

Звіт до лабораторної роботи з дисципліни «Обробка зображень»
Студента групи ІПС-32
Тоцького Олександра Павловича

Постановка задачі:

В зображеннях Saturn2.gif та saturn3.gif подавити імпульсний шум.
Описати алгоритм подавлення імпульсного шуму та програмно реалізувати його.

Опис алгоритму (Алгоритм медіанної фільтрації):

Проходимо по всім елементам нашої матриці зображення та додаємо в наш масив `temp_array` елементи, які утворюють квадрат, довжина якого 5 точок, навколо нашої поточної точки. Сортуюмо `temp_array` та беремо медіану з цього масиву. Цей результат записуємо на місце нашої поточної точки, для якої ми робили `temp_array`.

Виконавши всі ітерації, отримаємо зображення з подавленим імпульсним шумом.

Код алгоритму:

```
import numpy
import math
from PIL import Image

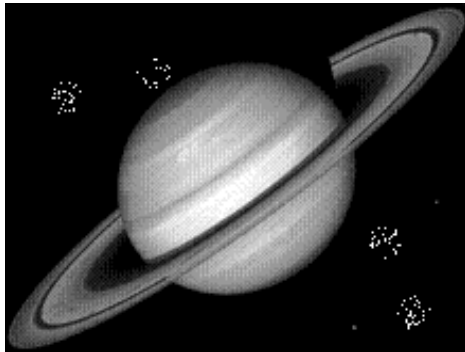
filter_size = 5 # the size of our square

def filter_median(data):
    temp_array = []
    indexer = math.floor(filter_size / 2)
    final_result = numpy.zeros((len(data), len(data[0])))
    for i in range(len(data)):
        for j in range(len(data[0])):
            for z in range(filter_size):
                if 0 <= i + z - indexer <= len(data) - 1:
                    for k in range(filter_size):
                        if 0 <= j + k - indexer <= len(data[0]) - 1:
                            temp_array.append(data[i + z - indexer][j +
k - indexer])
            temp_array.sort()
            final_result[i][j] = temp_array[math.floor(len(temp_array)
/ 2)]
            temp_array = []
    return final_result
```

```
def main():
    # img = Image.open("Saturn2.gif").convert("L") # translating a
    color image to greyscale
    img = Image.open("saturn3.GIF").convert("L")
    arr = numpy.array(img)
    removed_noise = filter_median(arr)
    img = Image.fromarray(removed_noise)
    img.show()

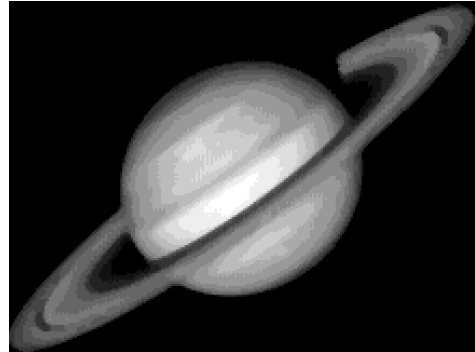
main()
```

Результати перетворень:



Saturn2.gif

->



saturn3.gif

->

