**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Систем автоматического управления**

отчет

**по лабораторной работе №6**

**по дисциплине «Техническое зрение»**

**Тема: Преобразование Хафа**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 6491 |  | Ларионов Е.А.  Зверев Д.Ю. |
| Преподаватель |  | Моклева К. А. |

Санкт-Петербург

2020 г.

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

**Задание 1.**

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

**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.

**Код задания1:**

from cv2 import cv2

import numpy as np

from math import sqrt

img = cv2.imread('C:/Users/greka/greka/technical vision/6\_1.png', cv2.IMREAD\_COLOR)

img\_gray = cv2.imread('C:/Users/greka/greka/technical vision/6\_1.png', cv2.IMREAD\_GRAYSCALE)

lines\_img = np.copy(img)

circles\_img = np.copy(img)

#инвертируем изображение в оттенках серого для поиска прямых

inv = img\_gray - 255

#функция изменения выходного изображения

def resized (img\_res):

    final\_wide=700

    r1=float(final\_wide)/img\_res.shape[1]

    dim1 = (final\_wide, int(img\_res.shape[0] \* r1))

    return dim1

lines=cv2.HoughLinesP(inv, rho=1, theta=np.pi/720, threshold = 250, maxLineGap=15) #поиск прямых прогрессивным вероятностным преобразованием Хафа

maxLen = 0

x\_1, y\_1, x\_2, y\_2 = 0, 0, 0, 0

#выделим все линии

for line in lines:

    x1, y1, x2, y2 = line[0]

    currentLen = sqrt((x2 - x1)\*\*2 + (y2 - y1)\*\*2)

    print('Current Lenght: ', currentLen, 'line:',line,'\n')

    if currentLen >= maxLen: #определение самой длинной линии

        maxLen = currentLen

        x\_1, y\_1, x\_2, y\_2 = x1, y1, x2, y2

    cv2.line(lines\_img, (x1, y1), (x2, y2), color=(255,0,0), thickness=1, lineType=cv2.LINE\_AA)   #выделение всех линий

cv2.line(lines\_img, (x\_1, y\_1), (x\_2, y\_2), color=(0,0,255), thickness=2, lineType=cv2.LINE\_AA) #выделение самой длинной линии

canny = cv2.Canny(img\_gray, 230, 250,3)

circles = cv2.HoughCircles(canny, cv2.HOUGH\_GRADIENT, dp = 1, minDist = 300, param1 = 255,

param2 = 100 , minRadius = 0, maxRadius = 0)

print("Circles: ", circles)

x\_max, y\_max, r\_max = 0, 0, 0

# Выделим все найденные окружности

for circle in circles[0]:

    x0, y0, r = circle

    if r > r\_max:

        x\_max, y\_max, r\_max = x0, y0, r

    cv2.circle(circles\_img, (x0, y0), r, color=(0,255,0), thickness=2, lineType=cv2.LINE\_AA)

# Выделим самую большую окружность

cv2.circle(circles\_img, (x\_max, y\_max), r\_max, color=(0,0,255), thickness=2,

lineType=cv2.LINE\_AA)

# cv2.imshow('canny', cv2.resize(canny, resized(canny), interpolation = cv2.INTER\_AREA))

# cv2.imshow('gray', cv2.resize(img\_gray, resized(img\_gray), interpolation = cv2.INTER\_AREA))

# cv2.imshow('inv\_gray', cv2.resize(inv, resized(inv), interpolation = cv2.INTER\_AREA))

cv2.imshow('lines', cv2.resize(lines\_img, resized(lines\_img), interpolation = cv2.INTER\_AREA))

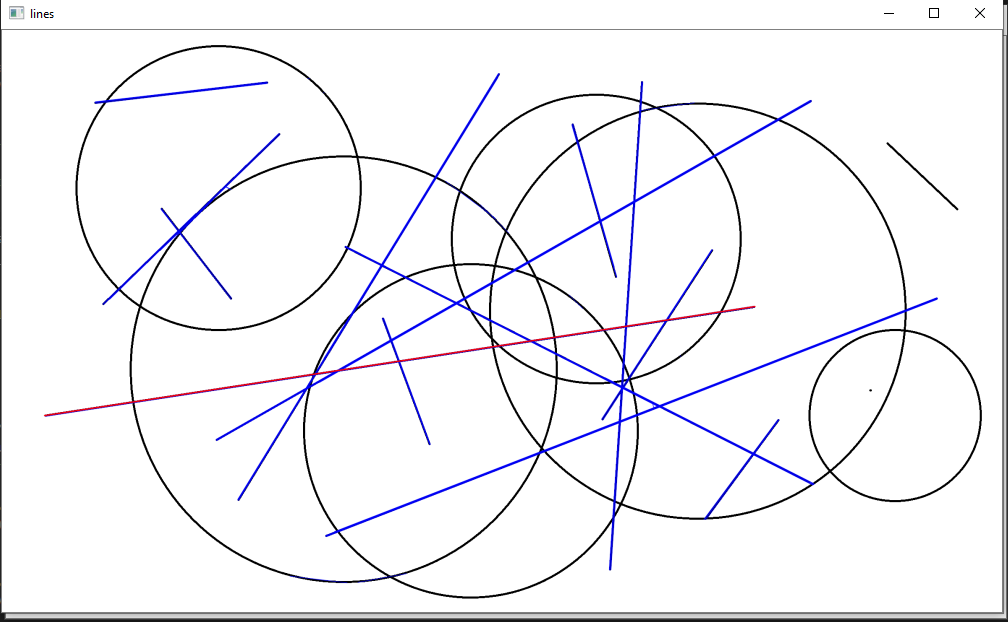
cv2.imshow('Circles', cv2.resize(circles\_img, resized(circles\_img),

interpolation = cv2.INTER\_AREA))

cv2.waitKey(-1)

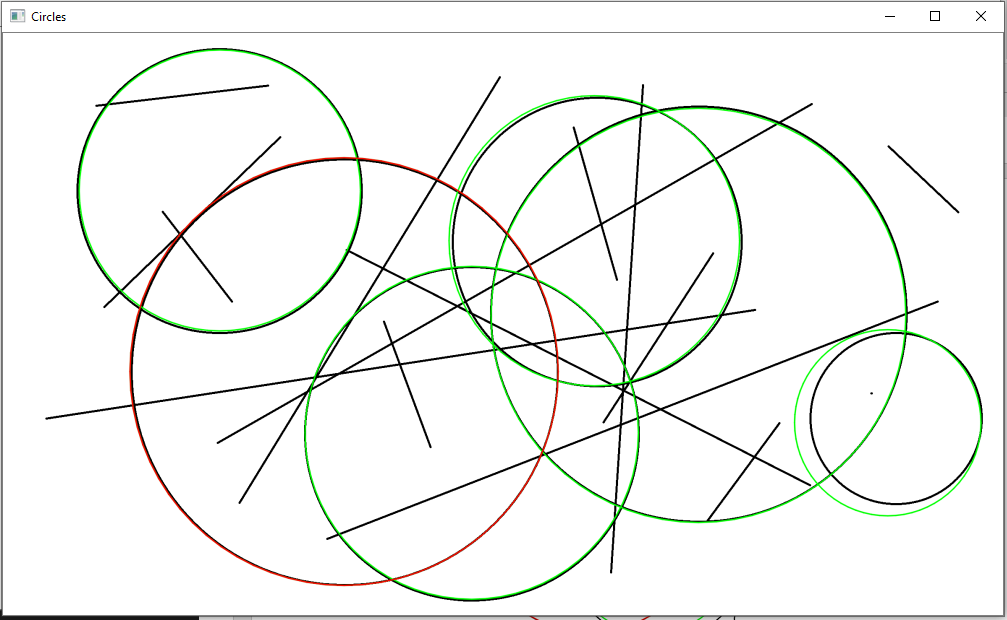
cv2.destroyAllWindows()

**Результат выполнения основного задания 1:**



Изображение с выделенными линиями

Все линии, кроме одной (в правом верхнем углу), были выделены алгоритмом. Самая длинная выделена красным цветом.



Изображение с выделенными окружностями

Все окружности были выделены алгоритмом. Самая большая выделена красным цветом.

**Код задания 2:**

from cv2 import cv2

import numpy as np

from math import sqrt

img = cv2.imread('C:/Users/greka/greka/technical vision/6\_2.png', cv2.IMREAD\_REDUCED\_COLOR\_2)

img\_gray = cv2.imread('C:/Users/greka/greka/technical vision/6\_2.png', cv2.IMREAD\_REDUCED\_GRAYSCALE\_2)

inv=img\_gray-255

img\_without=np.copy(img)

lines = cv2.HoughLinesP(inv, rho = 1, theta = np.pi/180, threshold = 250,

maxLineGap=20)

for line in lines:

    x1, y1, x2, y2 = line[0]

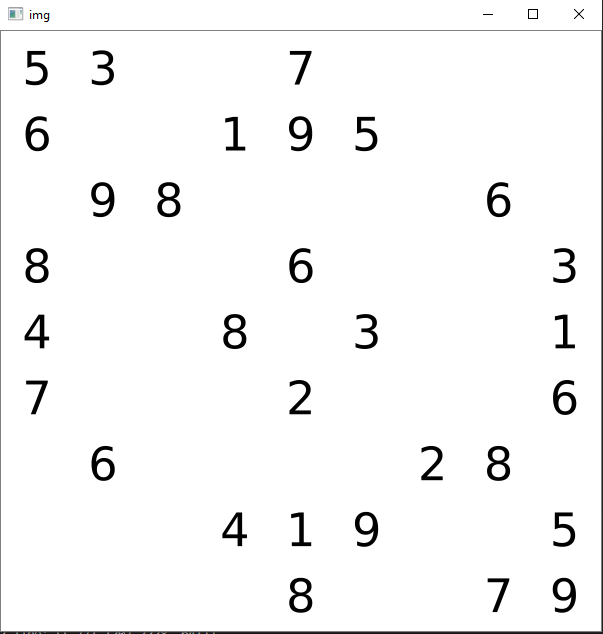
    cv2.line(img\_without, (x1, y1), (x2, y2), color=(255,255,255), thickness=2, lineType=cv2.LINE\_AA)

cv2.imshow('img',img\_without)

cv2.waitKey(-1)

cv2.destroyAllWindows()

**Результат выполнения основного задания 2:**



Изображение без черных линий с использованием вероятностного преобразования Хафа

**Вывод**:В ходе лабораторной работы были изучены методы преобразования Хафа для поиска линий и окружностей, и была проделана работа по их поиску на различных изображениях.