medianBlur(gray, 5)
        thresh = cv2.adaptiveThreshold(image_blur,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY,11,2)

        contours, hierarchy = cv2.findContours(thresh.copy(),cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)
        img_with_contours = image_rgb.copy()

        for contour in contours:
            cv2.drawContours(img_with_contours, contour, -1, (0,255,0), 2)

        res1 = [gray, thresh]
        title1 = ['Оригинал Ч/Б', 'Контуры']

        img_print(res1, title1, hard_type='gray')

        res2 = [image_rgb, img_with_contours]
        title2 = ['Оригинал RGB', 'Контуры']

        img_print(res2, title2)
```



Задание 7.3 Выделить границу методом Канни

Задание 7.3 Выделить границу методом Канни

```
In [53]: image = img_input('pic/jojo4.jpg')  
edges = cv2.Canny(image, 47, 0, apertureSize = 3)  
res = [image, edges]  
title = ['Оригинал', 'Границы по Канни']  
img_print(res, title)
```

Оригинал



Границы по Канни



Контрольные вопросы

1. Какие параметры выводит функция `cv2.findContours()`?

Функция `cv2.findContours()` выводит массив изображения, массив контуров и иерархию контуров.

2. Каковы параметры функции `cv2.drawContours()`?

Первым аргументом функции `cv2.drawContours()` является исходное изображение, вторым аргументом являются контуры, третьим аргументом является индекс контуров.

3. Каким образом можно нарисовать все контуры изображения?

Чтобы нарисовать все контуры изображения, нужно использовать функцию `img = cv2.drawContours(img, contours, -1, (0,255,0),3)`.

4. Как изменить параметры метода аппроксимации контура в функции `cv2.findContours()`?

Для изменения параметров метода аппроксимации контура в функции `cv2.findContours()` нужно указать третий аргумент, который определяет метод аппроксимации. Доступны следующие методы

`cv2.CHAIN_APPROX_NONE`,
`cv2.CHAIN_APPROX_SIMPLE`,
`cv2.CHAIN_APPROX_TC89_L1`,
`cv2.CHAIN_APPROX_TC89_KCOS`.

5. Почему хранение всех координат пикселей контура не рационально?

Хранение всех координат пикселей контура занимает много памяти, что неэффективно.

6. Каким образом можно сохранить только начальную и конечную координаты отрезка, если контур включает в себя прямые отрезки?

Для сохранения только начальной и конечной координаты отрезка в контуре можно использовать метод аппроксимации `cv2.CHAIN_APPROX_SIMPLE` в функции `cv2.findContours()`

7. Что означает аргумент `cv2.CHAIN_APPROX_SIMPLE` в функции `cv2.findContours()`?

Аргумент `cv2.CHAIN_APPROX_SIMPLE` в функции `cv2.findContours()` позволяет удалять все лишние точки в контуре и сжимать его, что экономит память.

8. Каким образом работает метод Канни?

Метод Канни обнаруживает границы связанной области изображения, выполняя поиск локальных максимумов градиента $f(x, y)$. Градиент вычисляется после применения фильтра Гаусса.

9. Какие аргументы принимает функция Канни `cv2.Canny()`?

Функция Канни `cv2.Canny(img, T_lower, T_upper, apertureSize = 3)` содержит в скобках следующие аргументы: `img` – входное изображение, `T_lower`: нижний порог, `T_upper`: верхний порог, `apertureSize` – размер апертюры оператора Собеля.

10. Каковы этапы алгоритма Канни?

Предварительное сглаживание по Гауссу для уменьшения шума, вычисление амплитуды и направления градиента изображения, обычно выбирается один из четырех возможных углов: 0° , 45° , 90° и 135° , не максимальное подавление: исключаются некраевые пиксели, это позволяет утончать края, использование порогового гистерезиса, для которого необходимы два порога: высокий и низкий порог. Если амплитуда градиента исследуемых пикселей больше, чем значения высокого порога, то эти пиксели резервируются как краевые пиксели. Если амплитуда положения пикселя меньше нижнего порога, то пиксель исключается. Если амплитуда интенсивности пикселя находится между двумя порогами, пиксель сохраняется только тогда, когда он подключен к пикселю выше верхнего порога.