**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САУ**

отчет

**по лабораторной работе №4**

**по дисциплине «Техническое зрение»**

**Тема: «Пороговые фильтры»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6491 |  | Бабурин А. Д. |
| Преподаватель |  | Моклева К.А. |

Санкт-Петербург

2020 г.

**Цель работы**: изучить принцип применения пороговых фильтров для

обработки изображений.

**Задание**:

1. Напишите свою “легкую” реализацию cv2.threshold() только для

варианта THRESH\_BINARY. Функция должна принимать значение

threshold. Пусть maxVal по умолчанию всегда будет 255.

1. Примените на практике бинаризацию всех типов для следующих

изображений (нескольких каждого типа):

a) фото дорожной разметки белого цвета;

b) изображение отсканированного текстового документа;

c) фото написанного от руки или напечатанного текста.

Для каждого типа опишите, какой результат был получен, насколько

конкретный тип пороговой фильтрации для него применим и почему.

**Код программы:**

**import** cv2.cv2 **as** cv2

**import** numpy

**def** FUN(img, treshold):

**for** i **in** range(len(img)):

**for** j **in** range(len(img[i])):

**if** img[i][j] > treshold:

img[i][j] = 255

**else**:

img[i][j] = 0

**return** img

IMG = cv2.imread('D:/LAB\_PYTHON/LAB\_4/test.jpeg', cv2.IMREAD\_GRAYSCALE)

IMG = FUN(IMG, 50)

cv2.imshow('F', IMG)

cv2.imwrite('D:/LAB\_PYTHON/LAB\_4/test\_F.jpeg',IMG)

road1 = cv2.imread('D:/LAB\_PYTHON/LAB\_4/R\_1.jpeg', cv2.IMREAD\_REDUCED\_GRAYSCALE\_2)

new\_threshold, new\_road1 = cv2.threshold(road1, 200, 255, cv2.THRESH\_BINARY)

cv2.imshow('R1', new\_road1)

road2 = cv2.imread('D:/LAB\_PYTHON/LAB\_4/R\_2.jpeg', cv2.IMREAD\_REDUCED\_GRAYSCALE\_8)

new\_road2 = cv2.adaptiveThreshold(road2, 255, cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C, cv2.THRESH\_BINARY, 701, -49)

cv2.imshow('R2', new\_road2)

SC1 = cv2.imread('D:/LAB\_PYTHON/LAB\_4/scan\_2.png', cv2.IMREAD\_GRAYSCALE)

new\_SC1 = cv2.adaptiveThreshold(SC1, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY, 25, 20)

cv2.imshow('SCAN\_1', new\_SC1)

SC2 = cv2.imread('D:/LAB\_PYTHON/LAB\_4/T\_1.jpeg', cv2.IMREAD\_GRAYSCALE)

new\_threshold, new\_SC2 = cv2.threshold(SC2, 180, 255, cv2.THRESH\_BINARY)

cv2.imshow('SCAN\_2', new\_SC2)

FT1 = cv2.imread('D:/LAB\_PYTHON/LAB\_4/T\_2.jpeg', cv2.IMREAD\_GRAYSCALE)

new\_threshold, new\_FT1 = cv2.threshold(FT1, 180, 255, cv2.THRESH\_OTSU)

cv2.imshow('FT1', new\_FT1)

SA\_T = cv2.imread('D:/LAB\_PYTHON/LAB\_4/s\_t.jpg', cv2.IMREAD\_REDUCED\_GRAYSCALE\_4)

new\_SA\_T = cv2.adaptiveThreshold(SA\_T, 255, cv2.ADAPTIVE\_THRESH\_MEAN\_C, cv2.THRESH\_BINARY, 45, 20)

cv2.imshow('SA\_T', new\_SA\_T)

cv2.waitKey(-1)

cv2.destroyAllWindows()

**Результаты:**

1. maxVal = 50

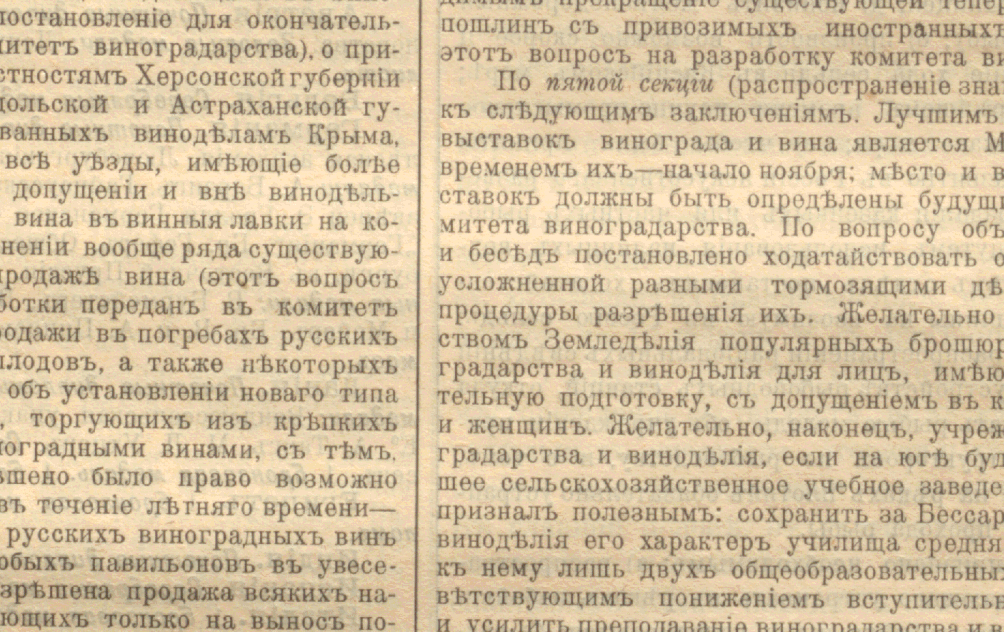
|  |  |
| --- | --- |
|  |  |
| Исходное изображение | TRESH\_BINARY |

1. Фото дорожной разметки белого цвета
   * Применена бинаризация TRESH\_BINARY, это обусловлено “равномерностью” изображения – отсутствие теней или засвеченных участков.

|  |  |
| --- | --- |
|  |  |
| Исходное изображение | TRESH\_BINARY |

* + Бинаризация с адаптивным вычислением порога. На изображении имеются области тени и засвета.

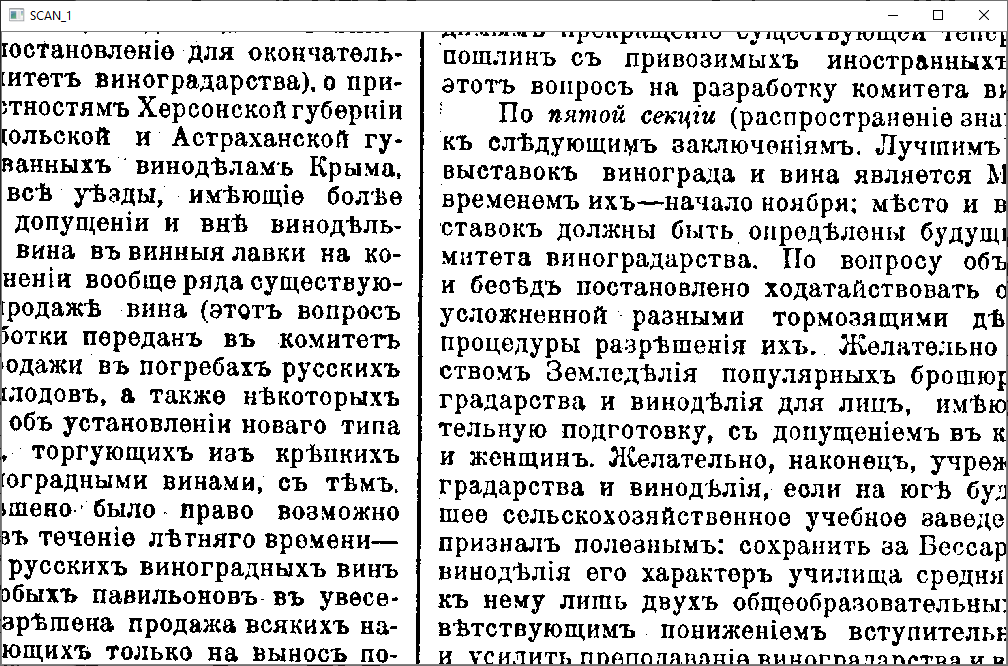
|  |  |
| --- | --- |
|  |  |
| Исходное изображение | adaptiveThreshold  ADAPTIVE\_THRESH\_GAUSSIAN\_C |

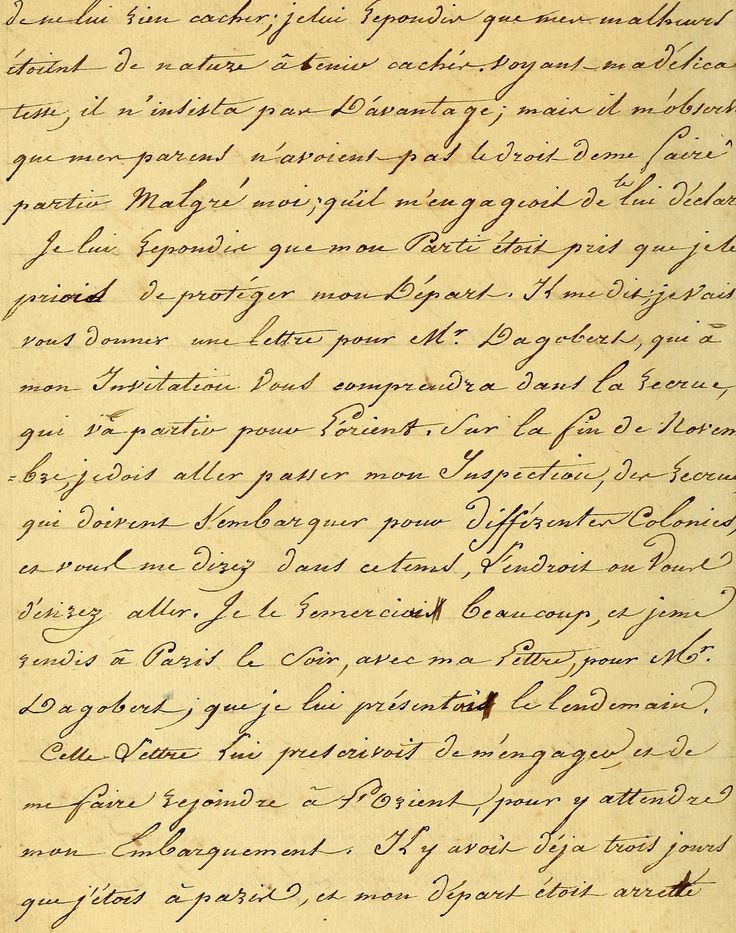
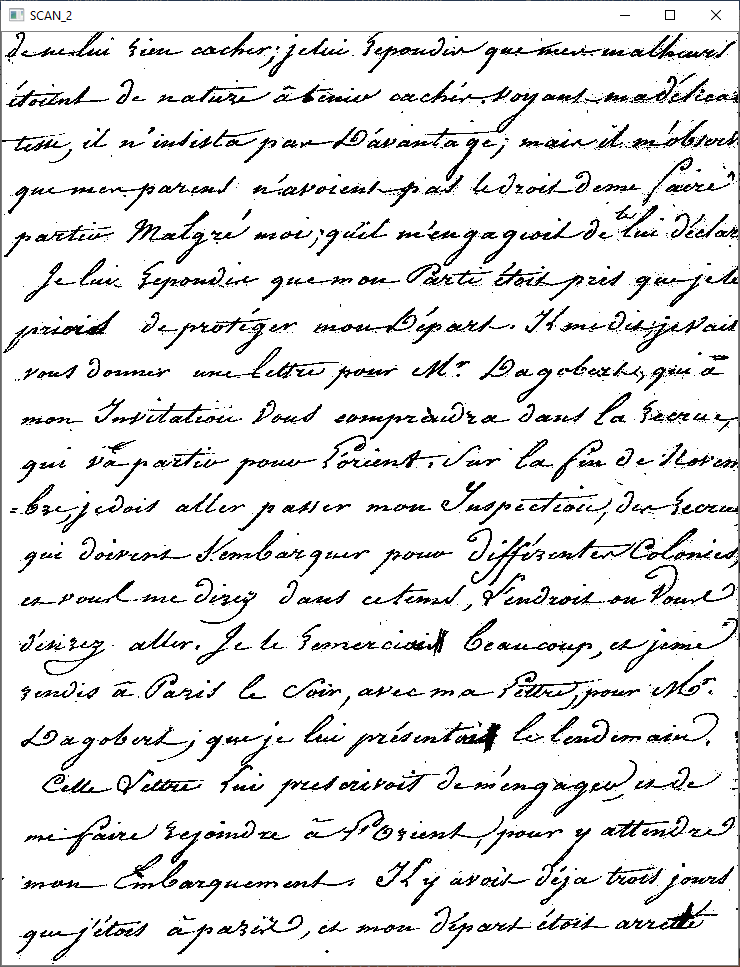
1. Изображение отсканированного текстового документа
   * Бинаризация с адаптивным вычислением порога (пиксели с одинаковым весом в области). На изображении просвечивает текст с обратной стороны листа.

Исходное изображение

adaptiveThreshold

ADAPTIVE\_THRESH\_MEAN\_C

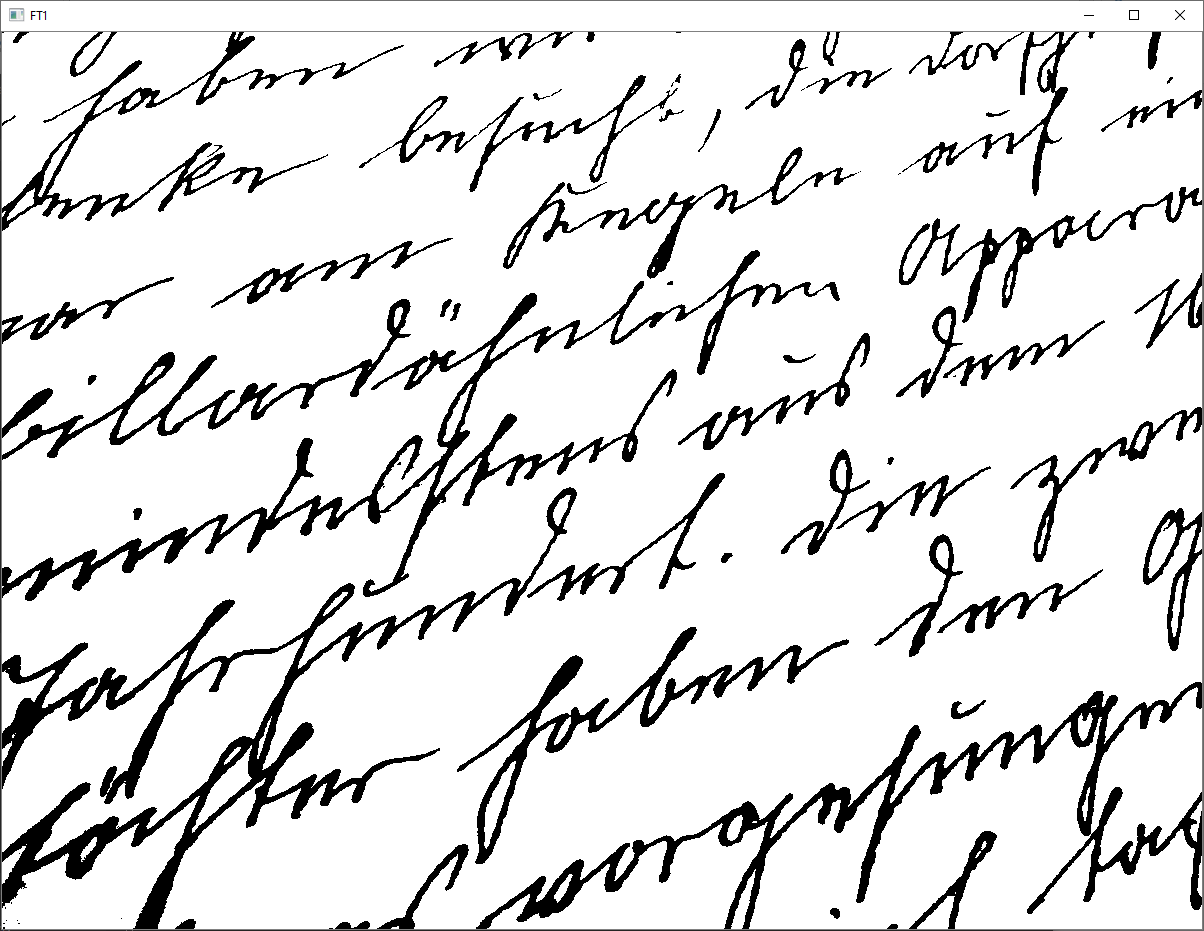


* + Бинаризация TRESH\_BINARY

TRESH\_BINARY

Исходное изображение

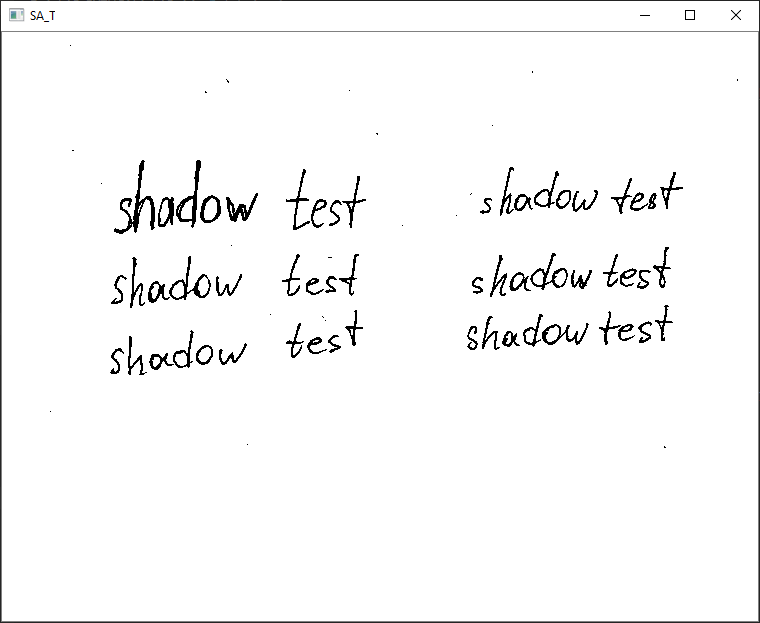
1. Фото написанного от руки или напечатанного текста.
   * Бинаризация с автоматическим вычислением порога (Оцу).

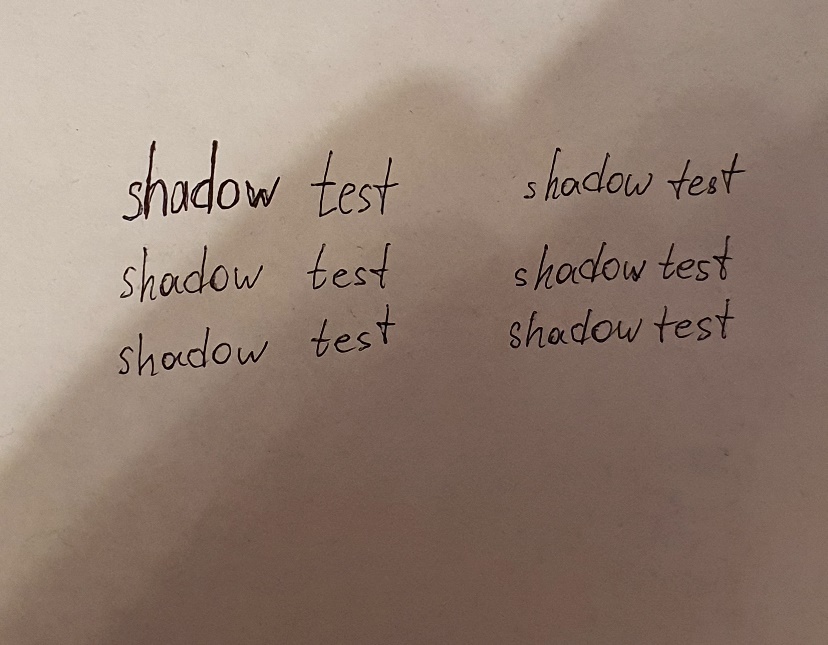
На изображении имеется светлая и темная области, определение порога вручную возможно, но затруднительно.

TRESH\_OTSU

Исходное изображение



* + Бинаризация с адаптивным вычислением порога. На изображении имеется тень и полутень.



adaptiveThreshold

ADAPTIVE\_THRESH\_MEAN\_C

Исходное изображение

**Вывод:** в ходе выполнения данной лабораторной работы было изучено применение порогового фильтра к различным типам изображений.