



НИУ ИТМО

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6**

«Морфологический анализ изображений»

Выполнили:

**Александр Иванов, Ф ТЕХ.ЗРЕНИЕ 1.1**

**Ани Аракелян, ТЕХ.ЗРЕНИЕ 1.1**

**Никита Братушка, ТЕХ.ЗРЕНИЕ 1.3**

Преподаватель:

**Шаветов С. В.**

Санкт-Петербург, 2024

# Содержание

<b>1. Базовые морфологические операции</b>	<b>3</b>
1.1. Исходное изображение . . . . .	3
1.2. Результаты . . . . .	5
<b>2. Разделение объектов</b>	<b>6</b>
2.1. Исходное изображение . . . . .	6
2.2. Программа на языке MATLAB . . . . .	6
2.3. Результаты . . . . .	7
<b>3. Сегментация</b>	<b>8</b>
<b>4. Выводы</b>	<b>9</b>
<b>5. Ответы на вопросы</b>	<b>9</b>

# 1. Базовые морфологические операции

## 1.1. Исходное изображение

Для первого задания выберем фотографию Яна Берри, сделанную в разгар событий Пражской весны:

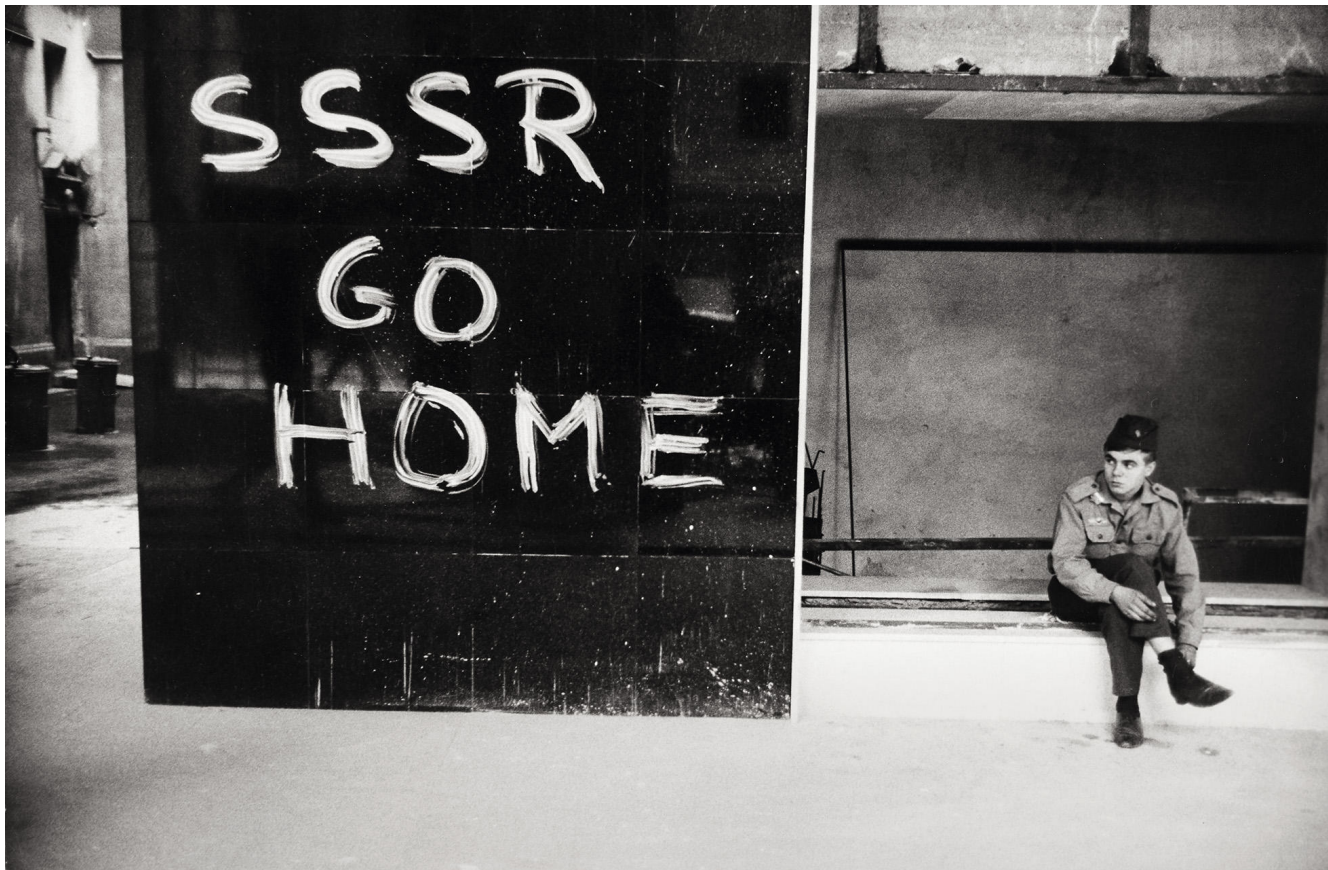


Рис. 1: Ян Берри. Молодой советский солдат отдыхает рядом с плакатом «СССР, возвращайся домой» на Вацлавской площади. Прага, Чехословакия. 1968

Нашей задачей будет минимизация дефектов формы на надписи *SSSR GO HOME*. В этом нам поможет *Python*. Для начала загрузим изображение и выберем область изображения с надписью для выполнения преобразований.

```
# loading grayscale image
src_img = cv.imread('source_images/Ian_Berry_A young_Russian
soldier.Jpeg', 0)
assert src_img is not None, "File could not be read"
result = np.copy(src_img)
# selecting specific area
spec_area = src_img[0:735, 200:1080]
```

Листинг 1: Исходный код для считывания изображения



Рис. 2: Надпись, которая подвергнется преобразованиям

С помощью эрозии избавимся от лишних точек на стене и выступов на буквах. Далее применим дилатацию от избавления от «внутренних дырок» и затем эрозию для избавления от частиц, появившихся в результате дилатации, и возвращения буквам их исходной толщины.

```
# morphological operations
disk = cv.getStructuringElement(cv.MORPH_ELLIPSE, (3, 3))
erosion = cv.erode(spec_area, disk, iterations=1)
dilation = cv.dilate(erosion, disk, iterations=7)
erosion2 = cv.erode(dilation, disk, iterations=5)
```

Листинг 2: Исходный код для преобразования надписи

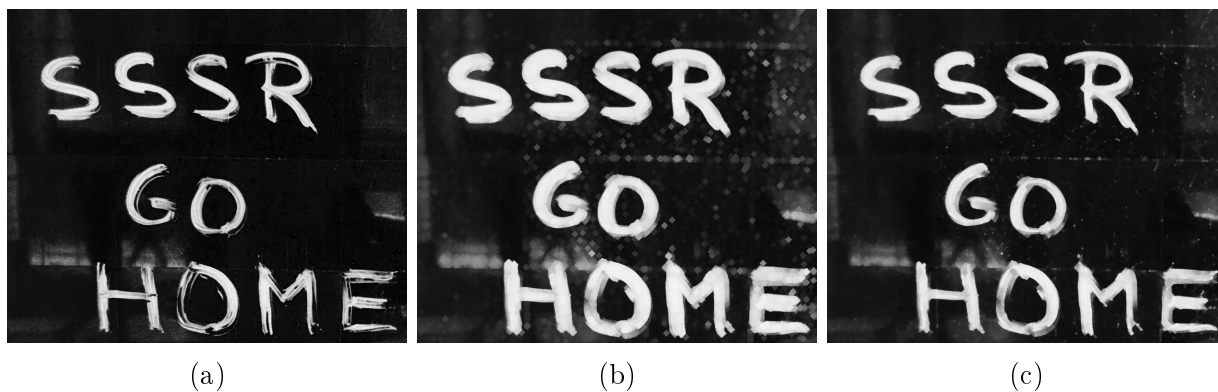


Рис. 3: Избавление от дефектов: (a) результат применения эрозии, (b) результат применения дилатации 7 раз, (c) результат применения эрозии 5 раз

## 1.2. Результаты

После преобразований необходимо вернуть «новую» надпись в исходное изображение.

```
# displaying transformed image  
result[0:735, 200:1080] = erosion2  
display_image('Result', result, 1)
```

Листинг 3: Исходный код для получения итогового изображения



Рис. 4: Результат применения базовых морфологических операций



## 2. Разделение объектов

### 2.1. Исходное изображение

Обратимся к кругам и их производным. Исходное изображение:

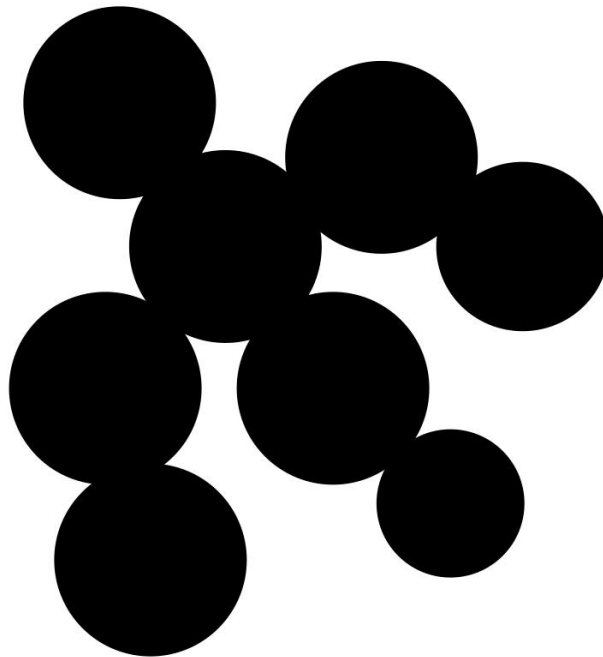


Рис. 5: Круги

Для выполнения этого задания воспользуемся *MATLAB*.

### 2.2. Программа на языке MATLAB

```
src_img = imread("circles.jpg");  
gray_img = im2gray(src_img);  
BW = imbinarize(gray_img);  
BW = ~BW;  
imwrite(BW,"binary_inv.jpg");  
BW2 = bwmorph(BW,'erode',45);  
imwrite(BW2,"eroded.jpg");  
BW2 = bwmorph(BW2,'thicken',Inf);  
imwrite(BW2,"boundaries.jpg");  
BW = ~(BW & BW2);  
imwrite(BW,"result.jpg");
```

Листинг 4: Исходный код программы для разделения объектов

## 2.3. Результаты

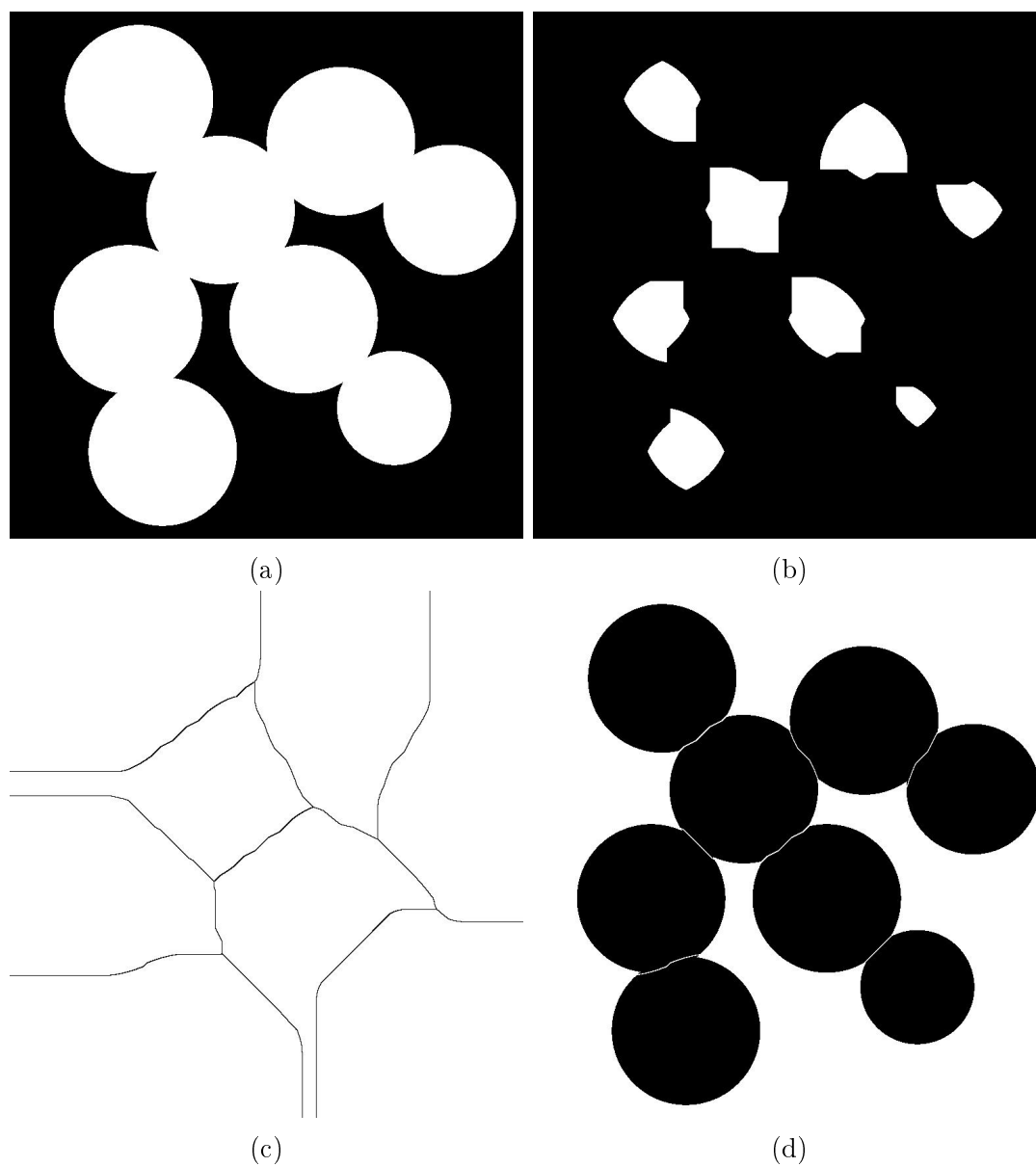


Рис. 6: Разделение «склеенных объектов»: (a) инвертированное бинарное изображение, (b) эрозия бинарного изображения, (c) расширение объектов, (d) результат разделения

### 3. Сегментация



## 4. Выводы

В результате выполнения работы мы познакомились с преобразованием Хафа для поиска геометрических примитивов.

Стоит отметить, что важную роль в успешном определении прямых линий и окружностей играет оператор Кэнни, который обнаруживает контуры изображения. Нам удалось убедиться в этом при использовании полутонного изображения для преобразования Хафа: результаты при обнаружении прямых были удручающими (см. рисунки ??, ?? и ??), а при определении окружностей программа не смогла ничего вывести.

Полный код программы и изображения можно найти на [GitHub](#).

## 5. Ответы на вопросы

Q1. Какая идея лежит в основе преобразования Хафа?

A1. В основе данного преобразования лежит идея поиска общего геометрического места точек (ГМТ) с помощью метода «голосования» точек. В классическом преобразовании также используется идея пространства параметров  $(\theta, \rho)$  уравнения прямой  $y = kx + b \Leftrightarrow x \cos(\theta) + y \sin(\theta) = \rho$

Q2. Можно ли использовать преобразование Хафа для поиска произвольных контуров, которые невозможно описать аналитически?

A2. Решение данной задачи возможно при использовании **обобщенного преобразования Хафа**.

Q3. Что такое рекуррентное и обобщенное преобразования Хафа?

A3. Особенность рекуррентного преобразования Хафа заключается в том, что мы применяем преобразование в скользящем окне, определяя аккумулятор преобразования для каждой области, после чего будем составлять общий аккумуляторный массив. В результате каждая точка общего аккумулятора будет характеризоваться параметрами наиболее достоверного отрезка прямой, проходящего через него.

Обобщенное преобразование Хафа применяется для случая произвольных контуров, не описываемых аналитически. В данном случае функция расстояния от пикселя границы

до центра является функцией  $R(\phi)$  от угла  $\phi$  радиус-вектора, направленного от точки контура к центру (точка локализации). При этом  $\phi$  может являться не только углом абсолютного направления на центр, но и относительным углом между направлением градиента и направлением радиуса-вектора.

Q4. Какие бывают способы параметризации в преобразовании Хафа?

A4. Бывают следующие способы параметризации: точки периметра  $(n, t)$  сетки изображения, точка периметра и угол  $(\alpha, n)$ , Наклон и смещение  $(\alpha, d)$ , основание нормали.