ГУАП

КАФЕДРА № 44

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент, канд. техн. наук |  |  |  | Н. В. Соловьёв |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 |
| РАЗРАБОТКА WINDOWS-ПРИЛОЖЕНИЯ |
| по курсу: ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4841 |  |  |  | Н. А. Скрябин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2021

1. **Индивидуальное задание**

Повышение резкости фильтром Лапласа и нечетким маскированием

1. **Теоретические положения**

Для увеличения резкости можно использовать вторую производную яркости изображения, т.е. результат свертки изображения с фильтром Лапласа. Тогда



где kL – коэффициент усиления резкости фильтром Лапласа, выбираемый экспериментально из диапазона kL =1…5, zL – результат свертки по (1.2) окрестности 3×3 обрабатываемого пикселя яркостью z с ядром HL (zL берется с учетом знака!). Дискретные маски фильтра Лапласа HL приведены в (2.5).

Если перепадов яркости в окрестности нет, то zL=0 и z’=z. Если центр окрестности находится в начале границы перехода от низкой к высокой яркости, то zL > 0 и согласно (3.10) z’ < z. Если центр окрестности в конце этой границы, то zL < 0 и соответственно z’>z. Следовательно, диапазон яркости границы увеличивается приблизительно на 2kLzL.

Подчеркивание границ фрагментов можно получить, используя методы нечеткого маскирования. Например, в результате свертки изображения с гауссианом можно получить размытое изображение. Маска гауссиана размером 3×3 приведена в (1.8). Тогда



где kG – коэффициент усиления резкости Гауссианом, выбираемый экспериментально из диапазона kG = 1…10, zG – результат свертки с Гауссианом. Если перепадов яркости в окрестности центрального пикселя с яркостью z нет, то z = zG и согласно (3.11) z’ = z, т.е. изменения яркости пикселя на преобразуемом изображении не происходит. Если центр окрестности расположен в начале границы перехода от низкой к высокой яркости, то zG > z и по (3.11) z’ < z при kG > 1. Если центр окрестности в конце этой границы, то zG < z и соответственно z’ > z. Следовательно, диапазон яркости границы увеличивается примерно на 2(kG – 1)|z –zG|.

1. **Процесс выполнения работы**

Для повышения резкости фильтром Лапласа и нечётким маскированием необходимо для каждого пикселя посчитать изменённое значение яркости по формуле.

Срез яркости берётся по средней вертикальной линии изображения.

Проблема выхода маски за пределы изображения решается проходом, начиная с элемента (1;1) и заканчивая элементом (image.shape[0] – 2;image.shape[1] - 2).

1. **Код программы**

import cv2  
import math  
from PyQt6.QtCore import QSize, Qt  
from PyQt6.QtWidgets import QApplication, QMainWindow, QPushButton, QLabel, QLineEdit, QVBoxLayout, QWidget  
import sys  
  
  
# Загрузка изображения  
def load\_image(text):  
 text += '.jpg'  
 image = imread(f'{text}', IMREAD\_GRAYSCALE)  
 return image  
  
  
# Лаплас 3x3  
def increase\_sharpness\_laplas(image, text):  
 mask\_laplas = ((0, 1, 0), (1, -4, 1), (0, 1, 0))  
 k = 1  
 result\_image = image.copy().astype(np.int32)  
  
 for i in range(1, image.shape[0] - 2):  
 for j in range(1, image.shape[1] - 2):  
  
 z\_x = 0  
  
 for di in range(-1, 2):  
 for dj in range(-1, 2):  
 z\_x += image[i + di][j + dj] \* mask\_laplas[1 + di][1 + dj]  
  
 result\_image[i][j] = image[i][j] - k \* z\_x  
 min\_result\_image = min(result\_image.ravel())  
 coeff = (max(result\_image.ravel()) + abs(min\_result\_image)) / 256  
  
 for i in range(1, result\_image.shape[0] - 2):  
 for j in range(1, result\_image.shape[1] - 2):  
 result\_image[i][j] += abs(min\_result\_image)  
 result\_image[i][j] = int(result\_image[i][j] / coeff)  
  
 result\_image = result\_image.astype(np.uint8)  
 imshow('result-laplas', result\_image)  
 waitKey(0)  
 imwrite(f'{text}-laplas.jpg', result\_image)  
 return result\_image  
  
  
# Обработка изображения методом нечёткого маскирования  
def mask(image, text):  
 result\_image = image.copy()  
 result\_image = result\_image.astype(np.int32)  
 mask\_gauss = ((1, 2, 1), (2, 4, 2), (1, 2, 1))  
 k = 9  
  
 for i in range(1, image.shape[0] - 2):  
 for j in range(1, image.shape[1] - 2):  
  
 z\_gaussian = 0  
  
 for di in range(-1, 2):  
 for dj in range(-1, 2):  
 z\_gaussian += image[i + di][j + dj] \* mask\_gauss[1 + di][1 + dj]  
 result\_image[i][j] = z\_gaussian + 2 \* (image[i][j] - z\_gaussian)  
  
 min\_result\_image = min(result\_image.ravel())  
 coeff = (max(result\_image.ravel()) + abs(min\_result\_image)) / 256  
  
 for i in range(1, result\_image.shape[0] - 2):  
 for j in range(1, result\_image.shape[1] - 2):  
 result\_image[i][j] += abs(min\_result\_image)  
 result\_image[i][j] = int(result\_image[i][j] / coeff)  
  
 result\_image = result\_image.astype(np.uint8)  
 imshow('result-mask', result\_image)  
 waitKey(0)  
 imwrite(f'{text}-mask.jpg', result\_image)  
 return result\_image

class MainWindow(QMainWindow):  
  
 def \_\_init\_\_(self):  
 super(MainWindow, self).\_\_init\_\_()  
  
 self.button\_is\_checked = True  
  
 self.setWindowTitle("Matrix 3x3 income/result")  
  
 self.label = QLabel()  
  
 self.input = QLineEdit()  
 self.input.setObjectName("host")  
 self.input.setText("host")  
 self.input.setMaxLength(10)  
 self.input.textEdited.connect(self.label.setText)  
  
 self.button3 = QPushButton("3x3")  
 self.button3.clicked.connect(self.button\_3)  
  
 self.button7 = QPushButton("7x7")  
 self.button7.clicked.connect(self.button\_7)  
  
 layout = QVBoxLayout()  
 layout.addWidget(self.input)  
 layout.addWidget(self.label)  
 layout.addWidget(self.button3)  
 layout.addWidget(self.button7)  
  
 container = QWidget()  
 container.setLayout(layout)  
  
 self.button = QPushButton("Start calculating")  
 self.button.clicked.connect(self.button\_is\_checked)  
  
 # Устанавливаем центральный виджет Window  
 self.setCentralWidget(container)  
  
 def set\_text(self, text):  
 self.label.setText(text)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 # Передаём sys.argv, чтобы разрешить аргументы командной строки для приложения  
 app = QApplication(sys.argv)  
  
 # Создаём виджет Qt - окно  
 window = MainWindow()  
 window.show()  
  
 # Цикл событий  
 app.exec()

1. **Результаты выполнения программы**



Рисунок 1 – исходное изображение 1

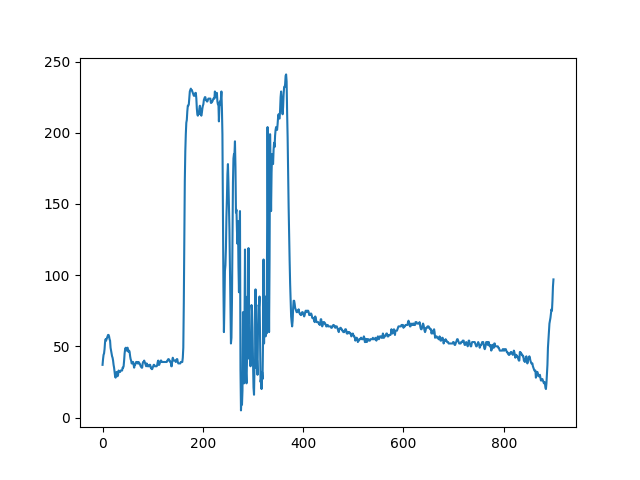


Рисунок 2 – срез яркости исходного изображения 1



Рисунок 3 – изображение 1, обработанное фильтром Лапласа

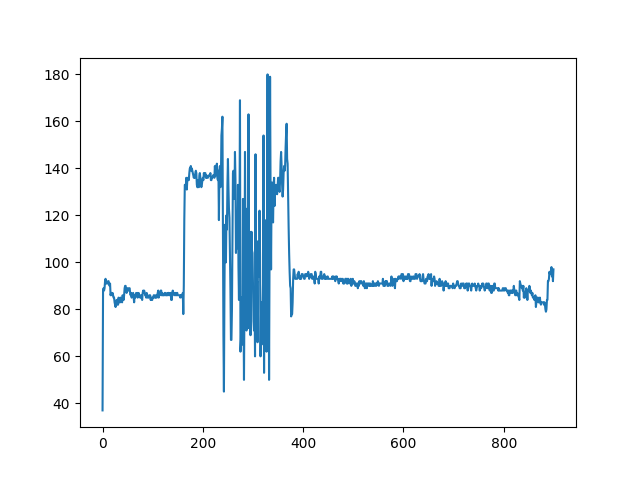


Рисунок 4 – срез яркости изображения 1, обработанного фильтром Лапласа



Рисунок 5 – изображение 1, полученное методом нечеткого маскирования

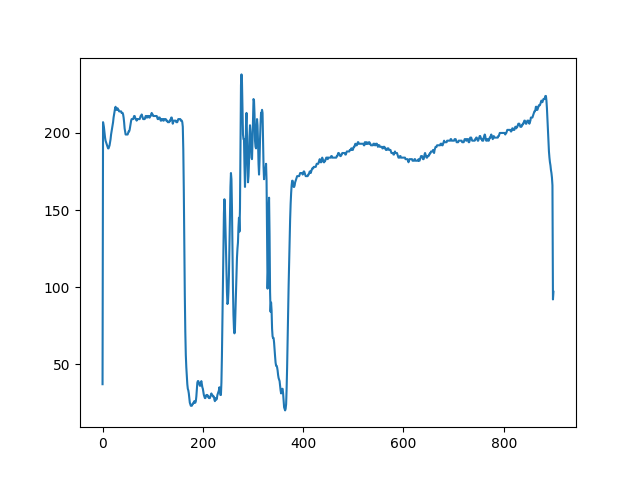


Рисунок 6 – срез яркости изображения 1, обработанного с помощью нечеткого маскирования



Рисунок 7 – исходное изображение 2

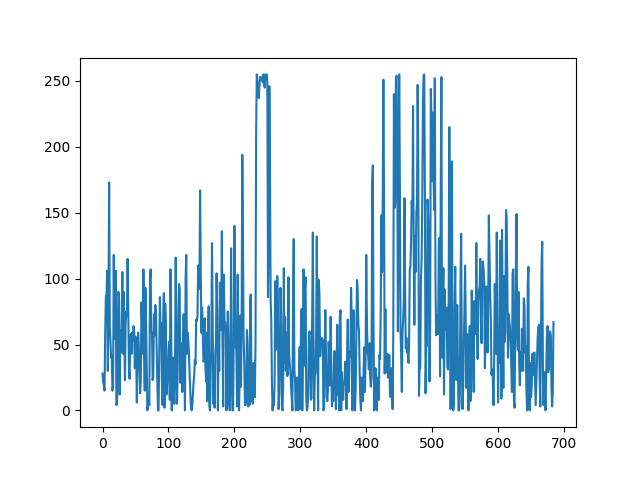


Рисунок 8 – срез яркости изображения 2

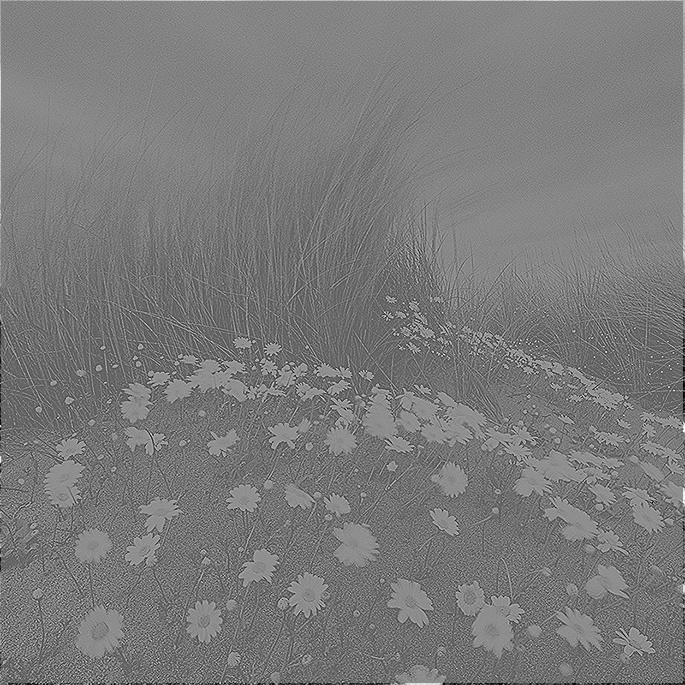


Рисунок 9 –изображение 2, обработанное фильтром Лапласа

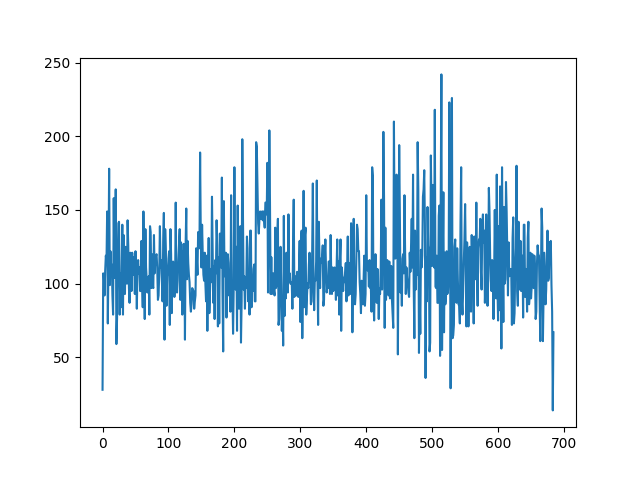


Рисунок 10 – срез яркости изображения 2, обработанного фильтром Лапласа

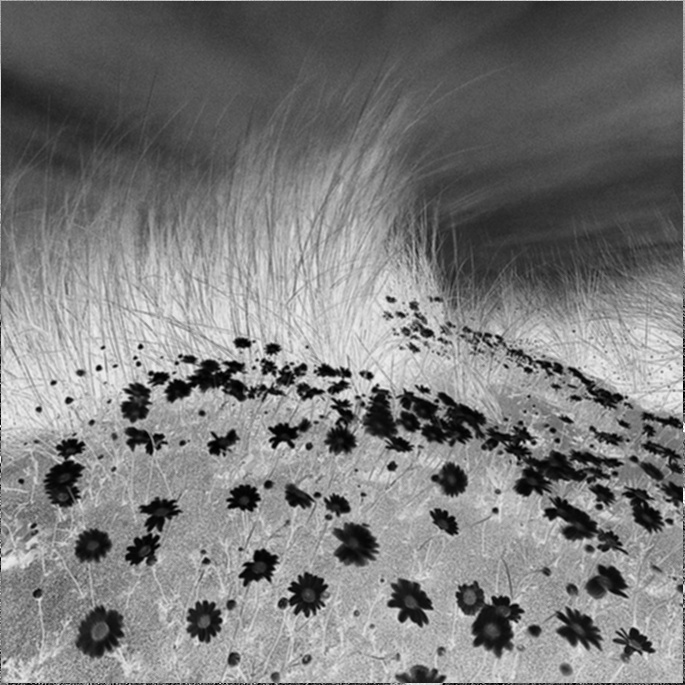


Рисунок 11 – изображение 2, обработанное с помощью нечеткого маскирования

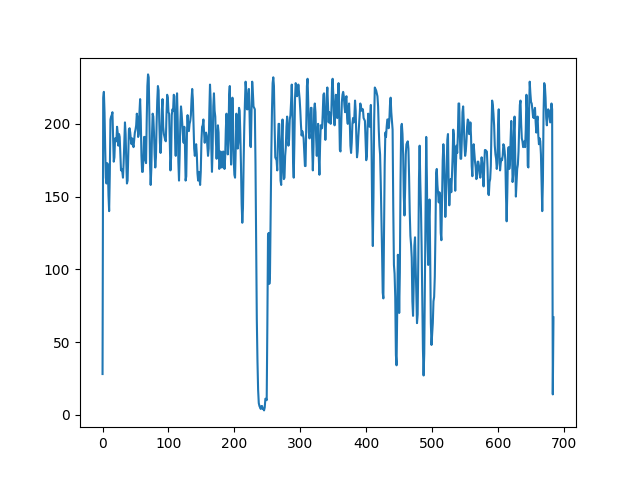


Рисунок 12 – срез яркости изображения 2, обработанного с помощью нечеткого маскирования

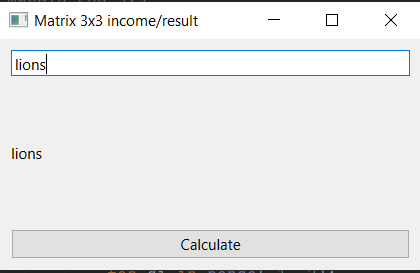
****

Рисунок 13 – интерфейс Windows-приложения

1. **Вывод**

Выполнена лабораторная работа “Разработка Windows-приложения”, получены навыки повышения резкости с помощью фильтра Лапласа и метода нечеткого маскирования.