

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра Систем автоматического управления**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Техническое зрение»**  
**Тема: «МОРФОЛОГИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ»**

Студент гр. 1492

Старцев Н.А.

Преподаватель

Федоркова А.О.

Санкт-Петербург

2024

**Цель работы:** научиться применять морфологические преобразования для исправления изображений и видео

**Задание 1** Напишите свою реализацию операций замыкания и размыкания для  $n$ -ного количества итераций. Каждую операцию оформите в виде отдельной функции (допустимый вариант - одна функция для двух операций, где тип операции задается через аргумент). Функция должна принимать исходное изображение, ядро, количество итераций и другие аргументы при необходимости. В вашей реализации замыкания и размыкания нужно использовать функции эрозии и наращивания, реализованные в OpenCV. На изображении 4-1 добейтесь полной заливки кругов с использованием первой функции. С помощью операции размыкания удалите всё содержимое кругов на исходном изображении, затем сохраните его на компьютер. Сравните результат работы ваших функций с реализацией замыкания и размыкания в OpenCV. Для сравнения результата можно использовать попиксельное сравнение изображений. (спойлер – если результаты отличаются, значит, Вы сделали что-то не то)

**Задание 2** Для выполнения этого задания используйте изображение 4-2. Примените морфологические преобразования к такому изображению так, чтобы максимально убрать шум в виде точек на фоне и при этом закрыть разрывы в линиях текста.

.

Примеры работы программы:

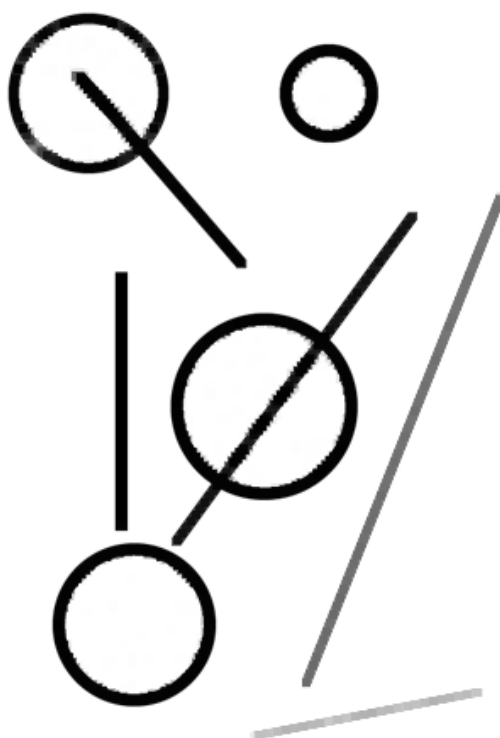


Рисунок 1 – Результат работы программы, реализующей размыкание

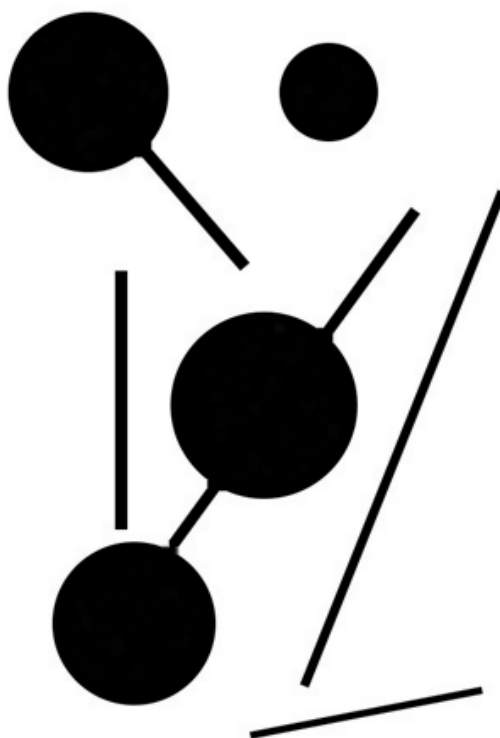


Рисунок 2 – Результат работы программы, реализующей замыкание.

topped by an almost perfectly symmetrical  
leaves. The ceaseless beat of minute waves  
the rocks I sit on, make me increase  
over and <sup>renewed</sup> I saw my dreams today...  
ling hills, covered with small yellow  
flowers, cut off abruptly, as cliffs in  
a sea. I am in awe. Looking off  
to the distance... snow-capped  
the sky, beautifully ~~rolling~~ smoothly  
smeared clouds. Vaguely appearing like  
onshore in ~~the distance~~ light. The sun  
is still out, and is beginning to set. The  
now on the Orosno Volcano turns orange, and  
see the sky on the opposite horizon. As  
I walk back through the Plaza, I see the  
his must be the hang-out area for the  
young people at night. At the same time it  
is where the older generation takes its walks.  
Interesting and unusual combination. <sup>That</sup>

Рисунок 3 – Обработка изображения при помощи морфологических преобразований.

**Вывод.**

В ходе лабораторной работы были изучены морфологические преобразования. Они позволяют удалить шумы с изображения, или наоборот выровнять изображения, например потертости разметки или букв на вывеске.

## Результаты работы

### Код программы:

#### Задание 1

```
import cv2
import numpy

def nothing(x):
    pass

def my_func(type, img, size, iterations):
    if type == "OPEN":
        img_new = cv2.dilate(
            cv2.erode(
                img,
                kernel=numpy.ones((size, size), dtype=int),
                # kernel = ker,
                anchor=(-1, -1),
                iterations=iterations,
                borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255)),
            kernel=numpy.ones((size, size), dtype=int),
            # kernel = ker,
            anchor=(-1, -1),
            iterations=iterations,
            borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255))
    elif type == "CLOSE":
        img_new = cv2.erode(
            cv2.dilate(
                img,
                kernel=numpy.ones((size, size), dtype=int),
                # kernel = ker,
                anchor=(-1, -1),
                iterations=iterations,
                borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255)),
            kernel=numpy.ones((size, size), dtype=int),
            # kernel = ker,
            anchor=(-1, -1),
            iterations=iterations,
            borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255))
    else:
        img_new = img
        print("error")
    return img_new

pathImg1 = "./lab4/4-1.jpg"
pathImg2 = "./lab4/4-2.jpg"
# pathTestImg = "./lab4/test2.jpeg"
winName = "test_window"

cv2.namedWindow(winName, cv2.WINDOW_GUI_NORMAL)

img = cv2.imread(pathImg1, flags=cv2.IMREAD_COLOR)

cv2.createTrackbar("size", winName, 0, 10, nothing)
cv2.createTrackbar("iterations", winName, 0, 10, nothing)
cv2.createTrackbar("type", winName, 0, 1, nothing)
i = 1
my_type = ["OPEN", "CLOSE"]
morph_type = [cv2.MORPH_OPEN, cv2.MORPH_CLOSE,
               cv2.MORPH_GRADIENT, cv2.MORPH_TOPHAT, cv2.MORPH_BLACKHAT]
while 1:
    size = cv2.getTrackbarPos("size", winName)+1
    iter = cv2.getTrackbarPos("iterations", winName)+1
    t = cv2.getTrackbarPos("type", winName)
    img_new = my_func(my_type[t], img, size, iter)
    img_new_2 = cv2.morphologyEx(img, morph_type[t], kernel=numpy.ones((size, size), dtype=int),
                                anchor=(-1, -1),
                                iterations=iter,
                                borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255),
                                )
    # img_new -=img_new_2
    cv2.imshow(winName, img_new)
    key = cv2.waitKey(100)
    if key == 27: break
```

## Задание 2

```
import cv2
import numpy

def nothing(x):
    pass

pathImg1 = "./lab4/4-1.jpg"
pathImg2 = "./lab4/4-2.jpg"
winName = "test_window"

cv2.namedWindow(winName, cv2.WINDOW_GUI_NORMAL)

img = cv2.imread(pathImg2, flags=cv2.IMREAD_COLOR)

cv2.createTrackbar("size", winName, 0, 10, nothing)
cv2.createTrackbar("iterations", winName, 0, 10, nothing)
cv2.createTrackbar("type", winName, 0, 4, nothing)
i = 1
my_type = ["OPEN", "CLOSE"]
morph_type = [cv2.MORPH_OPEN, cv2.MORPH_CLOSE,
               cv2.MORPH_GRADIENT, cv2.MORPH_TOPHAT, cv2.MORPH_BLACKHAT]
while 1:
    # ker = numpy.full((7,7), 1)
    size = cv2.getTrackbarPos("size", winName)+1
    iter = cv2.getTrackbarPos("iterations", winName)+1
    t = cv2.getTrackbarPos("type", winName)

    img_new = cv2.morphologyEx(img, morph_type[t], kernel=numpy.ones((size, size), dtype=int),
                               anchor=(-1, -1),
                               iterations=iter,
                               borderType=cv2.BORDER_CONSTANT, borderValue=(255, 255, 255),
                               )

    cv2.imshow(winName, img_new)
    key = cv2.waitKey(100)
    if key == 27:
        break
```