

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

**ОТЧЕТ**  
**по лабораторной работе №6**  
**по дисциплине «Техническое зрение»**  
**ТЕМА: Преобразование Хафа.**

Студент гр. 6492

\_\_\_\_\_

Мурашко А.С.

Огурецкий Д. В.

Преподаватель

\_\_\_\_\_

Моклева К.А.

Санкт-Петербург  
2020

## Лабораторная работа №6

**Цель:** изучить принцип применения преобразования Хафа для поиска прямых и окружностей.

**Исходные изображения:**

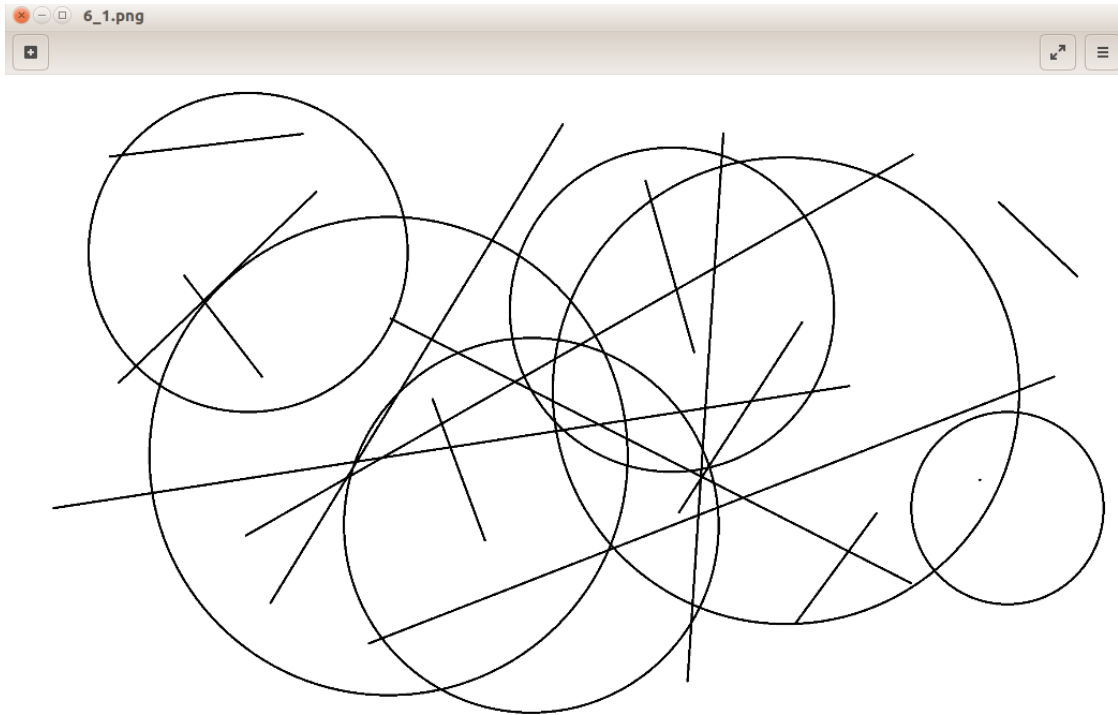
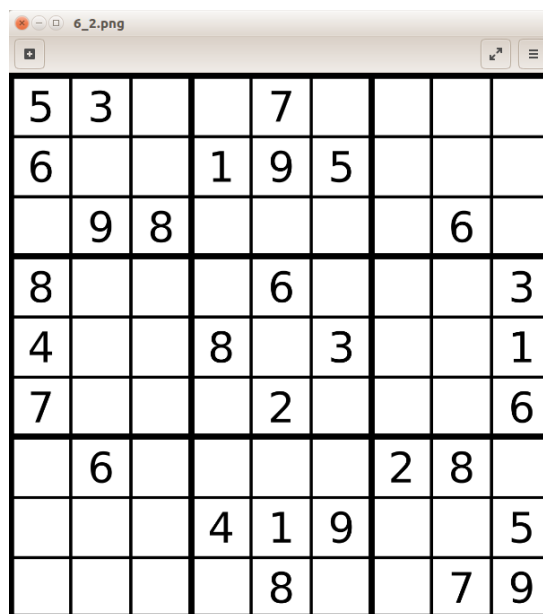


Рисунок 1 —

Исходное  
изображение 1.



5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Рисунок 2 — Исходное изображение 2.

### Ход работы.

Выделим самый длинный отрезок и самую длинную окружность:

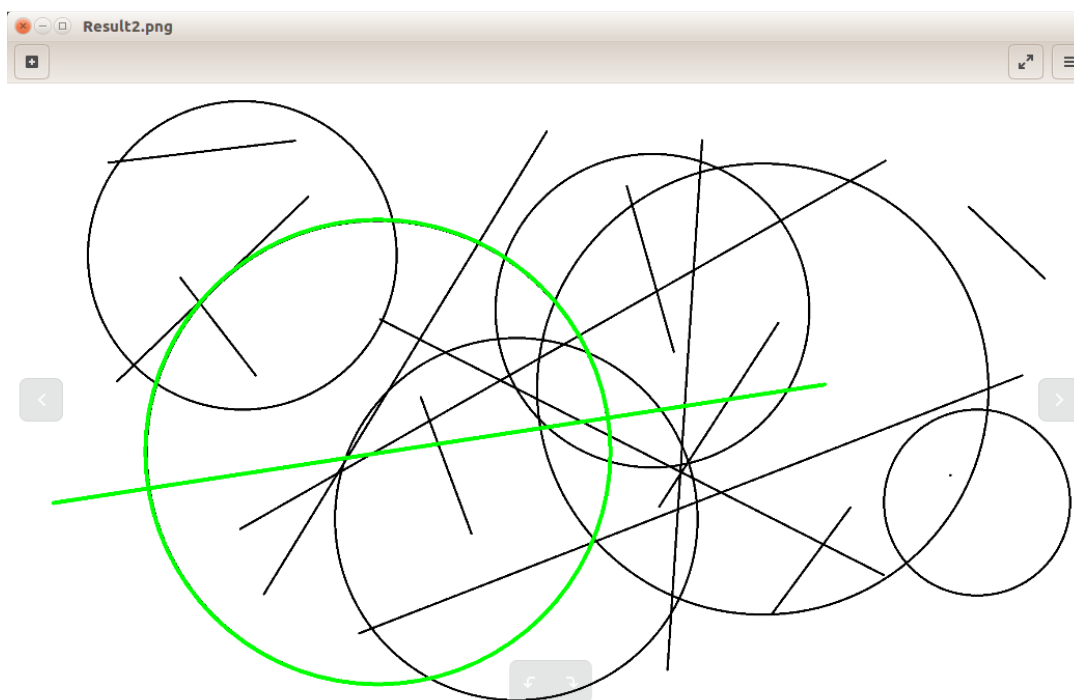


Рисунок 3 — Самая большая окружность и самый длинный отрезок.

Из рисунка 2 уберем  
все линии и оставим  
только цифры:

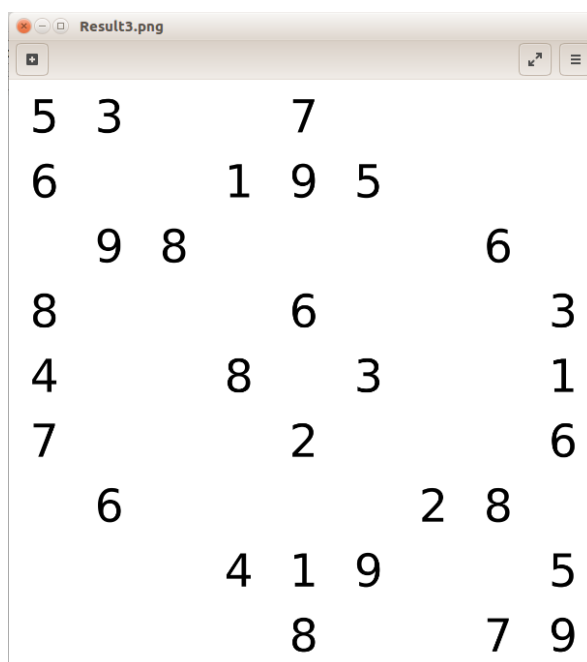


Рисунок 4 — Изображение без линий.

## Код программы.

```
import cv2
import numpy as np

# ----- Задание 1 -----
# читаем исходное изображение
img = cv2.imread('/home/andrew/foto/6_1.png')
# переводим его в серый формат
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# инвертируем изображение
img2 = cv2.bitwise_not(img2)
# размываем изображение для детектора Кенни
img3 = cv2.GaussianBlur(img, (5, 5), sigmaX = 0, sigmaY = 0, borderType = cv2.BORDER_DEFAULT)
Canny = cv2.Canny(img3, 100, 200, apertureSize = 5, L2gradient = True)
cv2.imwrite('/home/andrew/foto/Canny.png', Canny)
# Находим линии на изображении
linesP = cv2.HoughLinesP(img2, 1, np.pi/720, 200, minLineLength = 100, maxLineGap = 10)
# Находим окружности на изображении
circles = cv2.HoughCircles(Canny, cv2.HOUGH_GRADIENT, dp = 1, minDist = 200, param1 = 200,
param2 = 100)
# переменная для нахождения максимальной длины отрезка
lenghtMax = 0
for line in linesP:
    x1, y1, x2, y2 = line[0]
    lenght = np.sqrt((x2-x1)**2 + (y2-y1)**2)
    if lenght > lenghtMax:
        lenghtMax = lenght
        X1, Y1, X2, Y2 = x1, y1, x2, y2

print(X1, Y1, X2, Y2)
# рисуем самый длинный отрезок
cv2.line(img, (X1, Y1), (X2, Y2), (0, 255, 0), 7)
cv2.imwrite('/home/andrew/foto/Result.png', img)

# переменная для нахождения наибольшего радиуса окружности
R = 0
for c in circles[0]:
```

```

x, y, r = c
if r > R:
    X, Y, R = x, y, r

print(R)
# рисуем наибольшую окружность
cv2.circle(img, (X, Y), int(R), (0, 255, 0), 7)
cv2.imwrite('/home/andrew/foto/Result2.png', img)

# ----- Задание 2 -----
# читаем исходное изображение
img = cv2.imread('/home/andrew/foto/6_2.png')
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img2 = cv2.bitwise_not(img2)
# находим линии на изображении
linesP = cv2.HoughLinesP(img2, 1, np.pi/720, 200, minLineLength = 100, maxLineGap = 10)

for line in linesP:
    x1, y1, x2, y2 = line[0]
    # замазываем все линии белым, оставляя только цифры на изображении
    cv2.line(img, (x1, y1), (x2, y2), (255, 255, 255), 7)

cv2.imwrite('/home/andrew/foto/Result3.png', img)

```

**Вывод:** в ходе работы была изучена работа с преобразованием Хафа, а именно: поиск линий и окружностей на изображении, в частности самой большой окружности и самого длинного отрезка. А также рисовка и «удаление» линий с изображения.