**Дата: 12.06.23**

**ФИО: Козлов Евгений Юрьевич**

**Группа: 224-322**

**ПРАКТИЧЕСКАЯ РАБОТА №****4**  
**Применение пространственных фильтров размытия и повышения  
резкости**

**1. Цель работы**  
Познакомится с пространственными методами фильтрации.

**2. Исходные данные и программное обеспечение**

Используемая среда программирования: Visual Studio Code

Используемый язык программирования: Python 3.11.1 64-bit

Используемые библиотеки: numpy, scipy, skimage, matplotlib

**3. Выполнение работы**

**Формирование функциии периодического прямоугольного сигнала.**

Периодический прямоугольный сигнал формируется из значений {0.0, 1.0} замещается с помощью библиотеки NumPy.

**Получение изображения, заданного функцией из п. 1**

Изображение 200×200px с глубиной цвета 32 bpp (float32). Период 25px, и ширина полосы 4px:

A picture containing text, screenshot, pattern, rectangle

Description automatically generated

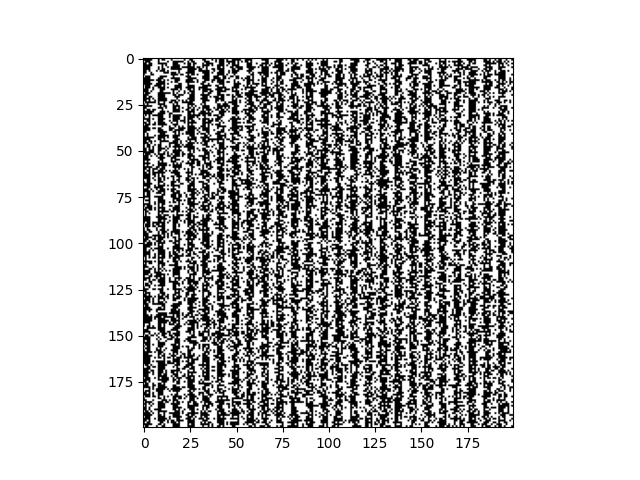
Рисунок 1 – исходное изображение

**Применение к изображению линейных сглаживающих фильтров**

|  |  |  |  |
| --- | --- | --- | --- |
| Размер маски | 4x4 | 20x20 | 100x100 |
| Изображение после фильтрации |  |  |  |

**Изображение после добавления импульсного шума**

P=.25



**Параметры медианной фильтрации**

Фильтрация реализована с помощью метода median\_filter из библиотеки scipy

def median\_fn(a,b):

    return sp.ndimage.median\_filter(NOISED, size=(a, b))

**Изображения после медианной фильтрации**

|  |  |  |
| --- | --- | --- |
|  |  |  |

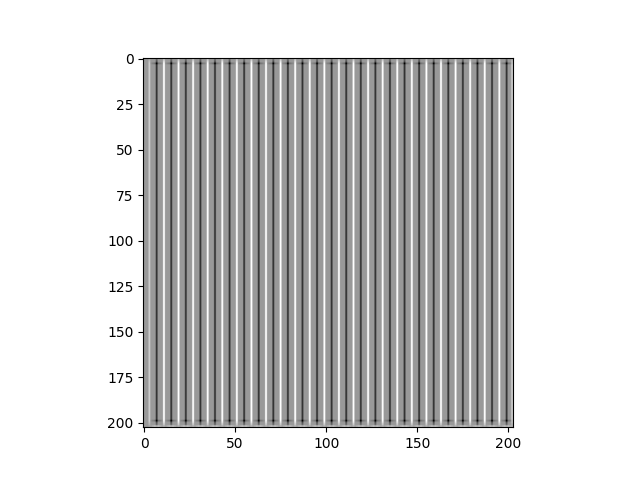
Параметры (a, b) = (4,4), (25,1) и (25, 25) соответственно

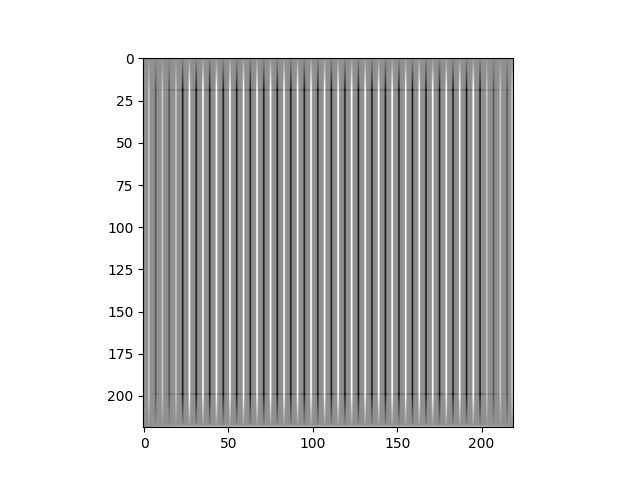
**Параметры фильтра повышения резкости на примере лапласиана**

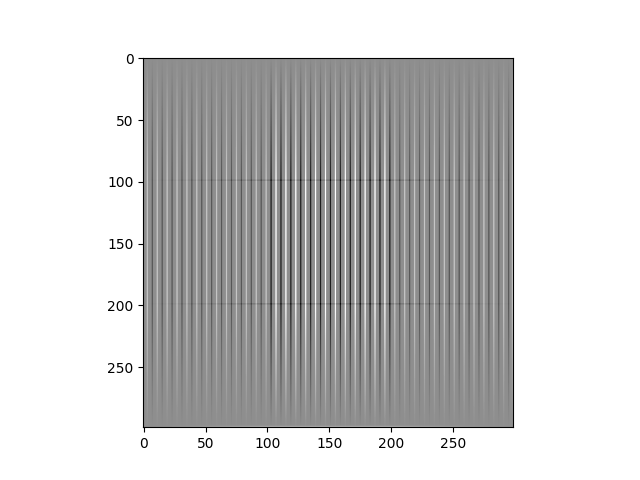
Фильтрация реализована с помощью метода laplace из библиотеки scipy

LAPLAS\_1 = sp.ndimage.laplace(BLURRED\_1)

**Изображения после повышения резкости**

****

****

****

Все изображения хранятся на гугл-диске:

<https://drive.google.com/drive/folders/1Tb3JQ6p5sTJNme5eF--8MToBTEfKQRKw?usp=sharing>

**Код программы:**

import numpy as np

import cv2

import scipy as sp

import matplotlib.pyplot as plt

from skimage.io import imread, imshow, imsave

from skimage import data, img\_as\_float

# Формирование функциии периодического прямоугольного сигнала.

def square\_wave(period, w=200, h=200):

    freq = w // period

    tile = np.repeat((0.0, 1.0), freq // 2)

    line = np.tile(tile, period)

    fig = np.repeat(line, h).reshape(w, h).T

    return fig

# Получение изображения, заданного функцией из п. 1

def show\_n\_save\_figure(fig, path='', cmap='gray'):

    plt.imshow(fig, cmap=cmap)

    plt.savefig(path)

    plt.show()

    plt.close()

INIT\_FIGURE = square\_wave(25)

# Применение к изображению линейных сглаживающих фильтров

# Воспользуемся методом для свёртки из библиотеки scipy

def blur(fig, x, y):

    return sp.signal.convolve2d(fig, np.ones((x, y)) / (x \* y))

BLURRED\_1 = blur(INIT\_FIGURE, 4, 4)

BLURRED\_2 = blur(INIT\_FIGURE, 20, 20)

BLURRED\_3 = blur(INIT\_FIGURE, 100, 100)

# Изображения после размытия

show\_n\_save\_figure(BLURRED\_1, 'img/1/BLURRED\_1.png')

show\_n\_save\_figure(BLURRED\_2, 'img/1/BLURRED\_2.png')

show\_n\_save\_figure(BLURRED\_3, 'img/1/BLURRED\_3.png')

# Изображение после добавления импульсного шума

def noise(fig, p):

    mask = np.random.choice((0, 255), size=fig.shape, p=(1-p, p))

    noised = fig.astype(np.uint8) \* 255 ^ mask

    return (noised // 255).astype(np.float32)

NOISED = noise(INIT\_FIGURE, .25)

show\_n\_save\_figure(NOISED, 'img/2/NOISED.png')

# Параметры медианной фильтрации

def median\_fn(a,b):

    return sp.ndimage.median\_filter(NOISED, size=(a, b))

MEDIAN\_1 = median\_fn(4, 4)

MEDIAN\_2 = median\_fn(25, 1)

MEDIAN\_3 = median\_fn(25, 25)

# Изображение после медианной фильтрации

show\_n\_save\_figure(MEDIAN\_1, 'img/2/MEDIAN\_1.png')

show\_n\_save\_figure(MEDIAN\_2, 'img/2/MEDIAN\_2.png')

show\_n\_save\_figure(MEDIAN\_3, 'img/2/MEDIAN\_3.png')

# Параметры фильтра повышения резкости на примере лапласиана

LAPLAS\_1 = sp.ndimage.laplace(BLURRED\_1)

LAPLAS\_2 = sp.ndimage.laplace(BLURRED\_2)

LAPLAS\_3 = sp.ndimage.laplace(BLURRED\_3)

# Изображения после повышения резкости

show\_n\_save\_figure(LAPLAS\_1, 'img/3/LAPLAS\_1.png')

show\_n\_save\_figure(LAPLAS\_2, 'img/3/LAPLAS\_2.png')

show\_n\_save\_figure(LAPLAS\_3, 'img/3/LAPLAS\_3.png')