

Федеральное агентство по образованию Российской Федерации
Государственное образовательное учреждение
высшего профессионального образования
Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Отчёт по лабораторной работе №3
«Применение преобразования Фурье для уничтожения полос на
снимках электронной микроскопии»

Выполнил:
студент ф-та ИИТММ
гр. 381908-1
Дыдыкин П.С.

Проверила:
ассистент кафедры МОСТ, ИИТММ
Гетманская А.А.

Содержание

Оглавление

Постановка задачи	3
Разработка кода.....	4
Заключение.....	5
Приложение.....	6
Код программы	6
Результаты эксперимента.....	7

Постановка задачи

Имеется несколько фотографий электронной микроскопии с полосами. Необходимо применить преобразование Фурье и специальные фильтры для устранения данных дефектов.

Разработка кода

Для выполнения лабораторной работы необходимо было реализовать прямое и обратное дискретные преобразования Фурье и Notch-фильтр. Также для улучшения итогового результата я применил реализацию фильтра Гаусса.

Код программы и вышеупомянутых функций представлены в репозитории (файл Lab_3.py), а также в разделе «Приложение»

Первым делом необходимо определить явным образом директорию, где расположены тестовые файлы с фотографиями микроскопии. Далее получаем массив данных из .png изображений.

Проходимся циклом по этому массиву, применяем сначала дискретное преобразование Фурье, затем применяем реализацию Notch-фильтра и при помощи функции **reverseDFFTnp(dfft)** получаем итоговое изображение.

Для улучшения итогового результата применяем фильтр Гаусса и выводим результаты эксперимента.

Заключение

Благодаря данной лабораторной работе я узнал больше о преобразовании Фурье, а также о его практическом применении.

Приложение

Код программы

```
1  import cv2 as cv
2  import numpy as np
3  from matplotlib import pyplot as plt
4  from matplotlib.colors import LogNorm
5  import glob
6
7
8  def DFFTNp(img):
9      f = np.fft.fft2(img)
10     fshift = np.fft.fftshift(f)
11     return fshift
12
13
14  def reverseDFFTNp(dfft):
15      f_ishift = np.fft.ifftshift(dfft)
16      reverse_image = np.fft.ifft2(f_ishift)
17      return reverse_image
18
19  def Gauss(img):
20      ksize = 10
21      kernel = np.zeros(img.shape)
22
23      blur = cv.getGaussianKernel(ksize, -1)
24      blur = np.matmul(blur, np.transpose(blur))
25      kernel[0:ksize, 0:ksize] = blur
26
27      fkshift = DFFTNp(kernel)
28      mult = np.multiply(fshift, fkshift)
29
30      reverse_image = reverseDFFTNp(mult)
31      return reverse_image
32
33
34  folder_path = r"C:\stripes/"
35  images = glob.glob(folder_path + '*.png')
36  for image in images:
37      img = np.float32(cv.imread(image, 0))
38      fshift = DFFTNp(img)
39
40      magnitude_spectrum = 20*np.log(np.abs(fshift))
41      plt.subplot(121), plt.title('Magnitude Spectrum')
42      plt.imshow(magnitude_spectrum)
43
44      w, h = fshift.shape
45      maxp = fshift[w//2][h//2]
46      for i in range(w):
47          for j in range(h):
48              if i != w//2 and j != h//2:
49                  if abs(np.abs(fshift[i][j]) - np.abs(maxp)) < np.abs(maxp) - 200000:
50                      fshift[i][j] = 0
51
```

```
52     plt.subplot(122), plt.title('Notch')
53     plt.imshow(np.abs(fshift), norm=LogNorm(vmin=5))
54     plt.show()
55
56     reverse_image = reverseDFFTNp(fshift)
57     reverse_image = Gauss(reverse_image)
58
59     plt.subplot(121), plt.title('Before')
60     plt.imshow(abs(img), cmap='gray')
61     plt.subplot(122), plt.title('After')
62     plt.imshow(abs(reverse_image), cmap='gray')
63     plt.show()
64
```

Результаты эксперимента

