



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ
НА ТЕМУ:
“Разработка алгоритма
сглаживания изображений”**

Студент

_____ Челядинов И.Д. _____
подпись, дата фамилия, и.о.

Группа ИУ7-53Б

Руководитель практики

_____ Майков К.А. _____
подпись, дата фамилия, и.о.

Москва, 2020 г.

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7
(Индекс)

И.В.Рудаков
(И.О.Фамилия)

« ____ » _____ 2020г.

З А Д А Н И Е на выполнение курсового проекта

по дисциплине Компьютерная графика

Студент группы ИУ7-536

Челядинов Илья Дмитриевич
(Фамилия, имя, отчество)

Тема курсового проекта Программная реализация эффективного алгоритма сглаживания изображений.

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 11 нед., 100% к 14 нед.

1. *Техническое задание*

Разработать программную реализацию эффективного алгоритма сглаживания изображений. Исходное изображение содержит комбинированный шум. Изображение загружается в формате PNG. Исходное изображение не изменяется, все операции производятся над копией.

2. *Оформление курсового проекта*

2.1. Расчетно-пояснительная записка на 25-30 листах формата А4.

Расчетно-пояснительная записка должна содержать постановку введение, аналитическую часть, конструкторскую часть, технологическую часть, экспериментально-исследовательский раздел, заключение, список литературы, приложения.

2.2. Перечень графического материала (плакаты, схемы, чертежи и т.п.) На защиту проекта должна быть представлена презентация, состоящая из 15-20 слайдов. На слайдах должны быть отражены: постановка задачи, использованные методы и алгоритмы, расчетные соотношения, структура комплекса программ, диаграмма классов, интерфейс, характеристики разработанного ПО, результаты проведенных исследований.

Дата выдачи задания « ____ » _____ 20__ г.

Руководитель курсового проекта

(Подпись, дата)

Майков К.А.
(И.О.Фамилия)

Студент

(Подпись, дата)

Челядинов И.Д.
(И.О.Фамилия)

Оглавление

Введение	4
1. Аналитическая часть	5
1.1 Анализ шумов. Понятие шума.....	5
1.2 Анализ алгоритмов, предназначенных для удаления Гауссова шума	6
1.3 Анализ алгоритмов, предназначенных для удаления импульсного шума ...	9
1.4 Комбинированные (гибридные фильтры)	11
1.5 Нелокальный алгоритм сглаживания.....	12
2. Конструкторская часть	13
2.1 Нелокальный алгоритм удаления шума.....	14
2.2 Общий алгоритм программы.....	15
3. Технологическая часть	15
3.1 Выбор языка, среды и платформы программирования.....	16
3.2 Структура и состав классов.....	17
3.3 Сведения о модулях программы.....	18
3.4 Описание пользовательского интерфейса.....	18
4. Экспериментально - исследовательский раздел	19
4.1 Технические характеристики	20
4.2 Цель эксперимента	20
4.3 Описание и выполнение эксперимента	20
4.4 Вывод из результата исследований	23
5. Заключение	23
Список использованной литературы	24

Введение

В современном мире алгоритмы сглаживания являются невероятно важной частью человеческой жизни. Мы часто не задумываемся, какие операции проходит картинка перед тем, как мы ее видим.[1] Алгоритмы сглаживания применяются почти везде: они могут быть самостоятельной процедурой, для избавления фотографии от ненужных шумов, могут являться первым шагом для более сложных процедур, например, для распознавания объектов на изображении, также могут использоваться для улучшения качества графики в играх, и даже для обработки фотографий из космоса. Развитие области фотографии привело к необходимости автоматизации процессов избавления изображений от шумов без потери качества.

В данный момент существует множество способов фильтрации шумов, каждый из которых обладает своими достоинствами и недостатками. Как правило, обычные сглаживающие фильтры корректируют значение пикселя, ориентируясь на информацию о соседних пикселях. Однако такие фильтры не всегда дают качественный результат, ведь на изображении могут находиться объекты, границы которых будут игнорированы, кроме того, обработка контрастных изображений тоже является проблемой.

Моей задачей во время выполнения курсовой работы был выбор и модифицирование существующего алгоритма сглаживания и его реализация. В качестве такого алгоритма был выбран нелокальный алгоритм, который для сглаживания использует информацию на всем изображении, а не только о близлежащих пикселях от выбранного. Для оптимизации времени выполнения алгоритма работа ведется с изображениями в оттенках серого цвета.

1. Аналитическая часть

1.1 Анализ шумов. Понятие шума.

Из – за несовершенства реального мира мы часто можем получить изображения плохого качества, содержащее в себе различного разного рода помехи. Такие помехи называются шумом.

Цифровой шум – дефект изображения, вносимый фотосенсорами и электроникой устройств, которые их используют (цифровой фотоаппарат, теле-/видеокамеры и т.п.) вследствие несовершенства технологий, а также фотонной природы света.

Цифровой шум заметен на фотографии в виде наложенной матрицы пикселей случайного цвета и яркости. То есть, в идеальном мире мы бы получили фотографию с идеальной матрицей пикселей I , но, в реальном мире мы получаем изображение $I + n$, где n – матрица шума.

Цифровой шум можно разделить на три группы:

- Гауссов шум
- Импульсный шум
- Комбинированный шум

Гауссов шум – это статистический шум, с плотностью распределения равной плотности вероятности нормального (Гауссова) распределения. Возникает из – за уровня освещенности или нагрева аппаратуры.

Импульсный шум – шум, характеризующийся заменой части пикселей на изображении значениями фиксированной или случайной величины, которые изолированы контрастными точками. Возникает из – за ошибок, возникающих при передаче изображений.

Комбинированный шум – шум, включающий в себя Гауссов шум и импульсный шум. В реальных фотографиях нам чаще всего приходится иметь дело именно с ним.

Алгоритмы устранения шума получили должное внимание в конце восьмидесятых. К сожалению, большинство алгоритмов специализируются на

удалении конкретного вида шума. Не существует универсальных фильтров, удаляющих все шумы.[2]

Цифровой шум может появиться из – за:

- Дефекты потенциального барьера вызывают утечку заряда, сгенерированного за время экспозиции. Проявляется в виде черных точек
- Шума квантования аналого-цифрового преобразователя – ошибки, возникающие при квантовании сигнала;
- Шумы, сгенерированные из – за взаимодействия фотонов света с атомами материала фотодиодов сенсора;
- Шумы, возникающие из – за неработающих пикселей матрицы фотоэлемента;
- Сжатия изображения.

Также выделим понятие аналогового шума:

Аналоговый шум – любые неисправности техники, с помощью которой было сделано изображение. К данному виду можно отнести:

- зернистость пленки;
- грязь;
- пыль;
- царапины;
- отслоение фотографической эмульсии;

Также необходимо выделить шумы, возникающие при рендеринге сгенерированных компьютером изображений.

В данной работе рассматривается удаление цифрового шума.

1.2 Анализ алгоритмов, предназначенных для удаления Гауссова шума

При выборе алгоритма удаления Гауссова шума нужно будет учесть особенности поставленной задачи. Проблема в том, что при удалении Гауссова шума фильтры обычно размывают границы объектов. Также высокая контрастность изображения тоже является проблемой для фильтров,

работающих для Гауссова шума. Для удаления такого вида шума используются как линейные, так и нелинейные алгоритмы.

Линейные алгоритмы:

Линейные фильтры также называются сглаживающими или усредняющими, потому что суть их работы крайне проста – они усредняют значения пикселей, находящихся в апертуре, таким образом сглаживая изображение.[3] В основе линейных фильтров находится некая функция, называемая ядром, которая применяется к апертуре. Апертюра с примененной к ней функцией называется маской. Применение линейных алгоритмов приводит к сглаживанию резких границ объектов. Они являются подходящими при гауссовом распределении помех. Линейные алгоритмы показывают высокие результаты при таком распределении шумов.

$$A_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad A_2 = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$
$$A_3 = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}.$$

Рисунок 1.1. Примеры масок.

Фильтр скользящего среднего

При его использовании значение каждого фильтра заменяется разновидностью среднего от значений соседних пикселей. Огромным минусом данного фильтра является размывание краев изображений, а значит, он не подходит для обработки реальных изображений. Однако фильтр дает неплохой результат при работе с гауссовым шумом. На рисунке 1.2 видно, что фильтр действительно справляется с гауссовым шумом, однако его применение ведет к потере качества.



А



В

Рисунок 1.2. (А) – исходное изображение. (В) – результат работы сглаживающего фильтра.

Фильтр Гаусса

Отличие фильтра Гаусса от фильтра скользящего среднего заключается в том, что в маске (апертуре с определенной функцией) используется функция Гаусса.

Благодаря данному фильтру можно уменьшить влияние пикселей друг на друга с расстоянием. Фильтр Гаусса действительно может удалять шум, подвергая границы объектов размыванию и приводя к потере качества в малой степени, что видно на рисунке 1.3.



А

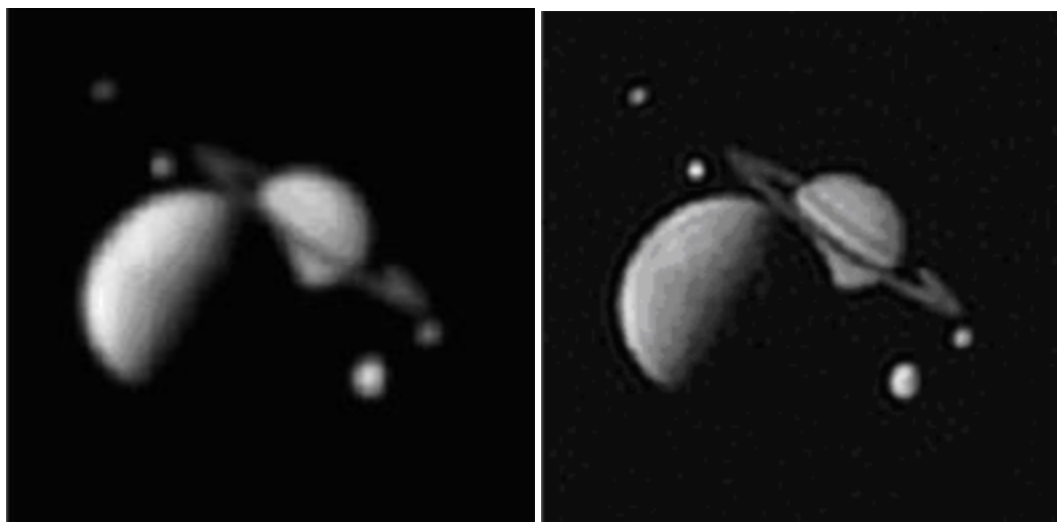


Б

Рисунок 1.3. (А) -исходное изображение. (В) – изображение, подвергнутое фильтру Гаусса.

Фильтр Винера

Метод Винера широко применяется для избавления изображений от шума. Он работает в частотной области, пытаясь снизить влияние деконволюционного шума на частотах с плохим отношением шум/сигнал. Метод Винера действительно дает хороший результат на большинстве изображений.



А

В

Рисунок 1.4. (А) – исходное изображение. (В) – восстановленное изображение.

Вывод

Линейные алгоритмы отлично подходят для коррекции изображений с шумом Гаусса. Однако, такие алгоритмы плохо справляются с импульсным шумом. В реальной жизни мы чаще всего сталкиваемся именно с шумом Гаусса, поэтому, чаще всего, для решения жизненных задач подойдут линейные алгоритмы.

1.3 Анализ алгоритмов, предназначенных для удаления импульсного шума

Нелинейные алгоритмы:

В последние два десятилетия наиболее активно развиваются нелинейные алгоритмы. Они основываются на ранговой статистике для удаления шумов. Благодаря нелинейным алгоритмам удастся избежать искажения изображений при повторной обработке. Нелинейные алгоритмы показывают хорошие результаты работы в случаях, когда нам нужно выделить границы объекта.

При обработке реальных изображений мы сталкиваемся с различного рода помехами, такими как перепады яркости, резкие переходы контрастности, поэтому обработка линейными фильтрами чаще всего даст плохой результат. На помощь приходят нелинейные фильтры. Стоит отметить, что нелинейные алгоритмы удаления шума имеют большие вычислительные затраты, чем линейные, поэтому, чаще всего, они проигрывают в скорости работы.

Медианная фильтрация

При обработке методом медианной фильтрации мы рассматриваем каждый пиксель в кадре, а для оценки мы используем некоторое пространство, называемое окном.[4] Часто применяются варианты окна в виде креста или квадрата. Размеры окна различаются в зависимости от характера изображения.

Идея фильтрации проста: из группы входных значений выбирается среднее и выдается на выход. С помощью медианной фильтрации можно действительно более эффективно устранить импульсные помехи, чем с помощью линейных фильтров, однако в результате работы на изображении может появиться гауссов шум.

Достоинства медианной фильтрации:

- Простая структура реализации;
- Отлично работает при наличии одиночного импульсного шума.

Недостатки медианной фильтрации:

- Нелинейна, что может усложнить математический анализ сигналов;
- Медианная фильтрация приводит к размытию контуров деталей;
- Подавление белого и гауссова шума менее эффективно, чем у линейных фильтров

Ранжирующий фильтр

Ранжирующий фильтр, как и сглаживающий, использует маску. Маска может включать или не включать центральный пиксель. Значения элементов, попадающих в маску, сортируются (по возрастанию или по убыванию), затем можно определить значение интенсивности или дисперсии. Выходным значением является взвешенная сумма интенсивности центрального пикселя и получившегося ряда.

Билатеральный фильтр

Данный фильтр комбинирует пространственную и яркостную фильтрацию. То есть, учитывается не только значения лежащих рядом пикселей, но и расстояние до текущего фильтруемого пикселя. В настоящее время алгоритм нашел широкое применение в сферах обработки изображения в связи с тем, что, благодаря такой обработке, сохраняются края объектов. Большой недостаток – вычислительные затраты.[5]

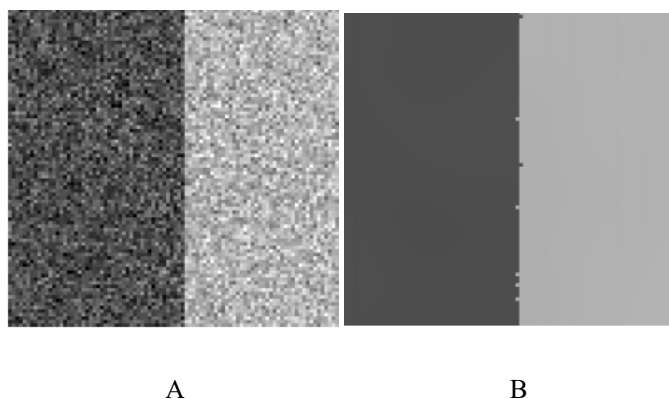


Рисунок 1.5. (А) – зашумленное изображение. (В) – результат работы билатерального фильтра.

Вывод

Нелинейные фильтры показывают отличные результаты при коррекции импульсного шума. Они обладают высокой эффективностью, однако, в сравнении с линейными, хуже справляются с гауссовым шумом.

1.4 Комбинированные (гибридные фильтры)

Для удаление комбинированных шумов в начале последнего десятилетия было предложено совместное применение смешанного и медианного фильтров. Такие фильтры называются комбинированными или гибридными.

Идея состоит в том, чтобы использовать на зашумленном изображении сначала медианный, а потом линейный фильтр. Таким образом, получается удалить и высококонтрастный импульсный шум, затем удалить гауссов шум. Таким образом, комбинированные фильтры превосходят медианные, потому что позволяют уменьшить искажение контуров малых объектов, однако, проигрывают во времени.

1.5 Нелокальный алгоритм сглаживания

Рассматриваемый мной к реализации алгоритм отличается от вышеперечисленных тем, что рассматривает не только информацию о близлежащих пикселях, а использует всю информацию на изображении. Если у нас есть несколько одинаковых изображений с различным уровнем шума, мы можем удалить шум. Нелокальный фильтр использует похожую идею. Однако, если изображения похожи, не означает, что объекты одинаковые, но, как правило, приближение является достаточно хорошим. Таким образом, фильтр отлично работает на изображениях, с большим количеством похожих элементов.

Сложность данного алгоритма – $O(n^2(2r+1))$, где r – радиус окна, по которому вычисляется “похожесть” частей изображения, n – количество пикселей.[6][7]

Вывод

Нелокальный алгоритм отлично подходит для корректирования изображений, содержащих множество повторяющихся объектов. Кроме того, данный алгоритм справляется с задачей сглаживания лучше линейных фильтров. Данный алгоритм был выбран для реализации.

2. Конструкторская часть

2.1 Нелокальный алгоритм удаления шума.

Для корректной работы алгоритма необходимо произвести проход по пикселям изображения, вычисляя матрицу весов. Матрица весов позволяет увидеть количество шума на изображении.

Для данного алгоритма мы вводим понятие окон. Выделяем область пикселей, называемую окном, сравниваем ее с другой областью. Для этого нам понадобится функция различия двух пикселей. Для черно-белых изображений используем модуль разности двух пикселей.[8] Полученную величину преобразуем к формату веса.

Математическая модель нелокального фильтра:

$$I_d(j) = \sum_{j \in I} w(i, j) I(j)$$

i -тый пиксель результирующего изображения равен сумме всех пикселей исходного изображения, взятых с весами W , где вес это

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|I(N_i) - I(N_j)\|_2^2}{h^2}}$$

A нормирующий делитель

$$Z(i) = \sum_{j \in I} e^{-\frac{\|I(N_i) - I(N_j)\|_2^2}{h^2}}$$

Альтернативная метрика:

$$\rho(x, y, t) = \begin{cases} 0, & \text{if } |x - y| < t. \\ |x - y| - t, & \text{otherwise.} \end{cases}$$

Сложность данного алгоритма – $O(n^2(2r+1))$.

Можно ускорить работу алгоритма, но при этом пожертвовать памятью: не пересчитывать веса все время, а хранить их в матрице весов, ведь для пикселя i

и для пикселя j и наоборот веса равны. В таком случае время выполнения сократится вдвое, но понадобится $O(n^2)$ памяти.

2.2 Общий алгоритм программы.

Работу программы можно описать следующим алгоритмом:

1. Запуск программы.
2. Выбор изображения.
3. Инициализация изображения.
4. Ввод коэффициента сглаживания, радиуса окна, сигмы.
5. Расчет матрицы весов.
6. Преобразование изображения, согласно рассчитанной матрице весов.
7. Сохранение изображения.
8. Завершение.

Для инициализации и сохранения изображения используются встроенные модули языка программирования. На этапе выбора изображения пользователю предлагается настроить различные параметры, которые отвечают за меру сглаживания изображения.

3. Технологическая часть

Этот раздел содержит выбор средств программной реализации, описание основных моментов программной реализации и методики работы с созданным программным обеспечением.

3.1 Выбор языка, среды и платформы программирования.

Для разработки программы был выбран объектно-ориентированный подход. Он позволяет обеспечить общий интерфейс взаимодействия с различными объектами, а также позволяет сделать программу гибкой для дальнейшего расширения и модификации, что является весомым аргументом для выбора данного подхода.

Языком программирования разрабатываемого приложения был выбран Python, в сочетании с языком программирования Cython. Cython – язык программирования, упрощающий написание модулей C/C++ для Python. Данный выбор обусловлен следующими факторами:

1. Python – гибкий язык программирования, позволяющий работать с объектно-ориентированным подходом.
2. Язык Cython – компилируемый, он может быть настроен на работу почти со скоростью C простым добавлением нескольких статических объявлений типов, без необходимости в сложном коде интерфейса.
3. Скорость кодирования и читабельность Cython остаются сравнимы с Python.
4. Код Cython преобразуется в C/C++ код для дальнейшего расширения программы на языке Python.

Для реализации приложения была выбрана среда разработки Visual Studio Code. Данный выбор обусловлен следующими особенностями:

1. Данная платформа является бесплатной.
2. Грамотно реализованная работа с директориями и файлами позволяет удобно отлаживать проект.
3. Поддержка синтаксиса Python и Cython.

Для реализации пользовательского интерфейса была выбрана библиотека Tkinter.

3.2 Структура и состав классов.

На рисунке 3.1 представлена структура классов программы. Класс Window отвечает за визуальную составляющую программы и существует для взаимодействия между пользователем и программным обеспечением. Класс ImageModel является абстрактным и одержит в себе атрибуты и методы для работы с изображением. Класс NLM наследуется от базового класса ImageModel и содержит методы для обработки изображения нелокальным фильтром.

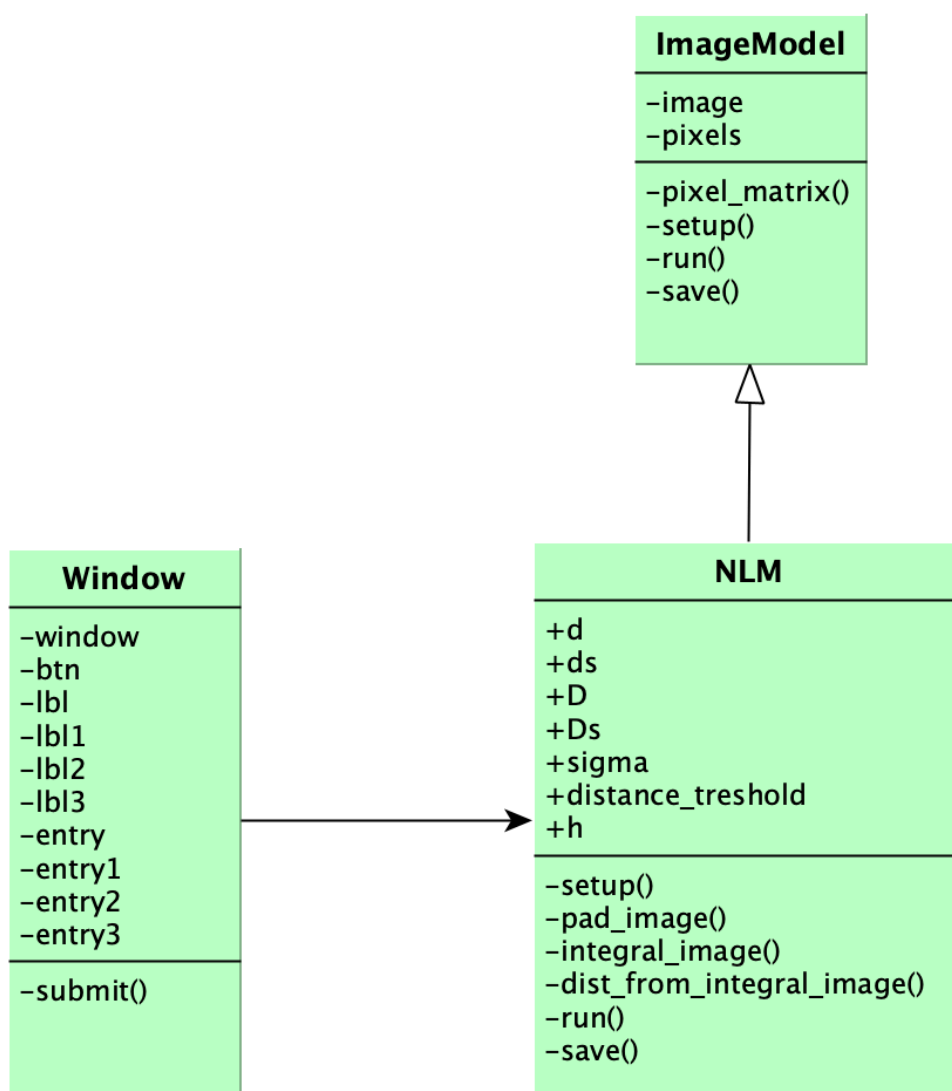


Рисунок 3.1. Диаграмма классов.

3.3 Сведения о модулях программы.

- `__main__.py` – главный модуль программы, содержащий класс пользовательского интерфейса, а также запускающий алгоритм.
- `setup.py` – файл, позволяющий преобразовать Cython-код в код на языке C/C++. Данный файл выполняется перед вызовом основной программы.
- `run.sh` – скрипт, запускающий выполнение программы.
- `src/model.pyx` – файл, содержащий код реализацию основных классов на языке Cython/Python.
- `src/model.c` – файл с кодом, генерирующимся на языке C, благодаря Cython коду.
- `src/in` – директория, содержащая изображения, передаваемые программе.
- `src/out` – директория, содержащая результирующие изображения.

3.4 Описание пользовательского интерфейса.

На рисунке 3.2 приведен пользовательский интерфейс.

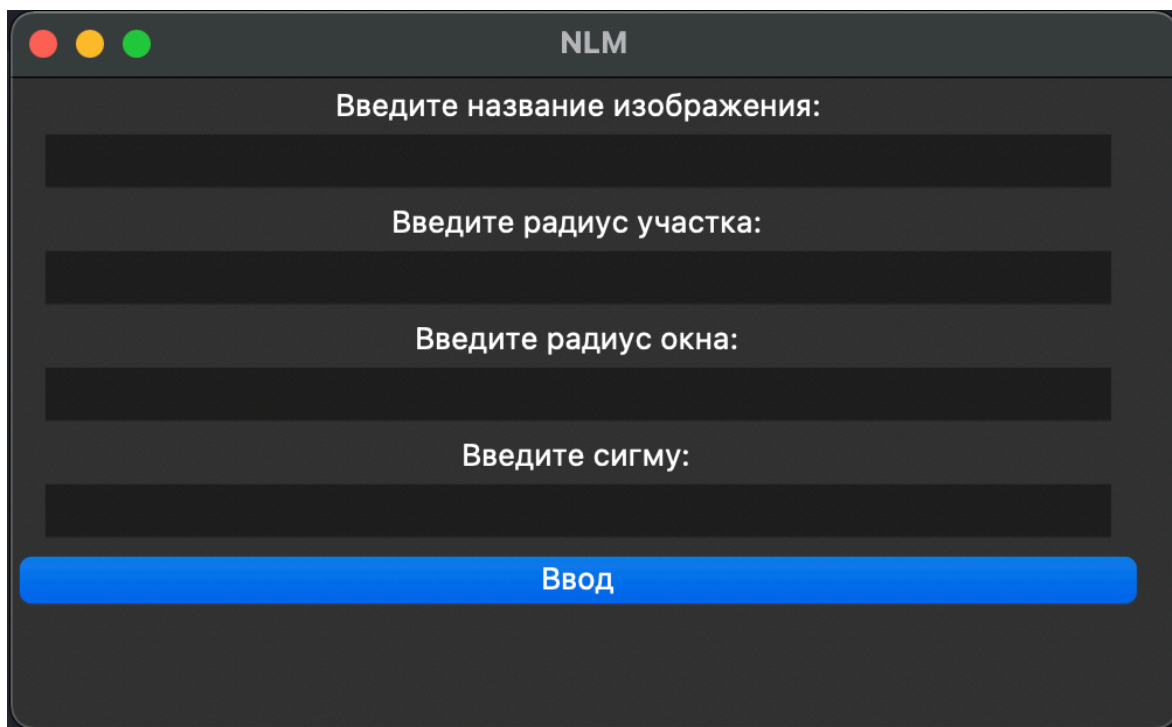


Рисунок 3.2. Пользовательский интерфейс.

На данном рисунке представлен минималистичный пользовательский интерфейс с полями ввода. Пользователю доступны параметры:

- Название изображения – изображение, с которым будет производиться работа.
- Радиус участка – параметр, отвечающий за величину участка вокруг пикселя, для которого будет считаться матрица весов, то есть определяться схожесть окна.
- Радиус окна – параметр, отвечающий за величину окна, с помощью которого будет высчитываться матрица весов.
- Сигма – определяет необходимую схожесть участков, то есть отвечает за степень сглаживания.

При нажатии кнопки “Ввод” начнется работа алгоритма. Итоговое изображение будет сохранено в рабочую папку.

4. Экспериментально - исследовательский раздел

В данном разделе будет поставлен эксперимент по определению оптимальных параметров работы программы и приведены примеры использования.

4.1 Технические характеристики

Экспериментальная часть курсовой работы проводилась на компьютере со следующими характеристиками:

- Операционная система: MacOS Big Sur
- Память: 8 GiB
- Процессор: Intel Core i5 1,4GHz

4.2 Цель эксперимента

Целью эксперимента является проверка правильности выполнения работы, оценка конечного результата при изменении различных параметров, а также нахождение оптимальных параметров.

4.3 Описание и выполнение эксперимента

Для проверки правильности выполнения поставленной задачи программой необходимо определиться с желаемым конечным результатом. В рамках данной курсовой работы – это изображение, в лучшей степени избавленное от шумов. Для более точного определения достижения цели приведем следующие критерии:

- 1) Изображение должно восприниматься как чистое, без шумов.
- 2) Изображение также не должно быть слишком сглаженным.
- 3) Потеря мелких деталей изображения не допустима.

При выполнении поставленных критериев, можно будет сформировать параметры, которые выставляются по умолчанию, чтобы конечный пользователь не был в замешательстве от некорректного результата.

На рисунке 4.1 представлено изображение, обработка которого будет производиться.

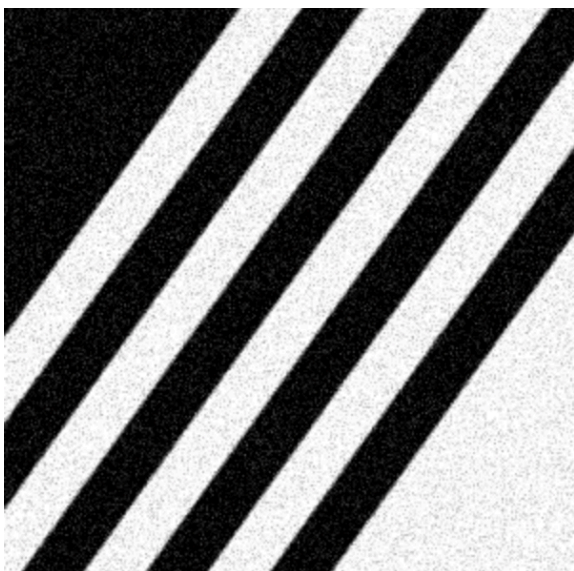


Рисунок 4.1. Исходное изображение.

На рисунке 4.2 можно увидеть первое обработанное изображение. Можно легко заметить, что большое количество шума осталось на изображении, из чего можно сделать вывод – параметры заданы неправильно.

На рисунке 4.3 представлено изображение, обработанное со следующими параметрами: 3-10-30. Данное изображение обработано приемлемо, границы полос не размыты, шума крайне мало.

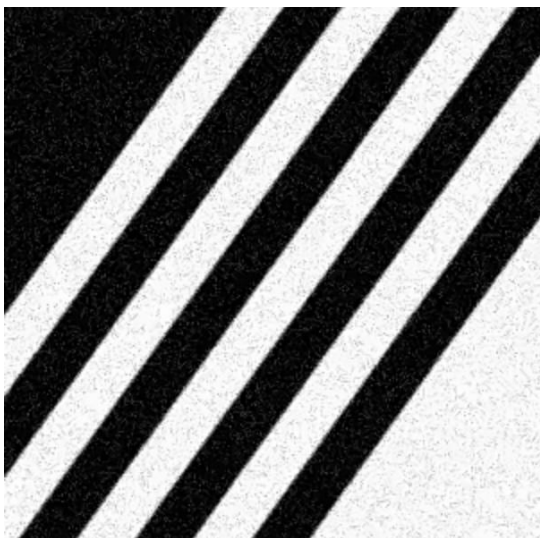


Рисунок 4.2. Параметры 3-10-5

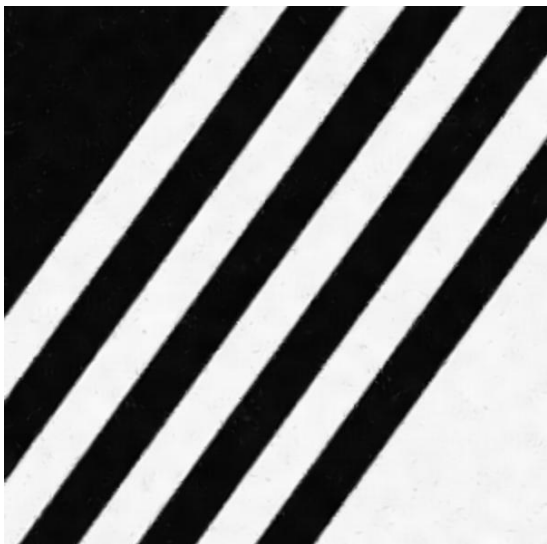


Рисунок 4.3. Параметры 3-10-30

Следующий рассматриваемый случай предоставлен на рисунке 4.4. Исходное изображение содержит много подобных элементов, также оно сильно зашумлено. При параметрах обработки 3-20-10 изображение не удовлетворяет поставленным требованиям, оно содержит артефакты, которые можно увидеть на изображении 4.5. На рисунке 4.6 представлено то же изображение, обработанное с параметрами 3-20-20. Данное изображение удовлетворяет поставленным требованиям.

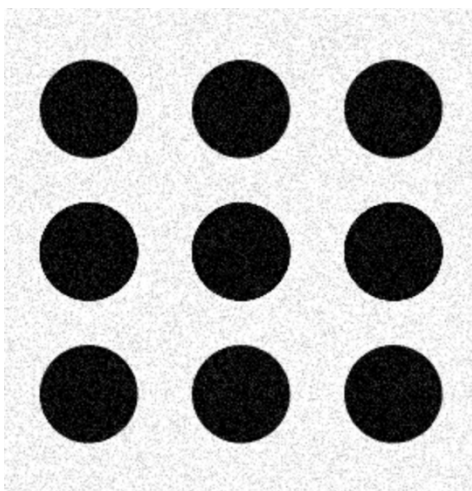


Рисунок 4.4. Исходное изображение

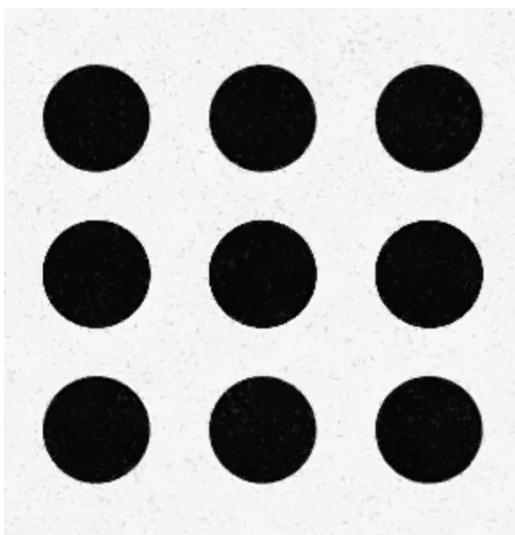


Рисунок 4.5. Изображение с артефактами.

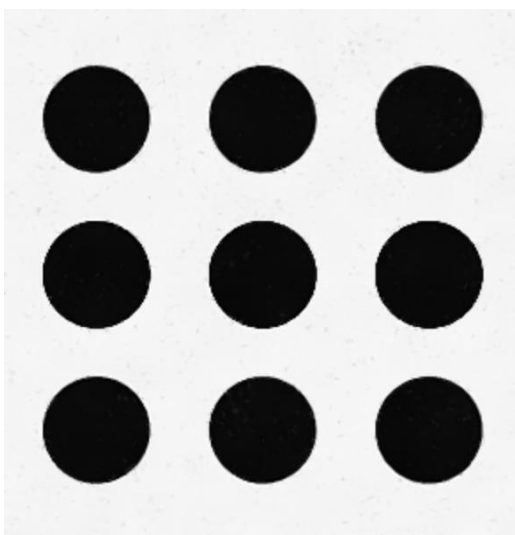


Рисунок 4.6. Итоговое изображение.

4.4 Вывод из результата исследований

Из проведенных исследований можно сделать вывод о том, что оптимальные параметры нельзя определить, как пограничные или усредненные значения. Каждое изображение, из – за своей специфики, требует ручного подбора параметров по личному усмотрению пользователя. Однако, полученные в ходе эксперимента значения установлены в программе по умолчанию.

5. Заключение

Во время выполнения поставленной задачи были рассмотрены и проанализированы основные алгоритмы сглаживания изображений, различные виды шумов, проблемы, которые возникают при их устранении. Были проанализированы достоинства и недостатки основных фильтров, использующихся в современном мире, выбран наиболее подходящий фильтр для поставленной задачи.

Реализованная в результате программа позволяет удалять шумы с различных изображений, давая компьютеру или человеку возможность для дальнейшей работы с ними.

В ходе выполнения поставленной задачи были получены обширные знания в области компьютерной графики.

Список использованной литературы

- [1] Цифровая обработка изображений [Электронный ресурс]. -Режим доступа:
https://ru.wikipedia.org/wiki/Цифровая_обработка_изображений
(Дата обращений 29.06.2020)
- [2] Шумопонижение [Электронный ресурс]. -Режим доступа:
<https://ru.wikipedia.org/wiki/Шумопонижение>
(Дата обращения 29.06.2020)
- [3] Линейная фильтрация изображений[Электронный ресурс]. -Режим доступа:
https://scask.ru/m_book_almg.php?id=27
(Дата обращения 29.06.2020)
- [4] Медианная фильтрация [Электронный ресурс]. -Режим доступа:
https://ru.bmstu.wiki/Медианная_фильтрация
(Дата обращения 29.06.2020)
- [5] Фильтрация изображений [Электронный ресурс]. -Режим доступа:
https://portal.tpu.ru/SHARED/a/AD/Education/Tab2/CG_filterring.pdf
(Дата обращения 29.06.2020)
- [6] Нелинейная фильтрация [Электронный ресурс]. -Режим доступа:
http://aco.ifmo.ru/el_books/image_processing/7_03.html
(Дата обращения 29.06.2020)
- [7] Адаптивная фильтрация [Электронный ресурс]. -Режим доступа:
https://ru.wikipedia.org/wiki/Адаптивный_фильтр
(Дата обращения 29.06.2020)
- [8] Основы машинной графики[Электронный ресурс]. -Режим доступа:
<http://automationlab.ru/index.php/2014-08-25-13-20-03/439-14--->
(Дата обращения 29.06.2020)