

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
Федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»**
(МИ ВлГУ)

Факультет: ИТР

Кафедра: ФПМ

ПРАКТИЧЕСКАЯ РАБОТА № 3

Дисциплина: Цифровая обработка изображений

Тема: Фильтрация изображений

Руководитель:

Мареев А.В.
инициалы, фамилия

подпись, дата

Студент гр. ПМИ-120:

Ширяев А.П.
инициалы, фамилия

подпись, дата

Муром 2023 год

Цель работы: Научиться применять фильтры для изображений.

Ход работы:

Вызов Функций:

```
image = imread('./src/practiceNumberThree/lena.jpg')

# реализация линейного фильтра

imageLineFilter = line_filter(image, 1/16, numpy.array([
    [2, 1, 2],[1, 4, 1],[2, 1, 2]
]))

imsave('./src/practiceNumberThree/lena-liner-filter.jpg', imageLineFilter)

# реализация рекурсивного фильтра

imageRecursiveFilter = recursive_filter(image, 1 / 16, 0.9, numpy.array([
    [2, 1, 2],[1, 4, 1],[2, 1, 2]
]))

imsave('./src/practiceNumberThree/lena-recursive-filter.jpg', imageRecursiveFilter)

# реализация медианной фильтрации

imageMedian = median_filter(image, 7, 7)

imsave('./src/practiceNumberThree/lena-median-filter.jpg', imageMedian)
```

Рисунок 1 – Вызов функций программы в потоке

1. Линейный фильтр:

Код:

```

def line_filter(image: numpy.ndarray, coef: float,
               maska: numpy.ndarray = numpy.array([
                   [1, 1, 1],
                   [1, 1, 1],
                   [1, 1, 1]
               ])) -> numpy.ndarray:

    def filter(image: numpy.ndarray, row: int, col: int) -> numpy.uint8:

        row = numpy.abs(row).astype('int')
        col = numpy.abs(col).astype('int')

        return image[row][col]

    maska = maska.dot(coef)

    rows, cols = maska.shape

    diffirenceRow = numpy.round(rows / 2).astype('int')
    diffirenceCol = numpy.round(cols / 2).astype('int')

    centerFilterRow:int = numpy.round((rows + diffirenceRow) / 2).astype('int')
    centerFilterCol:int = numpy.round((cols + diffirenceCol) / 2).astype('int')

    imageCopy = image.copy()

    for i in range(len(imageCopy) - 1):
        for j in range(len(imageCopy[0]) - 1):
            summa = 0
            for p in range(rows):
                for q in range(cols):
                    summa += filter(
                        imageCopy, i - centerFilterRow + p,
                        j - centerFilterCol + q) * maska[p, q]

            imageCopy[i, j] = numpy.uint8(numpy.round(summa))

    return imageCopy

```

Рисунок 2 – код алгоритма линейной фильтрации

Результат работы:

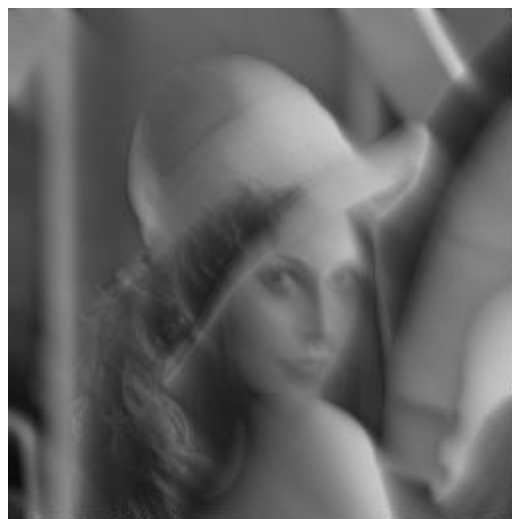


Рисунок 3 – Результат применение линейного фильтра

Из	Лис	№	Подпис	Дат

МИВУ.01.03.02 – 25.000 ПР

Лис

3

2. Рекурсивный фильтр:

Код:

```
def recursive_filter(image: numpy.ndarray, coef: float, k: float = 1, maska: numpy.ndarray = numpy.array([
    [1, 1, 1],
    [1, 1, 1],
    [1, 1, 1]
]))-> numpy.ndarray:
    return line_filter(image, coef, maska).dot(k).astype(numpy.uint8) + line_filter(image, coef, maska).dot(1 - k).astype(numpy.uint8)
```

Рисунок 4 – Код рекурсивного фильтра

Результат:

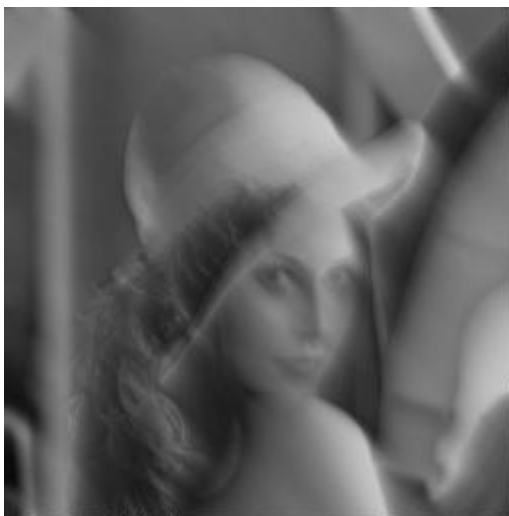


Рисунок 5 – Результат применения рекурсивного фильтра

3. Медианный фильтр:

Код:

```

def median_filter(image: numpy.ndarray, rows:int = 3, cols:int = 3)-> numpy.ndarray:

    imageCopy = image.copy()

    differenceRow = numpy.round(rows / 2).astype('int')
    differenceCol = numpy.round(cols / 2).astype('int')

    centerRow: int = numpy.round((rows + differenceRow) / 2).astype('int')
    centerCol: int = numpy.round((cols + differenceCol) / 2).astype('int')

    for i in range(0, len(imageCopy), rows):
        for j in range(0, len(imageCopy[0]), cols):

            filter = []

            maxIndexRow = 0
            maxIndexCol = 0

            maxValue = 0

            for p in range(rows):
                for q in range(cols):
                    value = imageCopy[numpy.abs(i - centerRow + p).astype('int'),
                                      numpy.abs(j - centerCol + q).astype('int')]

                    filter.append(value)

                    if maxValue <= value:
                        maxValue = value
                        maxIndexCol = numpy.abs(j - centerCol + q).astype('int')
                        maxIndexRow = numpy.abs(i - centerRow + p).astype('int')

            imageCopy[maxIndexRow, maxIndexCol] = numpy.median(filter).astype(numpy.uint8)

    return imageCopy

```

Рисунок 6 – Код медианного фильтра

Результат:



Рисунок 7 – Результат применения медианного фильтра

Вывод: В проделанной работе были применены фильтры для изображения лена. Применены такие фильтры как: линейный фильтр, рекурсивный фильтр и медианный фильтр.

					МИВУ.01.03.02 – 25.000 ПР	Лис
Из	Лис	№	Подпис	Дат		6