

|  |
| --- |
| Группа M32121 |
| Студенты Сахабутдинов Рустам, Захаров Даниил |
| Преподаватель Шоев Владислав Иванович |

Моделирование №1

Изучение интерференционной картине колец Ньютона

Санкт-Петербург

2022

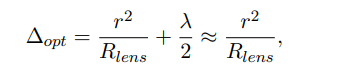
**Задача:**

Построение графика, отражающего интерференцию изображения колец Ньютона.

**Стэк для решения задачи:**

Python, matplotlib, numpy, scipy

**Формулы использованные в решении:**



Изображение выглядит как текст, часы

Автоматически созданное описание

Изображение выглядит как антенна, часы, датчик

Автоматически созданное описание

**Код решения:**

Edge\_detection.py:

import cv2  
import numpy as np  
from matplotlib import pyplot as plt  
from scipy.signal import argrelmin  
import sys  
  
  
img = cv2.imread("images/test\_2.jpg", 0)  
img = cv2.resize(img, (0, 0), fx=1, fy=1)  
img = cv2.medianBlur(img, 5)  
  
cimg = cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR)  
bwimg = cv2.cvtColor(img, cv2.COLOR\_GRAY2BGR)  
circles = cv2.HoughCircles(img, cv2.HOUGH\_GRADIENT, 5, 50, param1=110, param2=80, minRadius=70, maxRadius=0)  
circles = np.uint16(np.around(circles))  
center = np.array(np.shape(img)) / 2.0  
trueCenter = None  
  
for i in circles[0]:  
 if ((np.abs(i[1] - center[0]) \*\* 2 + np.abs(i[0] - center[1]) \*\* 2) \*\* 0.5 < 40):  
 trueCenter = i  
 cv2.circle(cimg, (i[0], i[1]), i[2], (0, 255, 0), 2)  
 cv2.circle(cimg, (i[0], i[1]), 2, (0, 0, 255), 3)  
  
size = int(np.shape(img)[0] / 2)  
dst = cv2.warpPolar(cimg, (size, size), (trueCenter[0], trueCenter[1]), maxRadius=size, flags=cv2.WARP\_POLAR\_LINEAR)  
dst2 = cv2.warpPolar(bwimg, (size, size), (trueCenter[0], trueCenter[1]), maxRadius=size, flags=cv2.WARP\_POLAR\_LINEAR)  
  
fig = plt.figure(figsize=(15, 15), dpi=80)  
plt.subplot(1, 2, 1)  
plt.imshow(cimg)  
plt.subplot(1, 2, 2)  
  
plt.plot(range(0, len(bwimg[trueCenter[1]]))[trueCenter[0]:], [i[1] for i in bwimg[trueCenter[1]]][trueCenter[0]:])  
  
plt.show()

**Ответ алгоритма на тестовых данных:**

