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. 6491 _____ Бузи Дарья

Преподаватель _____ Моклева К.А.

Санкт-Петербург

2020

Лабораторная работа 6

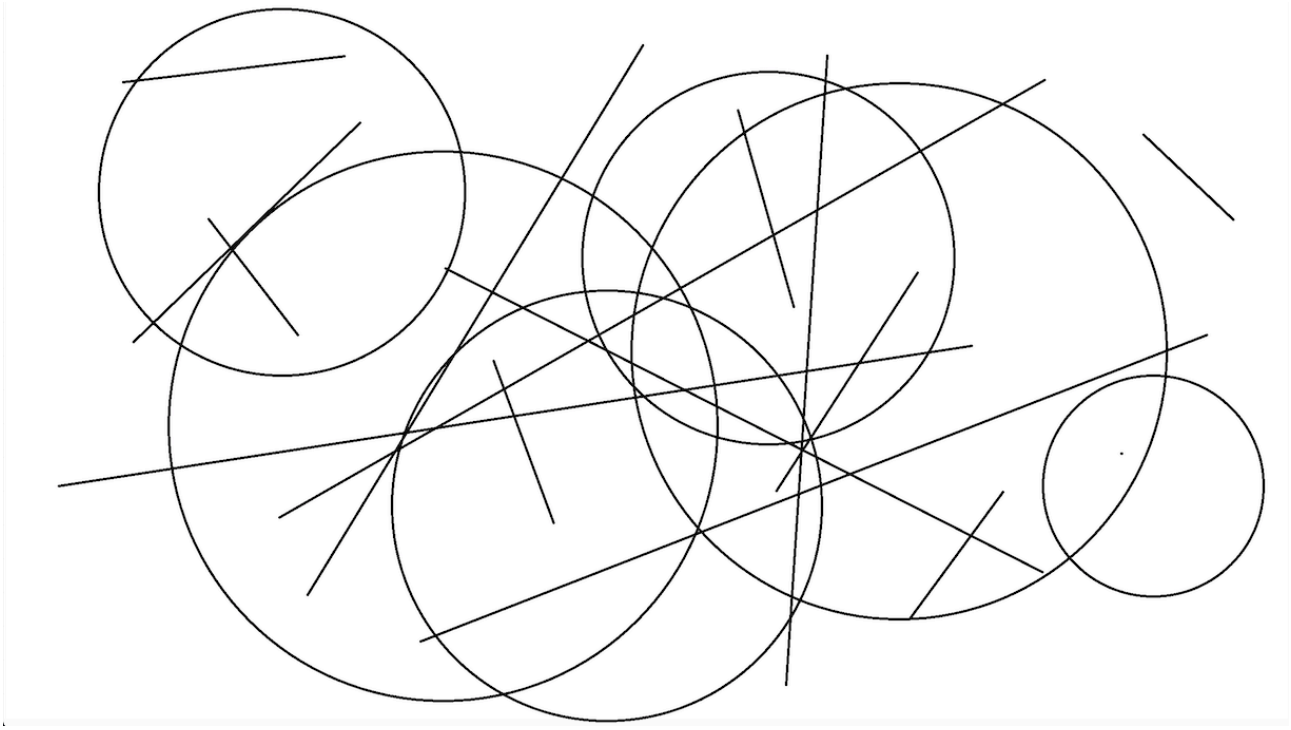
Преобразование Хафа

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

Ход работы:

Задание 1. Для выполнения этого задания нужно использовать файл 6_1.png. На этом изображении выделите цветом самую большую окружность и самый длинный отрезок.

Исходный рисунок 6_1.png:



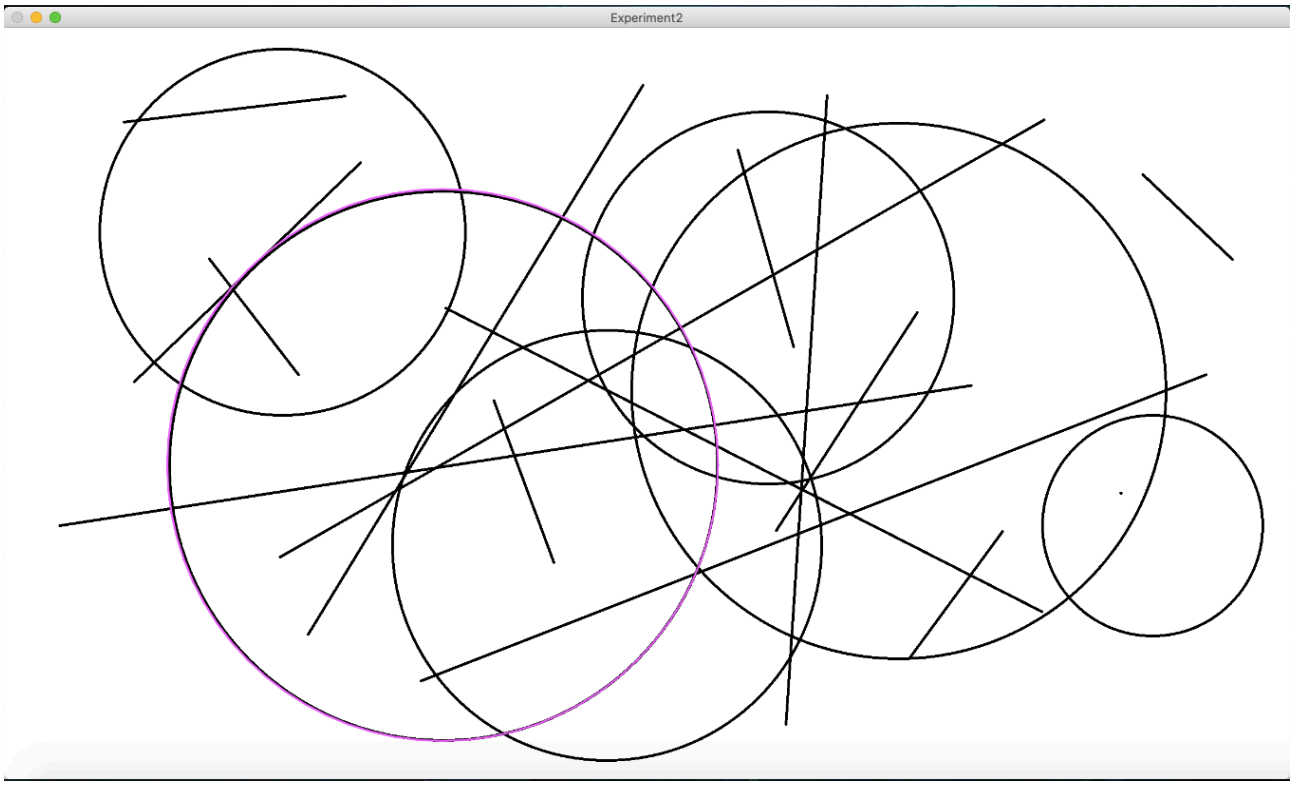
Код поиска наибольшей окружности:

```
1.  import cv2
2.  import numpy
3.
4.  image_colour = cv2.imread('/Users/dariabusi/Desktop/
    6_1.png',cv2.IMREAD_COLOR)
5.  image = cv2.imread('/Users/dariabusi/Desktop/
    6_1.png',cv2.IMREAD_GRAYSCALE)
6.
7.  #найдем границы с помощью детектора границ Кэнни
8.  edges = cv2.Canny(image, 75, 255)
9.  """функция поиска окружностей преобразованием Хафа, где:
10. circles – список векторов хранящие координаты центра и радиусы окружностей
11. edges – в какой картинке происходит поиск окружностей
12. cv2.HOUGH_GRADIENT – единственный метод поиска окружностей, с помощью
    градиентов
13. dp – – во сколько раз меньше разрешение у аккумулятора,
```

```
чем у входного изображения
minDist – минимальное расстояние между окружностями
param1 – верхняя граница для детектора границ Кэнни
param2 – порог аккумулятора"""
circles = cv2.HoughCircles(edges, cv2.HOUGH_GRADIENT, dp = 1, minDist =
280, param1 = 255, param2 = 97, minRadius = 0, maxRadius = 0)
print("Circles: ", circles)

x0_m, y0_m, r_m = [0,0,0]
for circle in circles[0]:
            (x0, y0, r) = circle
            if r>=r_m:
                x0_m, y0_m, r_m = (x0, y0, r)

print("maximum Radius: ", r_m)
cv2.circle(image_colour, (x0_m, y0_m), r_m, (255,0,255), 2, cv2.LINE_AA)
cv2.imshow ('Experiment2', image_colour)

cv2.waitKey(0)
```

Результат:

```
Circles:  [[[ 812.5  807.5  510.1]
  [1656.5  666.5  495.4]
  [1115.5  956.5  395.1]
  [ 516.5  375.5  335.9]
  [1413.5  501.5  343.3]
  [2134.5  915.5  199.1]]]
maximum Radius:  510.1
```

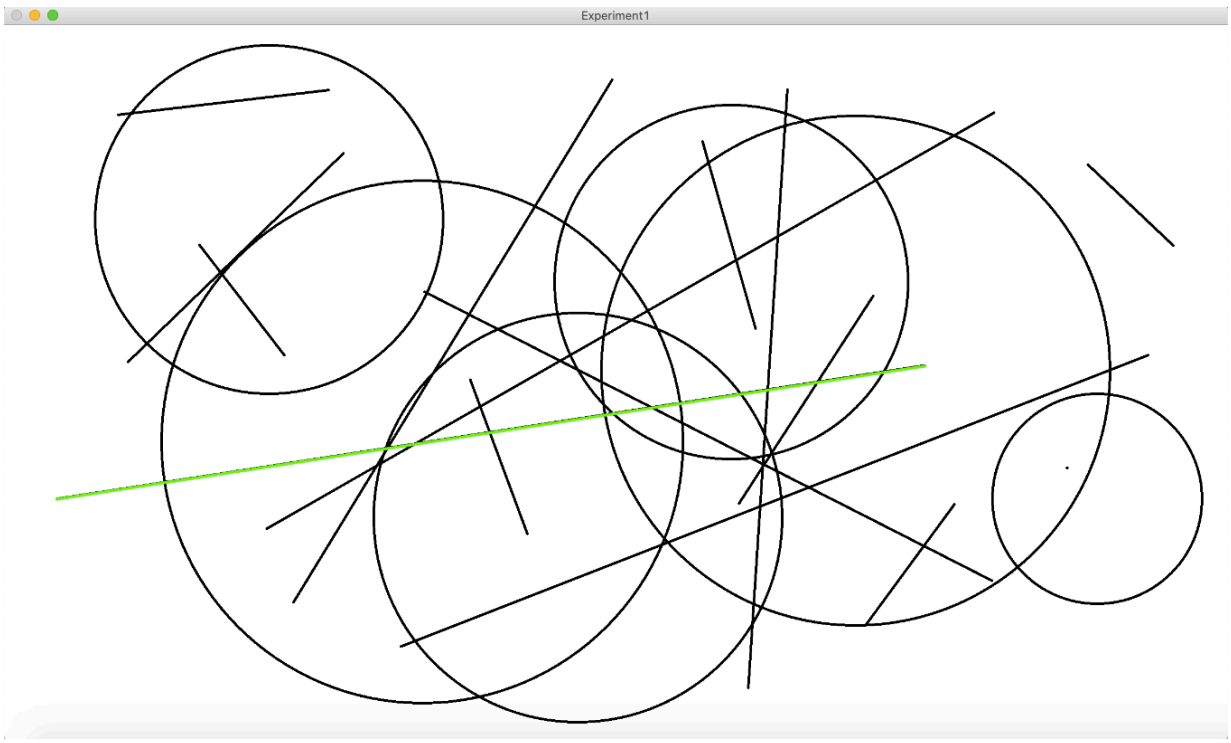


Код поиска наибольшего отрезка:

```
1.  import cv2
2.  import numpy
3.  from math import sqrt
4.
5.  image_colour = cv2.imread('/Users/dariabusi/Desktop/
    6_1.png',cv2.IMREAD_COLOR)
6.  image = cv2.imread('/Users/dariabusi/Desktop/
    6_1.png',cv2.IMREAD_GRAYSCALE)
7.  invert_image = 255 - image
8.  cv2.imshow ('Experiment', invert_image)
9.  """функция поиска отрезков преобразованием Хафа (Прогрессивное
    вероятностное
10. преобразование Хафа), где:
11. по полярным координатам rho - расстояние от начала координат до прямой
12.                         theta - угол между нормалью прямой и осью
13. """
14. linesP = cv2.HoughLinesP(invert_image, rho = 1, theta = numpy.pi/580,
    threshold = 255, minimalLength = 10, maxLineGap = 125)
15.
16. maxLength=0
17. x1,y1,x2,y2 = [0,0,0,0]
18. for i in range(0, len(linesP)):
19.             l = linesP[i][0]
20.             currentLength = sqrt((l[2]-l[0])**2 + (l[3]-l[1])**2)
21.             if currentLength>=maxLength:
22.                 maxLength=currentLength
23.                 x1,y1,x2,y2 = l
24.
25. print("maximum Lenght: ", maxLength)
26. cv2.line(image_colour, (x1, y1), (x2, y2), (0,255,0), 3, cv2.LINE_AA)
27.
28. cv2.imshow ('Experiment1', image_colour)
29. cv2.waitKey(0)
```

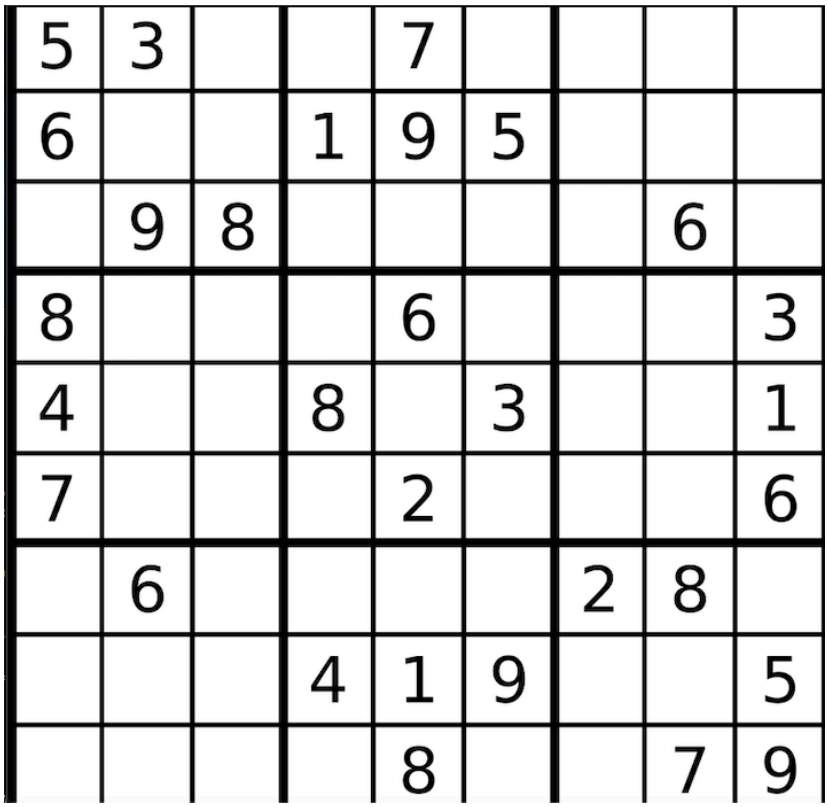
Результат:

```
maximum Lenght:  1711.7082111154343
```



Задание 2. Для выполнения этого задания нужно использовать файл 6_2.png. Исправьте это изображение так, чтобы линии таблицы исчезли, а числа остались. Для решения этой задачи воспользуйтесь преобразованием Хафа для поиска прямых.

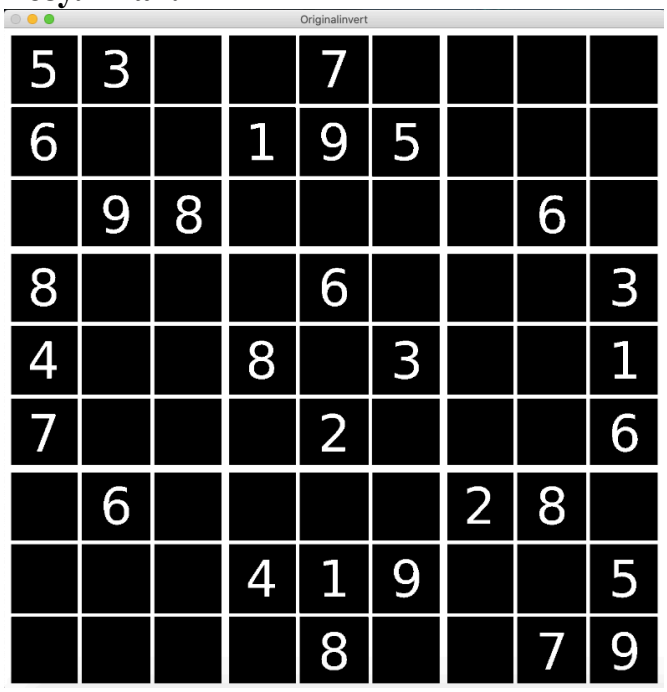
Исходный рисунок 6_2.png:

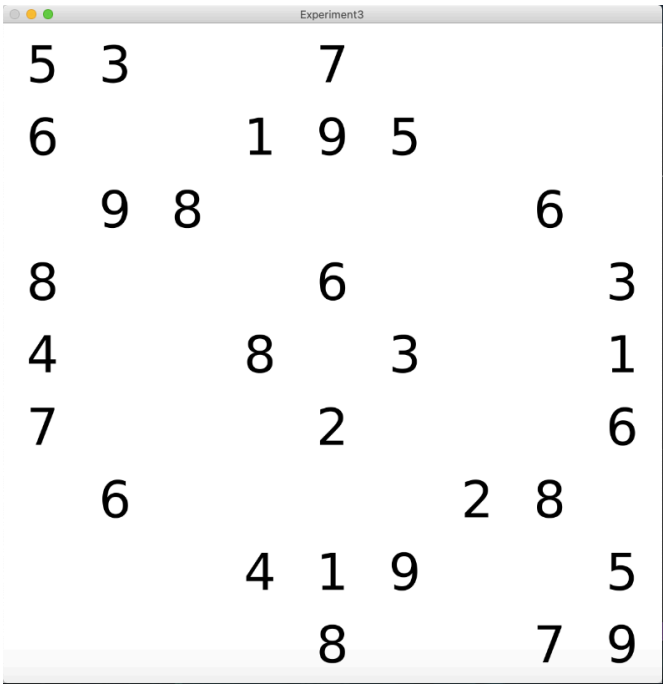


Код задания:

```
import cv2
import numpy

image_colour = cv2.imread('/Users/dariabusi/Desktop/6_2.png',cv2.IMREAD_COLOR)
cv2.imshow ('Original', image_colour)
image_grey = cv2.imread('/Users/dariabusi/Desktop/6_2.png',cv2.IMREAD_GRAYSCALE)

new_threshold, image_thresh = cv2.threshold(image_grey, 125, 255 , cv2.THRESH_BINARY_INV)
cv2.imshow ('Originalcanny', image_thresh)

linesP = cv2.HoughLinesP( image = image_thresh, rho = 1, theta = numpy.pi/100 , threshold = 255, maxLineGap = 15)

for i in range(0, len(linesP)):
            l = linesP[i][0]
            cv2.line(image_colour, (l[0], l[1]), (l[2], l[3]), (255,255,255), 5, cv2.LINE_AA)

cv2.imshow ('Experiment3', image_colour)



cv2.waitKey(0)
```

Результат:





Вывод: в ходе лабораторной работе было изучено прогрессивное вероятностное преобразование Хафа для нахождения линий и окружностей на различных изображениях. Со стандартным преобразованием Хафа возникают трудности.