

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Нижегородский государственный университет им. Н.И. Лобачевского»

Национальный исследовательский университет

Институт информационных технологий, математики и механики

Кафедра: Программной инженерии

Направление подготовки: «Программная инженерия»

«Обработка изображений»

## Отчёт по лабораторной работе

Выполнил:

студент группы 381908-3

Имя

Бражник Дмитрий \_\_\_\_\_

(подпись)

Выполнил:

студент группы 381908-3

Имя

Черемушкин Кирилл \_\_\_\_\_

(подпись)

Выполнил:

студент группы 381908-1

Имя

Козел Светлана \_\_\_\_\_

(подпись)

Проверил:

Гетманская А.А.

\_\_\_\_\_  
(подпись)

Нижний Новгород

2021 г.

# Содержание

|                                     |   |
|-------------------------------------|---|
| Задача .....                        | 3 |
| Проделанная работа .....            | 3 |
| Анализ полученных результатов ..... | 3 |
| Библиотеки.....                     | 5 |
| Код программы .....                 | 6 |

## Задача

Применить обработку спектра Фурье для уничтожения полос на снимках с электронной микроскопии.

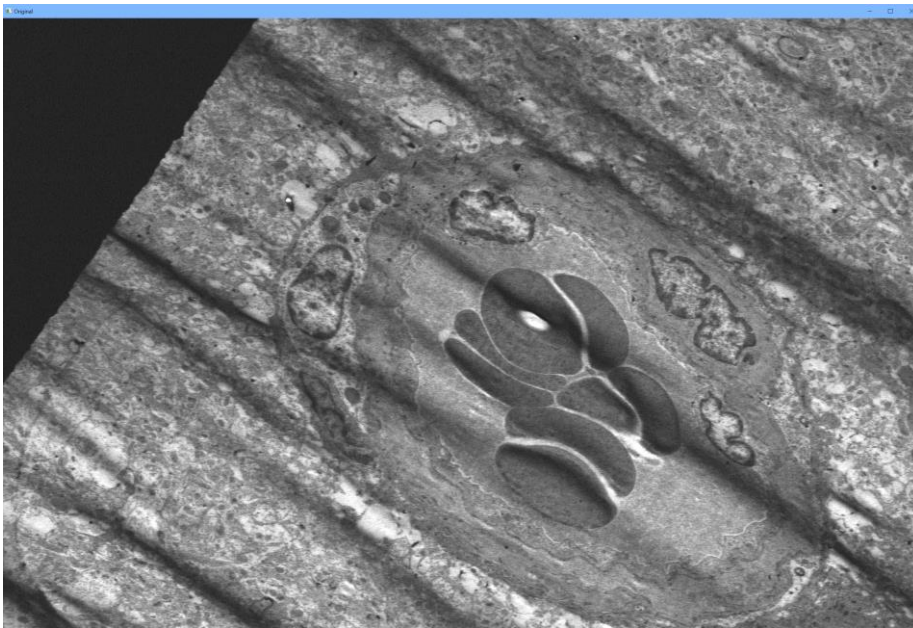
## Проделанная работа

Провели фильтрацию заранее заготовленных изображений, содержащих полосы. Целью было удалить полосы с изображения. С помощью Фурье преобразовали изображение и применили 3 фильтра.

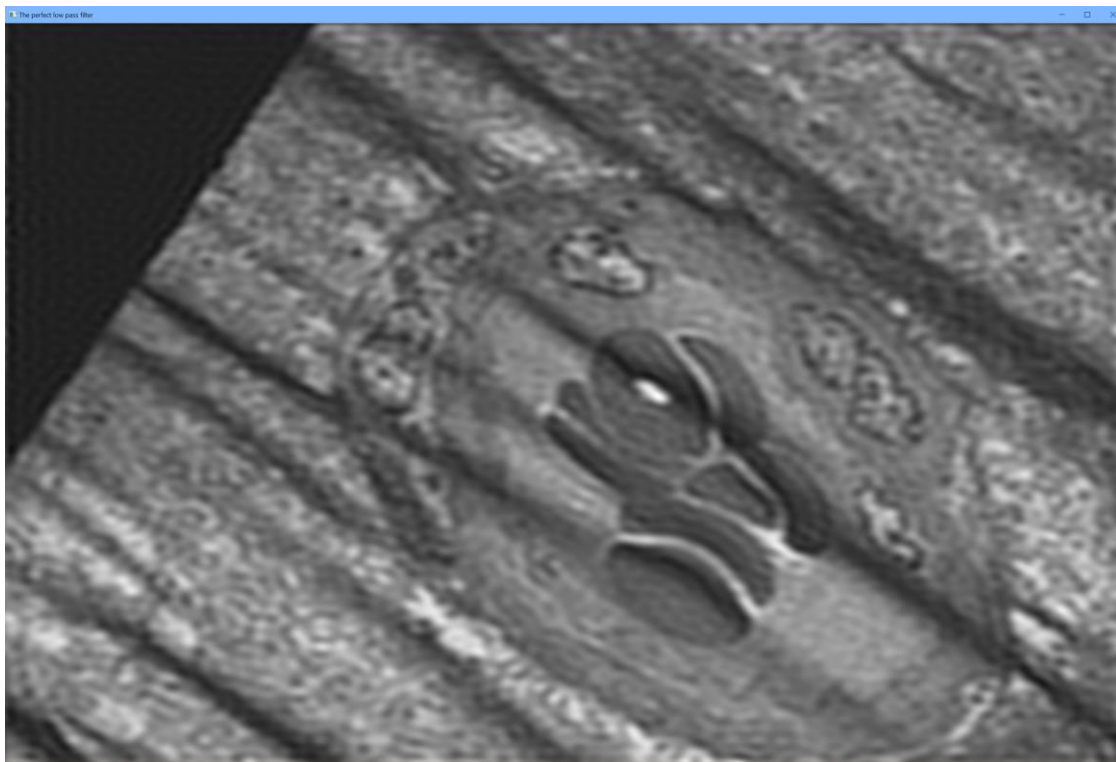
- 1) Идеальный фильтр нижних частот
- 2) Фильтр низких частот Баттерворта
- 3) Гауссова фильтрация нижних частот

## Анализ полученных результатов

Оригинальное изображение, на нём отчётливо видны черные полосы.



Применим первый фильтр идеальный фильтр нижних частот.



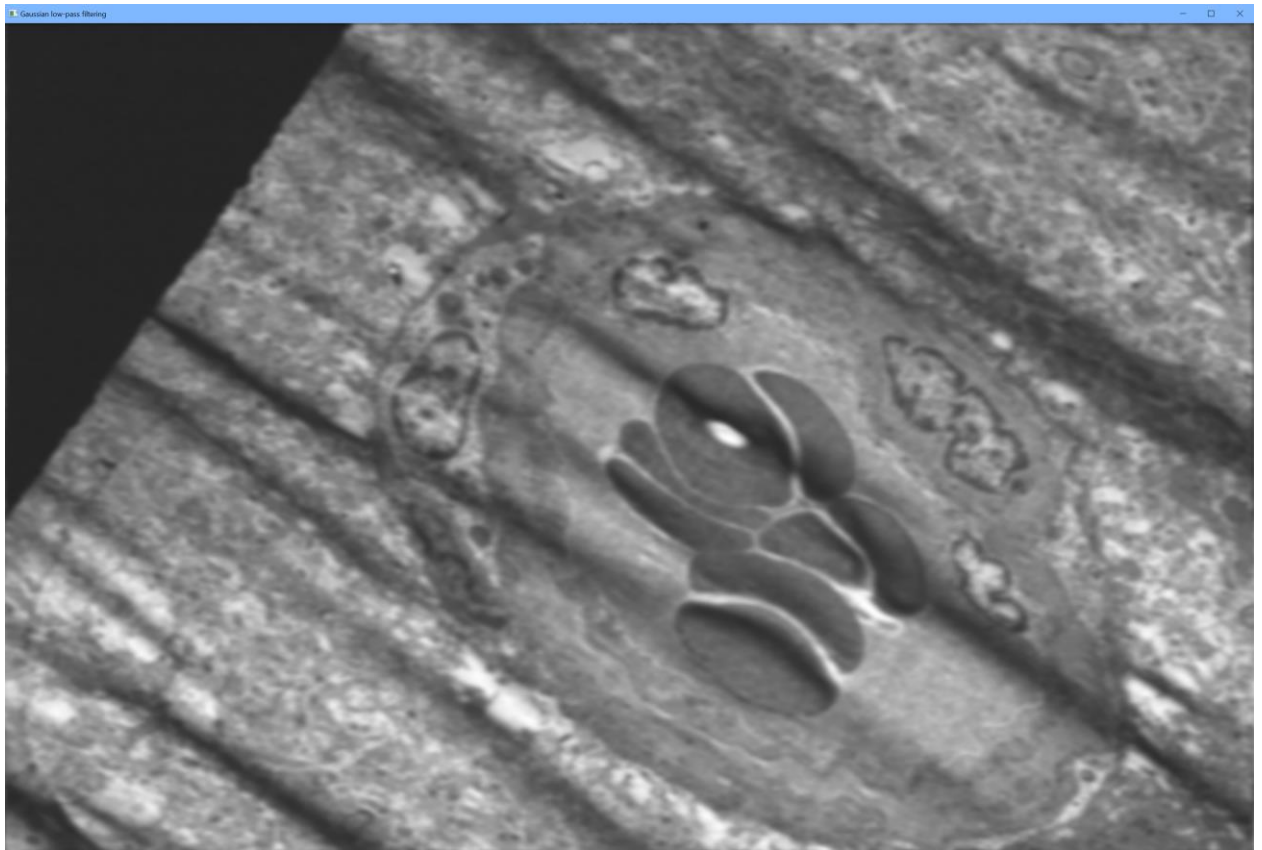
Чёрные полосы стали тусклее, но также всё изображение потеряло чёткость и стало слишком размыто. Результат неплох, но далёк от нужного нам идеала

Применим второй фильтр низких частот Баттерворта.



На данном изображении видны улучшения, черные полосы стали ещё менее заметны и тусклее, так же в верхнем левом углу чёрный треугольник стал чётче относительно предыдущего фильтра. Результат стал намного лучше.

Применим последний третий фильтр Гауссова фильтрация нижних частот.



Относительно предыдущих двух фильтров, данный фильтр показывает наилучший результат, чёрные полосы практически пропали, чёрный треугольник вверху слева не размыт, единственный побочный эффект, яркость пикселей сильно увеличилась. Данный фильтр можно признать наилучшим для решения подобного рода задач.

## Библиотеки

```
import cv2
import numpy as np
```

## Код программы

```
def fft_distances(m, n):
    u = np.array([i - m/2 for i in range(m)], dtype=np.float32)
    v = np.array([i - n/2 for i in range(n)], dtype=np.float32)
    ret = np.ones((m, n))
    for i in range(m):
        for j in range(n):
            ret[i][j] = np.sqrt(u[i]*u[i] + v[j]*v[j])
    u = np.array([i if i<=m/2 else m-i for i in range(m)], dtype=np.float32)
    v = np.array([i if i<=n/2 else n-i for i in range(n)], dtype=np.float32)
    return ret

def change_filter(flag):
    if flag == 1:
        filter_mat = np.zeros((nrows, ncols, 2), np.float32)
        cv2.circle(filter_mat, (np.int32(ncols/2), np.int32(nrows/2)), d0, (1,1,1), -1)
    elif flag == 2:
        n = 2
        filter_mat = None
        duv = fft_distances(*fft_mat.shape[:2])
        filter_mat = 1 / (1+ np.power(duv/d0, 2*n))
        filter_mat = cv2.merge((filter_mat, filter_mat))

    else:
        filter_mat = None
        duv = fft_distances(*fft_mat.shape[:2])
        filter_mat = np.exp(-(duv*duv) / (2*d0*d0))
        filter_mat = cv2.merge((filter_mat, filter_mat))
    return filter_mat

def ifft(fft_mat):
    f_ishift_mat = np.fft.ifftshift(fft_mat)
    img_back = cv2.idft(f_ishift_mat)
    img_back = cv2.magnitude(*cv2.split(img_back))
    cv2.normalize(img_back, img_back, 0, 255, cv2.NORM_MINMAX)
    return np.uint8(np.around(img_back))[rows:cols]

img = cv2.imread("E:\\__129.png", 0)
rows, cols = img.shape[:2]
d0 = 70
nrows = cv2.getOptimalDFTSize(rows)
ncols = cv2.getOptimalDFTSize(cols)
nimg = np.zeros((nrows, ncols))
nimg[:rows,:cols] = img
fft_mat = cv2.dft(np.float32(nimg), flags = cv2.DFT_COMPLEX_OUTPUT)
fft_mat = np.fft.fftshift(fft_mat)

img1 = ifft(change_filter(1) * fft_mat)
img2 = ifft(change_filter(2) * fft_mat)
img3 = ifft(change_filter(3) * fft_mat)

cv2.imshow("Original", img)
cv2.imshow("The perfect low pass filter", img1)
cv2.imshow("Butterworth Low Pass Filter", img2)
cv2.imshow("Gaussian low-pass filtering", img3)
cv2.waitKey()
```