



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
K ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

на тему:

*«Метод восстановления дефокусированных
изображений на основе определенных параметров
искажения»*

Студент ИУ7-86Б
(Группа)

(Подпись, дата)

П. Ю. Сироткина
(И. О. Фамилия)

Руководитель ВКР

(Подпись, дата)

М. В. Филиппов
(И. О. Фамилия)

Нормоконтролер

(Подпись, дата)

Д. Ю. Мальцева
(И. О. Фамилия)

2023 г.

РЕФЕРАТ

Расчетно-пояснительная записка 71 с., 41 рис., 3 табл., 30 источн., 2 прил.

Ключевые слова: цифровое изображение, преобразование Фурье, искажение, дефокусировка, размытие, функция рассеяния точки, конволюция, «слепая» деконволюция, спектр, кепстр.

Объектом исследования является цифровое изображение, искаженное дефокусировкой фотокамеры.

Целью работы являлась разработка метода восстановления дефокусированных изображений на основе определенных параметров искажения.

Для достижения поставленной цели были решены следующие задачи:

- проведен анализ предметной области восстановления дефокусированных изображений;
- проведен сравнительный анализ методов восстановления без учета априорной информации об искажении;
- разработан метод восстановления дефокусированных изображений на основе определенных параметров искажения;
- спроектировано и реализовано программное обеспечение для реализации разрабатываемого метода;
- разработанный метод исследован на применимость при работе с различными типами дефокусированных изображений.

Разработанный метод имеет ряд ограничений, однако может быть применим даже в условиях отсутствия априорной информации об искажающем процессе.

Было выявлено, что зависимость времени обработки изображения от его размера и цветовой модели имеет линейный вид, а обработка цветного изображения в среднем требует примерно в 3 раза больше времени, чем обработка серого.

Была определена область применимости предложенного метода: в диапазоне радиуса дефокусировки от 10 до 20 единиц точность восстановления является удовлетворительный, а в случае обработки серого изображения этот

диапазон еще шире, поэтому если учет информации о цвете изображения некритичен, то рекомендуется использовать серое изображение вместо цветного в целях сокращения времени обработки и повышения качества результата.

В качестве перспектив дальнейшего развития реализованного программного обеспечения можно рассмотреть учет сложных моделей дефокусировки, возникающих в реальных условиях, применимость к видеофайлам, а также адаптацию предложенного решения под другие языки программирования в целях достижения независимости и кроссплатформенности.

СОДЕРЖАНИЕ

РЕФЕРАТ	5
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	9
ВВЕДЕНИЕ	10
1 Аналитический раздел	11
1.1 Терминология предметной области	11
1.1.1 Фотоаппарат	11
1.1.2 Цифровое изображение	12
1.1.3 Причины дефокусировки фотокамеры	13
1.1.4 Функция рассеяния точки	14
1.1.5 Основные цветовые модели	16
1.1.6 Основные форматы файлов изображения	18
1.1.7 Фурье – анализ	20
1.1.8 Шум	24
1.1.9 Свертка (конволюция)	25
1.2 Методы восстановления дефокусированных изображений	27
1.2.1 Методы классической деконволюции	27
1.2.2 Сравнительный анализ методов классической деконв- люции	32
1.2.3 Подходы к решению задачи «слепой» деконволюции	33
1.3 Методы оценки качества восстановления цифрового изображения	34
1.4 Постановка задачи	35
2 Конструкторский раздел	38
2.1 Требования и ограничения к разрабатываемому методу	38
2.2 Требования к разрабатываемому программному обеспечению	39
2.3 Основные этапы разрабатываемого метода	40
2.3.1 IDEF-0 диаграмма уровня A1	40
2.3.2 Схемы алгоритмов	40
2.3.3 Структура разрабатываемого программного обеспечения	46

3 Технологический раздел	47
3.1 Выбор средств реализации программного обеспечения	47
3.1.1 Выбор среды разработки	47
3.1.2 Используемые расширения	47
3.2 Формат входных и выходных данных	48
3.3 Детали реализации предлагаемого метода	48
3.4 Описание взаимодействия пользователя с программным обеспечением	51
3.5 Сборка и запуск проекта	52
4 Исследовательский раздел	54
4.1 Технические характеристики	54
4.2 Исследование времени работы предложенного метода в зависимости от размера изображения и цветовой модели	54
4.3 Исследование полноты решения поставленной задачи	56
4.4 Демонстрация вариантов работы метода в зависимости от точности вычислений	60
ЗАКЛЮЧЕНИЕ	64
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	65
ПРИЛОЖЕНИЕ А Полный код функции слепой деконволюции	68
ПРИЛОЖЕНИЕ Б	71

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящей расчетно-пояснительной записке применяют следующие сокращения и обозначения.

ФРТ — функция рассеяния точки (PSF, Point Spread Function).

Спектр — результат применения преобразования Фурье к сигналу.

Кепстр — результат применения преобразования Фурье к логарифму амплитудного спектра.

Конволюция — операция в функциональном анализе, которая при применении к двум функциям f и g возвращает третью функцию, соответствующую взаимокорреляционной функции $f(x)$ и $g(-x)$.

Деконволюция — операция, обратная конволюции.

Классическая деконволюция — деконволюция изображения на основе априорной информации об искажении.

Слепая деконволюция — деконволюция изображения без учета априорной информации об искажении.

PSNR (англ. Peak Signal-to-Noise Ratio) — пиковое отношение сигнал — шум.

ВВЕДЕНИЕ

Среди способов восприятия человеком информации об окружающем мире посредством органов чувств зрение занимает особое место — с помощью глаз в среднем воспринимается до 80 % информации [1], поступающей из внешней среды. Именно поэтому зрительные образы, часто запечатляемые снимками фотокамеры, играют важнейшую роль в жизни человека.

Системы фотосъемки могут быть использованы в различных сферах деятельности: криминалистика, кинематография, микроэлектроника, биомедицина, археология, исследования космоса, оборонное производство [2].

Однако часто полученный снимок фотокамеры оказывается искаженным за счет различных причин: шумы, турбулентность атмосферы, элементы интерференции и размытие, вызванное дефокусировкой, движением и нелинейностью пленки. Установлено, что дефокусировка и смаз — наиболее частые дефекты при съемке [3] .

Повторное получение изображений в указанных ранее сферах с целью устранения искажений часто является либо дорогостоящим, либо вовсе невозможным.

Большинство искажений являются частично или полностью обратимыми, таким образом, несмотря на распространенное заблуждение, потеря информации в исследуемом изображении может быть устранена. Например, дефокусировка является частично обратимым преобразованием: качество восстановления существенно зависит от наличия априорной информации о процессе искажения.

1 Аналитический раздел

В данном разделе рассмотрены основные термины предметной области восстановления дефокусированных изображений. Приведен обзор методов восстановления с учетом априорной информации об искажении, а также сравнительный анализ методов «слепой» деконволюции. Приведены методы оценки качества восстановления дефокусированных цифровых изображений.

1.1 Терминология предметной области

1.1.1 Фотоаппарат

Фотоаппарат — это прибор для получения на фотографическом материале действительного изображения предмета при фотографировании [4].

Цифровой фотоаппарат — фотоаппарат, в котором для регистрации изображения используется фотоэлектрический принцип.

Среди фотоаппаратов, выпускающихся в настоящее время, данный тип является значительно более распространенным (пленочными фотоаппаратами, в основе которых лежит химическая обработка фотоматериалов, пользуется менее 1% населения) [5], поэтому в данной работе будут рассматриваться изображения, полученные с помощью данного типа фотоаппаратов.

На рисунке 1.1 представлены основные элементы классического цифрового фотоаппарата.

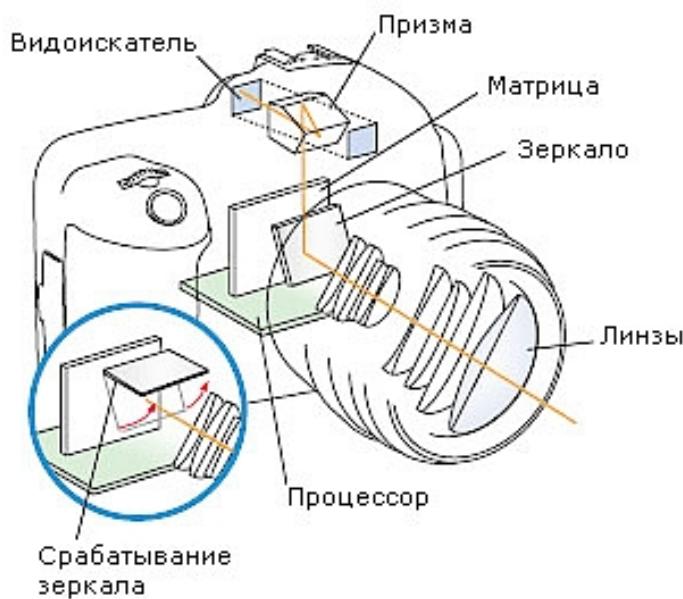


Рисунок 1.1 – Основные элементы цифрового фотоаппарата

Наиболее важным элементом с точки зрения рассмотрения вопроса дефокусировки является *объектив* фотоаппарата, представляющий собой оптическую систему, состоящую из совокупности центрированных линз и диафрагмы, заключенных в общую оправу. Свет, проходящий через объектив, попадает на зеркало, после чего, отразившись в призме через видоискатель, попадает в глаз.

Основными характеристиками объектива являются:

1. Фокусное расстояние — расстояние от оптического центра объектива до фокальной плоскости (матрицы), где происходит фокусировка света. Увеличение расстояния от предмета до линзы влечет за собой уменьшение расстояния от его изображения до линзы, и наоборот. Указанное условие обеспечивается фокусировкой объектива фотоаппарата перед съемкой [6].
2. Диафрагма — система подвижных лепестков, формирующих отверстие определенного размера, регулирующее поток света, поступающий на светочувствительную матрицу, которая в свою очередь регистрирует этот поток и преобразует его в электрический сигнал. Данные, полученные с матрицы, после обработки записываются на карту памяти фотоаппарата.

Каждый тип объектива имеет свои особенности и подходит для разных ситуаций съемки.

1.1.2 Цифровое изображение

Изображение, получаемое фотокамерой, можно определить как двумерную функцию $f(x, y)$, где x и y — пространственные координаты. Значение функции f в некоторой точке, задаваемой парой координат (x, y) , является положительной скалярной величиной, называемой интенсивностью, или яркостью (уровнем серого) изображения в этой точке [7] [8].

Для изображений, получаемых цифровой фотокамерой, величины x , y и f принимают конечное число дискретных значений. Такие изображения называются цифровыми.

1.1.3 Причины дефокусировки фотокамеры

На рисунке 1.2 приведена схема [9], иллюстрирующая физический принцип получения дефокусированного изображения.

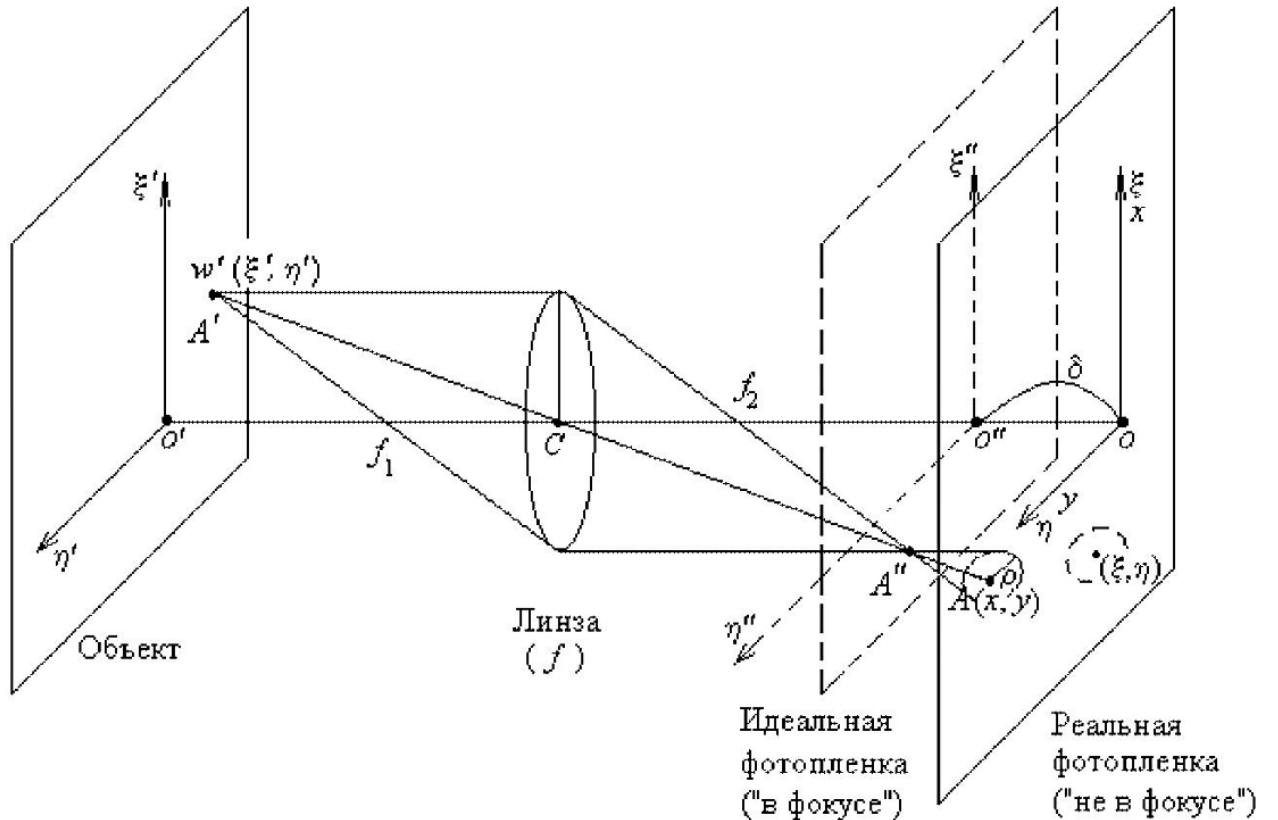


Рисунок 1.2 – Принцип получения дефокусированного изображения

Пусть снимаемый объект, полагаемый плоским из-за его удаленности, и фотопленка (или матрица сенсоров) расположены параллельно тонкой линзе по разные стороны от нее. Пусть δ – погрешность фокусировки изображения, f_1 – расстояние от линзы до объекта, f_2 – расстояние от линзы до фотопленки (матрицы), установленной в «фокусе» ($\delta = 0$).

Как видно из рисунка, лучи из точки A' после их прохождения через линзу отобразятся на реальной фотопленке не в точку, а в некоторое размытое пятно радиуса $\rho = \frac{ab}{f_2}$ с центром в точке $A(x, y)$, где a – радиус апертуры линзы. Данное пятно определяется т. н. функцией рассеяния точки, или функцией искажения ядра.

Дефокусированным изображением называется изображение, полученное на реальной пленке («не в фокусе») при $\delta \neq 0$.

1.1.4 Функция рассеяния точки

Функция рассеяния точки (англ. Point Spread Function) описывает, как оптическая система формирует изображение при наблюдении точечного источника или точечного объекта.

На рисунке 1.3 представлен пример формирования изображения.

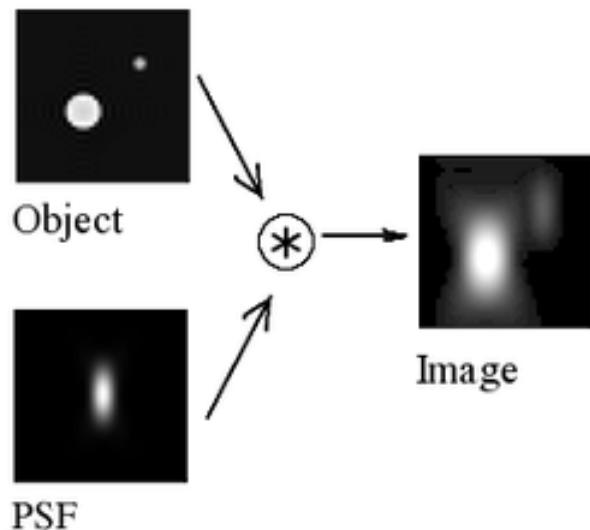


Рисунок 1.3 – Пример формирования изображения

Степень рассеяния является мерой качества системы формирования изображений. В зависимости от типа рассеивания ФРТ может быть приближена некоторыми распространенными математическими моделями, например гауссова функция или функция Эйри, также ФРТ может быть синтезирована на основе априорной информации о ее характере. Выбор конкретной модели функции рассеяния точки зависит от характеристик системы и требований к точности моделирования.

Опишем математически задачу дефокусировки. Рассмотрим помимо круга с центром $A(x, y)$ некоторый другой круг с центром в точке (ξ, η) . Радиусы этих (и других) кругов одинаковы и равны ρ , а их площади соответ-

ственно $\frac{\pi}{\rho^2}$. В результате некоторая интенсивность $w(\xi, \eta)$, соответствующая точке (ξ, η) , будет распределена по кругу радиуса ρ и площади $S = \frac{\pi}{\rho^2}$ с плотностью интенсивности $\frac{w(\xi, \eta)}{\pi\rho^2}$, постоянной в пределах круга в первом

приближении.

Интенсивность в точке A будет результатом суммирования (интегрирования) по всем тем кругам, которые накрывают точку $A(x, y)$. Условие накрытия точки кругом с центром в точке и радиуса ρ имеет следующий вид:

$$\sqrt{(x - \xi)^2 + (y - \eta)^2} \leq \rho. \quad (1.1)$$

В результате интенсивность в точке $A(x, y)$:

$$g(x, y) = \iint_{\sqrt{(x-\xi)^2+(y-\eta)^2} \leq \rho} \frac{w(\xi, \eta)}{\pi \rho^2} d\xi d\eta. \quad (1.2)$$

Соотношение (1.2) есть *двумерное интегральное уравнение I рода относительно* $w(\xi, \eta)$, однако оно записано в нестандартной форме. Приведем его к стандартной форме:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(x - \xi, y - \eta) w(\xi, \eta) d\xi d\eta = g(x, y), \quad -\infty < x, y < \infty, \quad (1.3)$$

где

$$k(x - \xi, y - \eta) = \begin{cases} \frac{1}{\pi \rho^2}, & \sqrt{(x - \xi)^2 + (y - \eta)^2} \leq \rho, \\ 0, & \text{иначе.} \end{cases} \quad (1.4)$$

Или

$$k(x, y) = \begin{cases} \frac{1}{\pi \rho^2}, & \sqrt{x^2 + y^2} \leq \rho, \\ 0, & \text{иначе.} \end{cases} \quad (1.5)$$

Соотношение (1.5) есть *двумерное интегральное уравнение Фредгольма I рода типа свертки*. Ядро интегрального уравнения $k(x, y)$ называется *функцией рассеяния точки*.

Таким образом, установлено, что в случае дефокусировки ФРТ имеет вид диска [10] радиуса ρ и интенсивностью внутри диска равной $\frac{1}{\pi \rho^2}$. Дальнейшая задача заключается в нахождении радиуса этого диска для последующего восстановления искажения.

1.1.5 Основные цветовые модели

Цифровая модель цифрового изображения — это математическая модель, используемая для представления цвета на компьютере или ином цифровом устройстве. Она определяет способ представления цвета в цифровой форме.

1. Модель *RGB* (англ. Red, Green, Blue — Красный, Зеленый, Синий) — аддитивная цветовая модель, использующая три основных цвета для кодирования цифрового изображения. Выбор основных цветов обусловлен особенностями восприятия цвета человеческим глазом.

Данная модель используется для создания цветов на электронных устройствах: телевизоры, компьютеры, планшеты, смартфоны и т.д.

Цветовое пространство модели можно представить в виде куба, координатными осями которого являются красный, синий и зеленый цвета. Диагональ куба даст градации серого. На рисунке 1.4 представлен пример такого куба.

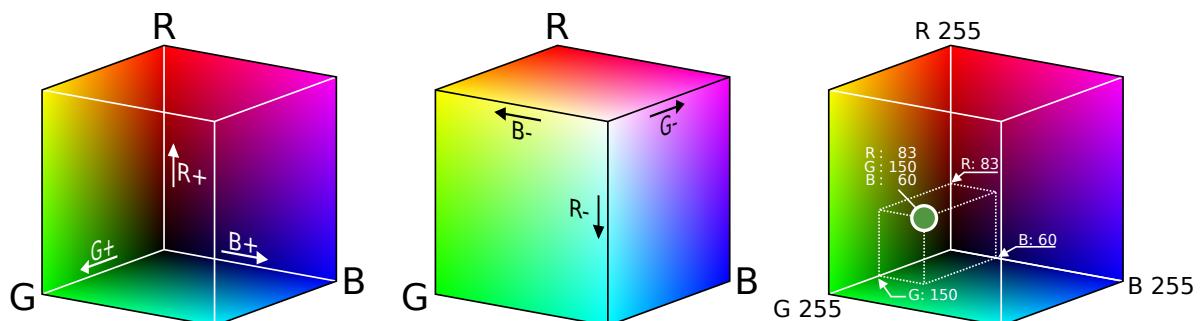


Рисунок 1.4 – Цветовое пространство RGB – модели

В современных системах каждая из этих компонент представлена в 255 градациях, что порождает около 16 млн различных оттенков. Однако, несмотря на размер цветового пространства, цветовой диапазон ограничен — RGB не может воспроизвести многие воспринимаемые человеческим глазом цвета (например, спектрально чистый голубой), что может стать проблемой при работе с профессиональными цифровыми изображениями.

Также недостатком данной модели является применимость только к линейным алгоритмам обработки. В противном случае данная модель не подходит.

2. Модель *CMYK* (англ. Cyan, Magenta, Yellow, blacK — Голубой, Пурпурный, Желтый, Черный) — субтрактивная модель, используемая в типографии.

Основные цвета модели образовались при смешивании основных цветов RGB модели:

- красный и зеленый цвета порождают желтый;
- красный и синий цвета порождают пурпурный;
- зеленый и синий порождают голубой.

Также в модели используется черный цвет в качестве дополнительного к первым трем с целью улучшения контрастности, глубины теней и экономии красок, т.к. смешивание основных цветов даст грязно – серый, но не черный цвет.

Цветовое пространство модели также можно представить в виде куба, координатными осями которого являются голубой, пурпурный и желтый цвета. На рисунке 1.5 представлен пример такого куба.

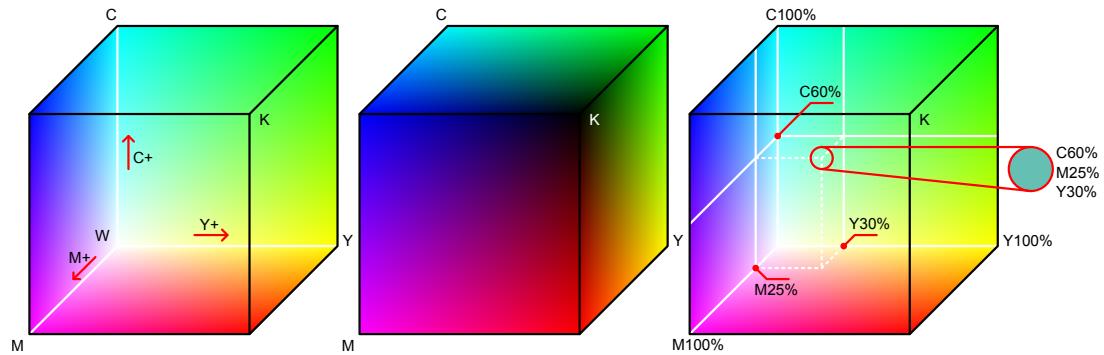


Рисунок 1.5 – Цветовое пространство CMYK – модели

3. Модель *HSL* (англ. Hue, Saturation, Lightness — Тон, Насыщенность, Светлота) — модель, использующая для определения цвета три параметра: тон, насыщенность и светлоту.

Эта модель часто используется в области обработки изображений и визуализации данных. Идея заключается в том, что человеческий глаз в большей степени воспринимает изменение интенсивности и в меньшей — изменение цвета, как в RGB.

Цветовое пространство данной модели можно представить в виде двух симметричных конусов, расположенных на одной оси. На рисунке 1.6 представлен пример таких конусов. На оси конусов отсчитывается интенсивность от черного до белого. По окружности отсчитывается тон. Вдоль радиуса откладывается насыщенность.

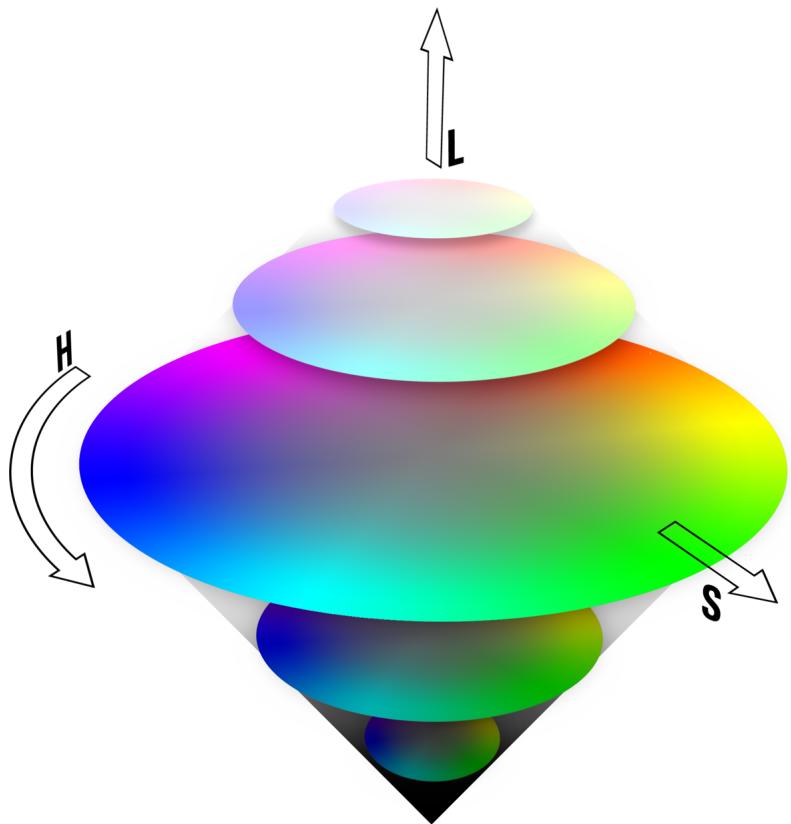


Рисунок 1.6 – Цветовое пространство HSL – модели

1.1.6 Основные форматы файлов изображения

Формат файла изображения определяет способ хранения и передачи информации. Как известно, цифровое изображение, полученное с помощью фотоаппарата, не может быть векторным. Ниже приведены наиболее распространенные форматы файлов растрового изображения:

- **BMP** (англ. Bitmap — битовая карта) — формат хранения однослойных растровых изображений. Данный способ записи не сжимает исходную информацию.
- **TIFF** (англ. Tagged Image File Format — формат файла изображения с тегами) — формат хранения высококачественных растровых изображений.

Такие файлы могут содержать множество слоев, включая изображения, метаданные, цветовые профили и другую информацию, что делает их полезными для профессиональных приложений в графическом дизайне, фотографии и печати.

- **JPEG** (англ. Joint Photographic Experts Group — объединенная группа экспертов по фотографии) — использует метод сжатия с потерями, однако размер таких файлов существенно меньше, поэтому данный формат может быть предпочтителен в случае терпимости к потерям.
- **PNG** (англ. Portable Network Graphics — портативная сетевая графика) — распространенный формат хранения и передачи растровых изображений, использующий использует алгоритм сжатия без потерь, что означает, что изображение сохраняется точно в том же качестве, что и оригинал, без потери деталей или цветовой точности. Также этот формат предоставляет возможности для работы с прозрачными изображениями и характеризуется малым размером файла в результате сжатия.

Каждый формат имеет свои преимущества и недостатки. Выбор формата зависит от конкретных потребностей и требований, предъявленных к поставленной задаче.

1.1.7 Фурье – анализ

Идея, лежащая в основе Фурье – анализа, заключается в том, что периодическая функция $f(t)$ непрерывной переменной t , имеющая период τ , может быть представлена в виде взвешанной суммы косинусов и синусов. Данная сумма, известная как *ряд Фурье*, имеет следующий вид:

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n \cdot e^{in\frac{2\pi}{\tau}t}, \quad (1.6)$$

где $i = \sqrt{-1}$, а коэффициенты c_n равны:

$$c(n) = \frac{1}{\tau} \cdot \int_{-\frac{\tau}{2}}^{\frac{\tau}{2}} f(t) \cdot e^{in\frac{2\pi}{\tau}t} dt, n = 0, \pm 1, \pm 2, \dots \quad (1.7)$$

Тот факт, что выражение 1.6 есть разложение на синусы и косинусы, следует из формулы Эйлера:

$$e^{i\theta} = \cos \theta + i \cdot \sin \theta, \quad (1.8)$$

где e — число Эйлера ($2,71828\dots$).

Преобразование Фурье — операция, сопоставляющая одной функции вещественной переменной другую функцию вещественной переменной, которая описывает коэффициенты при разложении исходной функции на гармонические колебания (элементарные составляющие) с разными частотами.

Пусть $f(x)$ — одномерная произвольная функция. Преобразование Фурье для этой функции определяется выражением (1.9):

$$F(u) = \frac{1}{2\pi} \cdot \int_{-\infty}^{+\infty} f(x) \cdot e^{-ixu} dx, \quad (1.9)$$

где u — частотная переменная, $F(v)$ — Фурье – образ, который в общем случае является комплексным.

Обратное преобразование Фурье имеет следующий вид:

$$f(x) = \frac{1}{2\pi} \cdot \int_{-\infty}^{+\infty} F(u) \cdot e^{ixv} du. \quad (1.10)$$

Выражения 1.9 и 1.10 составляют т.н. Фурье-пару.

На рисунке 1.7 представлена связь между временной (или пространственной) и частотной областью вычислений.

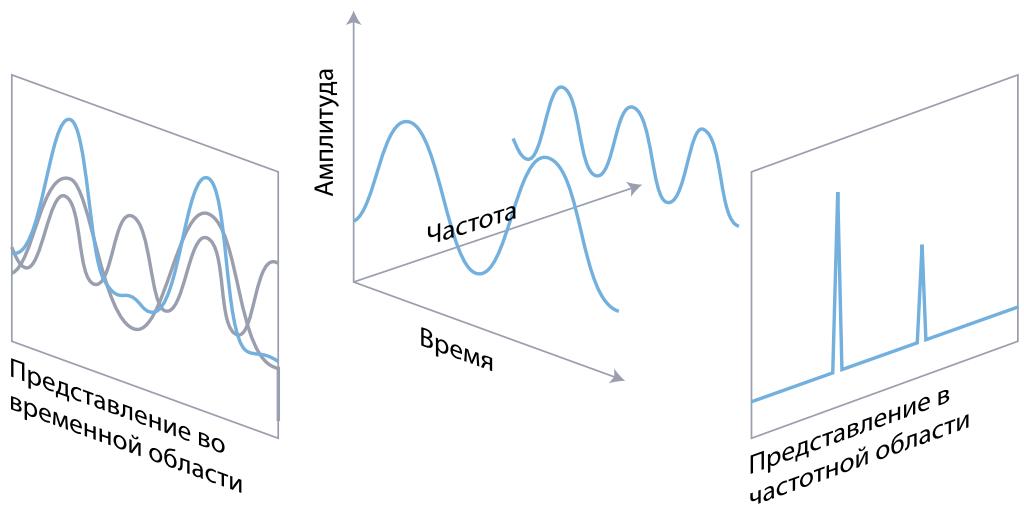


Рисунок 1.7 – Связь между временной и пространственной областью вычислений

Таким образом, для описания полного преобразования достаточно знать один полный период. Из этого следует, что обратным преобразованием Фурье можно восстановить $f(t)$ из единственного периода.

Эффективность применения рассматриваемого математического аппарата заключается в том, что функция, заданная преобразованием Фурье (или его рядом), *может быть* полностью, восстановлена при помощи некоторой процедуры обращения. Таким образом, можно работать в т.н. Фурье — области, а затем вернуться в исходную область определения функции.

Т.к. цифровая обработка сигналов основана на дискретизации и квантовании аналоговых сигналов, на практике необходимо обрабатывать конечное число отсчетов, вследствие чего было разработано аналогичное преобразование для дискретных сигналов — дискретное преобразование Фурье, позволяющее восстановить непрерывный периодический сигнал.

Прямое дискретное преобразование Фурье функции $f(x, y)$ цифрового изображения размерами $M \times N$ имеет следующий вид:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-2\pi i (\frac{ux}{M} + \frac{vy}{N})}, \quad (1.11)$$

Обратное ДПФ изображения $f(x, y)$ имеет следующий вид:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot e^{-2\pi i (\frac{ux}{M} + \frac{vy}{N})}. \quad (1.12)$$

Выражения 1.11 и 1.12 составляют пару ДПФ.

Анализ сигналов, основанный на преобразовании сигнала из временной области в частотную область, называется *спектральным* [11]. Представление изображения в таком пространстве дает возможность наблюдать его структурные особенности, связанные с периодичностью повторения элементов, наличием мелких деталей и др.

На рисунках 1.8 и 1.9 представлены исходное изображение [12] и его спектр, являющийся результатом преобразования Фурье.



Рисунок 1.8 – Исходное изображение

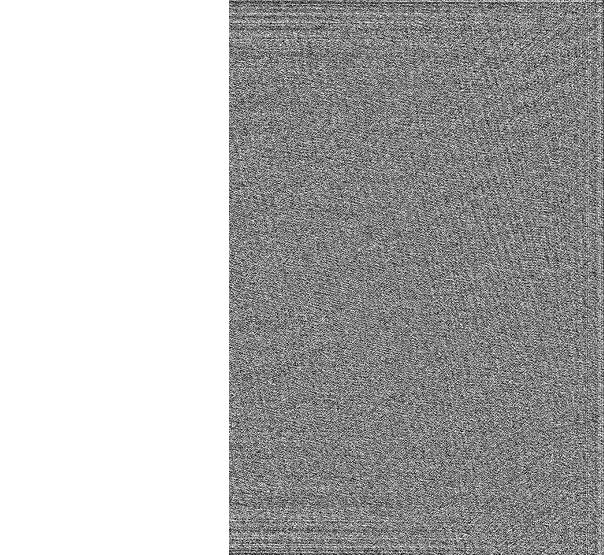


Рисунок 1.9 – Спектр изображения

Следует отметить, что в общем виде спектр сигнала является комплексным, однако в целях улучшения визуализации характерных особенностей сигнала рассматривают его амплитудный спектр, соответствующий реальной части комплексного числа, и фазовый, соответствующий мнимой части.

Выбор между анализом амплитудного или фазового спектра зависит от требований и ограничений в рамках решаемой задачи, однако в большинстве случаев амплитудный спектр является более распространенным инструментом анализа характеристик сигнала, т.к. он позволяют определить, какие частоты присутствуют и какова их амплитуда, а также более прост в интерпретации.

На рисунках 1.10 и 1.11 представлены амплитудный и фазовый спектры исходного изображения. Здесь и далее спектры смещены в центр. В центре амплитудного спектра присутствует одна видимая белая точка.

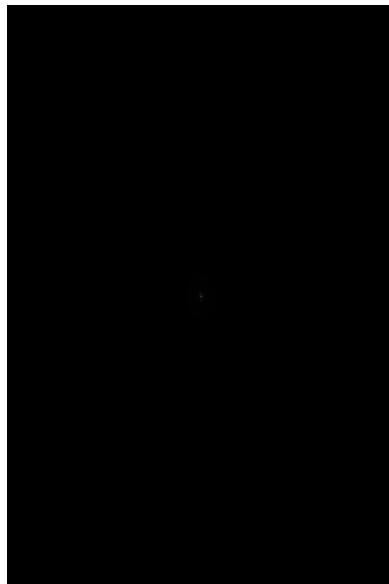


Рисунок 1.10 – Амплитудный спектр

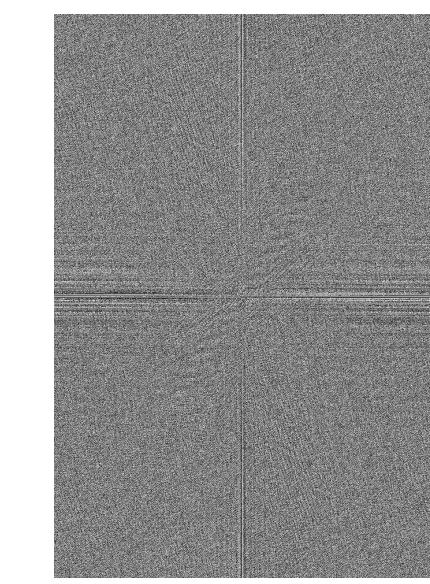


Рисунок 1.11 – Фазовый спектр

Размерности амплитудного и фазового спектра всегда совпадают с размерностью исходного изображения.

Также существует *кепстральный* анализ цифровых сигналов, основанный на применении преобразования Фурье к логарифму амплитудного спектра.

Данный подход позволяет извлечь полезную информацию о спектральных характеристиках сигналов и обеспечивает новые возможности для анализа и обработки, которые не всегда могут быть доступны при использовании рассмотренного выше спектрального анализа.

Кепстр [13] [14] изображения может быть представлен с помощью следующего выражения:

$$C_s(u) = \ln |F(u)^2|, \quad (1.13)$$

где $F(u)$ — спектр сигнала.

На рисунке 1.12 представлено исходное дефокусированное изображение.

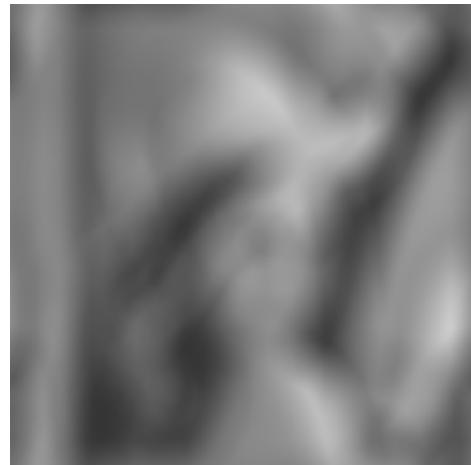


Рисунок 1.12 – Дефокусированное изображение

На рисунках 1.13 и 1.14 представлены амплитудные спектр и кепстр соответственно для искаженного изображения.

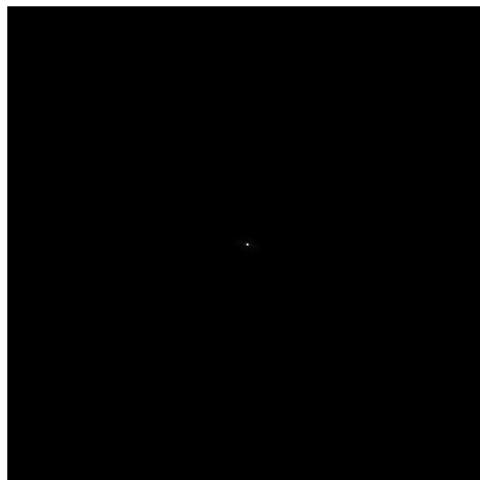


Рисунок 1.13 – Амплитудный спектр

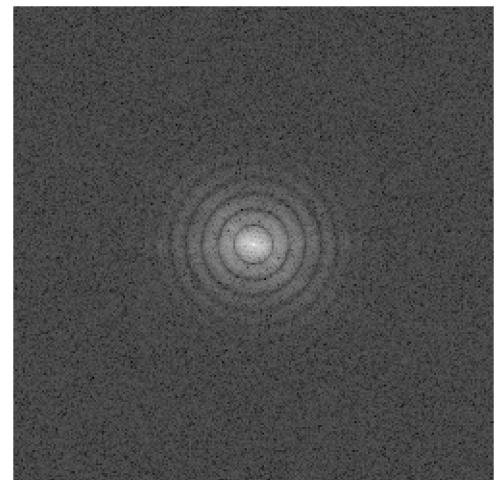


Рисунок 1.14 – Кепстр

Как можно заметить, если искажение сигнала вызвано дефокусировкой, то спектр сигнала также имеет яркую точку в центре, а кепстр будет содержать концентрические кольца [15].

1.1.8 Шум

Шум в цифровом сигнале представляет собой нежелательные случайные изменения полезного сигнала, которые могут вносить искажения и, следствен-но, ухудшать его качество. Эта помеха может существенно влиять на процесс обнаружения и извлечения значимой информации из сигнала, в частности, изображения.

Основными источниками шума на цифровом изображении являются сам процесс его получения (оцифровки), а также процесс передачи. Шум может быть как и аддитивным (не корректирующим с изображением и не зависящим от координат пикселя), так и мультипликативным (изменяющим форму сигнала).

Выделяют следующие типы шума:

- электронный шум, полученный вследствие теплового движения электронов в электронных частях снимающих систем;
- фотоэлектрический шум, возникающий в результате статистической природы света;
- шум, возникающий из-за зернистости фотопленки;
- дискретный шум, появляющийся при дискретизации изображения.

Для борьбы с шумом могут применяться различные методы фильтрации, такие как: медианный фильтр, фильтр среднего значения, фильтр Каламана и др. Одним из важных аспектов при обработке сигнала является достижение баланса между уровнем шума и сохранением полезной информации о сигнале.

1.1.9 Свертка (конволюция)

Применительно к обработке цифровых изображений операция свертки может быть интерпретирована следующим образом: на основе некоторого множества пикселей исходного изображения вычисляется новый пиксель результирующего (искаженного ядром свертки) изображения.

В зависимости от выбранного ядра свертки, применяемого к изображению, можно получить тот или иной эффект: размытость, повышение резкости, обнаружение контуров, граничное обнаружение и т. д.

Математически операцию двумерной свертки цифрового изображения размером $M \times N$ в пространственной области можно описать в виде выражения (1.14) [16]:

$$f(x, y) \oplus h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \cdot h(x - m, y - n). \quad (1.14)$$

На рисунке 1.15 [17] представлен пример выполнения операции свертки. Для вычисления новых значений используется т.н. ядро свертки. На представленном примере ядром является матрица серого цвета размером 3×3 ячейки.

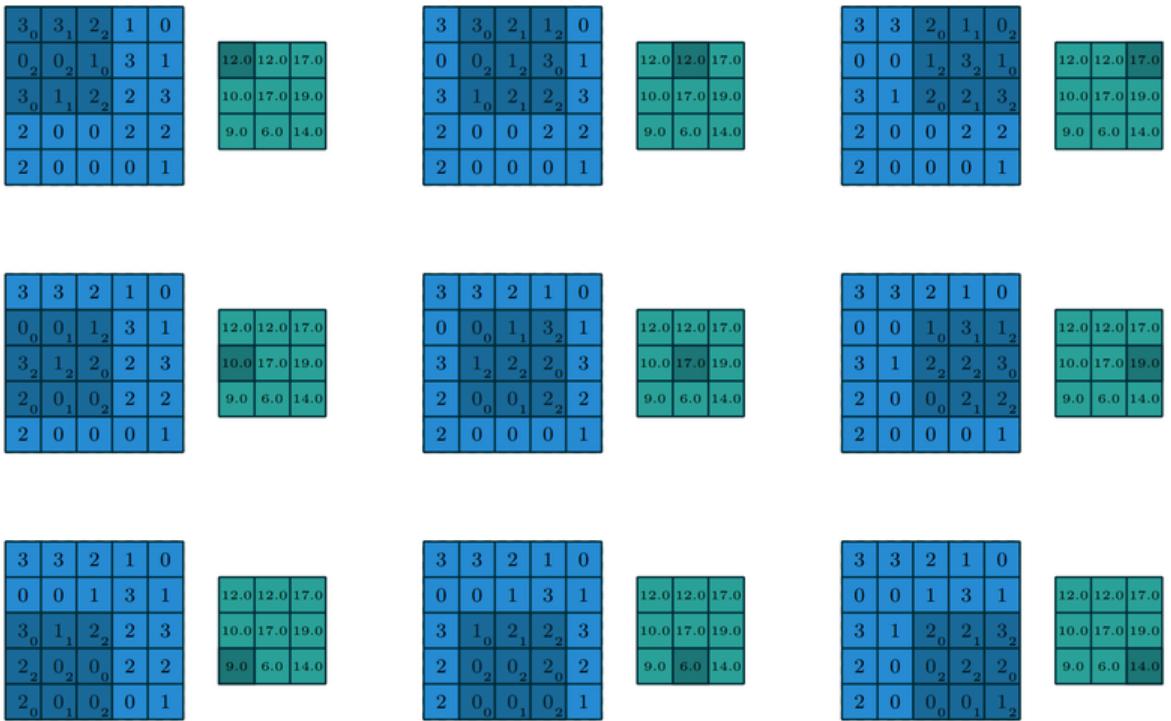


Рисунок 1.15 – Пример конволюции цифрового изображения

Вычисление функций $f(x, y)$ и $h(x, y)$ из выражения 1.14 путем выполнения обратных действий приводит к получению большой системы уравнений, решение которой является нетривиальной и трудоемкой задачей. Упростить ее решение может *теорема о свертке*, согласно которой операция свертки в пространственной области эквивалентна поэлементному умножению в частотной области:

$$f(x, y) \oplus h(x, y) \longleftrightarrow F(u, v) \cdot H(u, v), \quad (1.15)$$

где $F(u, v)$, $H(u, v)$ – Фурье – образы (спектры) функций $f(x, y)$ и $h(x, y)$ соответственно.

1.2 Методы восстановления дефокусированных изображений

Модель процесса получения дефокусированного цифрового изображения в пространственной области может быть представлена в виде выражения (1.16) [18]:

$$g(x, y) = f(x, y) \oplus h(x, y) + \eta(x, y), \quad (1.16)$$

где:

1. $f(x, y)$ — функция, описывающая исходное (неискаженное) изображение.
2. $g(x, y)$ — функция, описывающая дефокусированное (искаженное) изображение.
3. $h(x, y)$ — функция размытия точки (в общем случае функция импульсного отклика).
4. $\eta(x, y)$ — функция шума.
5. Символ « \oplus » — оператор свертки.

Задача восстановления изображения заключается в поиске наилучшего приближения $\hat{f}(x, y)$ исходного изображения. Методы, решающие поставленную задачу, делятся на два класса в зависимости от наличия априорной информации о параметрах искажения.

1.2.1 Методы классической деконволюции

Инверсный фильтр

Одним из самых простых методов решения задачи деконволюции является инверсная фильтрация.

Согласно теореме о свертке, имеем (1.17):

$$G(u, v) = F(u, v) \cdot H(u, v) + N(u, v). \quad (1.17)$$

Было предложено разделить обе части выражения на $H(u, v)$ и получить следующую оценку исходного изображения:

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} = F(u, v) + \frac{N(u, v)}{H(u, v)}. \quad (1.18)$$

Если на изображении отсутствует шум, то восстановление происходит достаточно точно. Однако, если шум присутствует, то составляющая $\frac{N(u, v)}{H(u, v)}$ стремится к бесконечности ввиду того, что в частотной области $H(u, v)$ стремится к нулю, что приводит к получению некачественного результата [19]. Даже при малом шуме, который практически не заметен на изображении, результат содержит существенные помехи.

Таким образом, данный метод практически никогда не применяется в реальных условиях.

Фильтр Винера

Усовершенствованием инверсной фильтрации можно считать фильтр Винера [20]. Данный метод, в отличие от предыдущего, учитывает информацию о шуме на изображении.

Метод базируется на рассмотрении функций изображения и шума как случайных процессов и нахождении такой оценки $\hat{F}(u, v)$ для неискаженного изображения $f(x, y)$, чтобы среднеквадратическое отклонение этих величин было минимальным:

$$\hat{F}(u, v) = \frac{1}{H(u, v)} \cdot \frac{|H(u, v)|^2}{|H(u, v)|^2 + \frac{S_\eta(u, v)}{S_f(u, v)}} \cdot G(u, v). \quad (1.19)$$

Функциями S_η и S_f обозначают энергетические спектры шума и исходного изображения соответственно. Т. к. эти величины обычно неизвестны, то их заменяют на некоторую константу K , которую можно охарактеризовать как приблизительное соотношение сигнал – шум:

$$\hat{F}(u, v) = \frac{1}{H(u, v)} \cdot \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \cdot G(u, v). \quad (1.20)$$

На рисунках 1.16 и 1.17 представлен пример восстановления дефокусированного изображения с наличием шумовой составляющей.



Рисунок 1.16 – Размытое и зашумленное изображение



Рисунок 1.17 – Результат восстановления фильтром Винера

Если шум на изображении отсутствует, то фильтр Винера сводится к инверсной фильтрации. Фильтр Винера эффективен для снижения шума и искажений, особенно когда статистические характеристики шума и искажений известны или могут быть оценены. Однако он предполагает линейность и стационарность сигнала и шума, что может быть ограничением в некоторых реальных ситуациях.

Важно отметить, что фильтр Винера может быть применен не только для восстановления изображений, но и для других типов сигналов, таких как аудио или временные ряды, где присутствуют шумы и искажения.

Регуляризация по Тихонову

Метод также реализовывается в частотной области. Этот метод также называют методом минимизации сглаживающего функционала со связью, или методом наименьших квадратов со связью.

Идея заключается в формулировке задачи в матричном виде с дальнейшим решением соответствующей задачи оптимизации. Решение задачи имеет следующий вид:

$$\hat{F}(u, v) = \frac{H'(u, v)}{|H(u, v)|^2 + \gamma|P(u, v)|^2} \cdot G(u, v), \quad (1.21)$$

где γ — параметр регуляризации, $P(u, v)$ — результат Фурье – преобразования оператора Лапласа, $H'(u, v)$ — функция, комплексно сопряженная $H(u, v)$.

На рисунках 1.18 и 1.19 представлен пример восстановления дефокусированного изображения с наличием шумовой составляющей.



Рисунок 1.18 – Искаженное изображение



Рисунок 1.19 – Результат восстановления

Важным является вопрос выбора параметра регуляризации γ , сводящего поставленную задачу восстановления изображения к корректной. Если параметр $\gamma = 0$, то данный метод сводится к инверсной фильтрации.

Метод регуляризации Тихонова применяется для борьбы с проблемой неоднозначности восстановления и ограничениям, связанным с обратной задачей. Этот метод предполагает минимизацию суммы квадратов ошибок и квадратов регуляризационных членов в целевой функции.

Одним из преимуществ регуляризации Тихонова является возможность контролировать компромисс между точностью восстановления и сглаживанием. Параметр регуляризации, также называемый коэффициентом регуляризации, позволяет настраивать важность регуляризационного члена относительно суммы квадратов ошибок.

Если параметр регуляризации слишком большой, то сглаживание будет слишком сильным и могут быть потеряны важные детали. С другой стороны, если параметр регуляризации слишком мал, то артефакты могут сохраняться в восстановленном результате.

Все рассмотренные выше методы являются линейными и не итерационными, в следствие чего обладают одним общим недостатком — чувствительностью к определению параметров искажения, т.е. в случае небольшого расхождения оценки искажающей функции качество восстановленного изображения резко ухудшается. Этому недостатку менее подвержен итерационный пространственный алгоритм Люси – Ричардсона.

Метод Люси — Ричардсона

Этот метод, в отличие от рассмотренных ранее, является итерационным, нелинейным и реализовывается в пространственной области.

Идея заключается в использовании метода максимального правдоподобия, для которого предполагается, что яркость изображения подчиняется распределению Пуассона.

Математическое выражение для данного метода имеет следующий вид:

$$\hat{f}_{k+1}(x, y) = \hat{f}_k(x, y) \cdot \left[h(-x, -y) \oplus \frac{g(x, y)}{h(x, y) \oplus \hat{f}(x, y)} \right], \quad (1.22)$$

где $\hat{f}_{k+1}(x, y)$ — оценка изображения f на k -ом шаге вычислений.

Т.к. обработка происходит в пространственной области, то нет необходимости в использовании преобразований Фурье, что позволяет снизить вычислительную сложность алгоритма.

На рисунках 1.20 и 1.21 представлены искаженное (размытое и зашумленное) изображение и результат восстановления методом Люси — Ричардсона соответственно.



Рисунок 1.20 – Искаженное изображение



Рисунок 1.21 – Результат восстановления

Недостатком метода является возникновение краевых артефактов в виде горизонтальных и вертикальных полос на изображении. Также возникает вопрос об оптимальном количестве операций (выбор критерия остановки итерационного алгоритма).

1.2.2 Сравнительный анализ методов классической деконволюции

В рамках рассматриваемой задачи для метода классической деконволюции рассматриваются следующие критерии:

- область вычислений — пространственная или частотная;
- устойчивость к шуму на изображении;
- необходимость пост – или предобработки;
- сложность вычислений.

Обозначим данные критерии как K_1 , K_2 , K_3 и K_4 соответственно.

В таблице 1.1 представлен сравнительный анализ методов классической деконволюции.

Таблица 1.1 – Сравнительный анализ методов классической деконволюции

Метод	K_1	K_2	K_3	K_4
<i>Инверсная фильтрация</i>	Частотная	Низкая	Нет	Низкая
<i>Фильтр Винера</i>	Частотная	Низкая	Нет	Низкая
<i>Метод Люси – Ричардсона</i>	Пространственная	Средняя	Да	Средняя
<i>Регуляризация</i>	Частотная	Высокая	Да	Высокая

Инверсный фильтр практически никогда не применяется в обработке реальных изображений, т.к. не учитывает наличие шума. Фильтр Винера является улучшением метода инверсной фильтрации, однако также устойчивость к шуму остается неудовлетворительной в реальных задачах, т.к. предполагает наличие информации о модели шума.

Метод Люси – Ричардсона является итерационным, в связи с чем имеет относительно высокую вычислительную сложность, однако является более устойчивым по отношению к шумовому компоненту.

Регуляризация Тихонова является наиболее стабильным методом, однако требует существенной нетривиальной предобработки изображения и имеет высокую вычислительную сложность.

1.2.3 Подходы к решению задачи «слепой» деконволовции

На практике информация об искажении практически никогда не бывает известна заранее, поэтому в данной работе будет разработан метод, учитывающий отсутствие любой априорной информации. Такая задача порождает целое семейство методов, которые называются методами «слепой» деконволюции.

В основе реализации методов «слепой» деконволюции лежат методы классической деконволюции, предполагающие наличие информации об искажении, базирующихся на методе максимального правдоподобия, где целевой функцией является исходное (неискаженное) изображение.

Как правило, метод состоит из трех этапов:

1. Ядро искажения оценивается по входному изображению.
2. Используя оценочное ядро, применяется стандартный алгоритм деконволюции для оценки скрытого изображения.
3. Оценка качества восстановленного сигнала.

Методы слепой деконволюции могут использовать различные подходы, такие как статистическая оценка, регуляризация или использование априорной информации о свойствах исходного сигнала.

Существует два основных подхода к решению задачи вычисления «слепой» деконволюции [21]:

1. *Определение ФРТ отдельно от восстанавливаемого изображения.* Полученная информация используется после применения одного из известных классических методов восстановления. Оценка функции импульсного отклика и восстановление изображения — это отдельные процедуры для данного подхода. Алгоритмы, используемые для реализации данного метода, являются вычислительно простыми.

В таких случаях обычно матрица, соответствующая функции рассеяния точки, может быть инициализирована некоторыми стандартными значениями (часто выбирается значение $\frac{1}{N}$, где N — размерность квадратной матрицы ФРТ).

2. *Включение идентификационной процедуры ФРТ в восстанавливающий алгоритм.* Этот подход подразумевает одновременную оценку функции импульсного отклика и восстанавливаемого изображения, что приводит к более сложным вычислительным алгоритмам.

Целью задачи слепой деконволюции является получение функции исходного неискаженного изображения $f(x, y)$, и, соответственно, искажающей функции $h(x, y)$, которая была применена к изображению, зная только функцию искаженного изображения $g(x, y)$.

1.3 Методы оценки качества восстановления цифрового изображения

Существует несколько методов оценки качества восстановленного дефокусированного изображения:

- *Пиковое отношение сигнал – шум* (англ. PSNR — Peak Signal-to-Noise Ratio) — скалярная метрика, обозначающая соотношение между максимальной мощностью сигнала и шума. Для вычисления PSNR чаще всего используется среднеквадратичная ошибка (англ. Mean Square Error). Метрика выражается в логарифмической шкале и измеряется в децибелах. Чем меньше значение этой метрики, тем хуже качество восстановления.

Является основным количественным критерием для оценки эффективности восстановления сигналов в виду, однако имеет ряд ограничений, основным из которых является тот факт, что PSNR (и MSE в частности) слабо коррелируют с физиологией человеческого восприятия.

- *Индекс SSIM* (англ. Structure Similarity Index) характеризует структурное сходство между оригинальным и восстановленным сигналами,

принимает значение в диапазоне от -1 до 1, где 1 соответствует совпадению оригинального сигнала с восстановленным, а -1 — полному различию сигналов. Метрика учитывает яркость, контрастность и структуру изображения.

- *Метрика оценки восприятия человека* (англ. НРМ — Human Perception Metric) характеризует качество восстановленного сигнала на основе психофизиологических особенностей восприятия сигналов человеком. Является наиболее трудозатратной, т.к. требует участие людей в проведении эксперимента и анализе данных, однако является наиболее достоверной.

Выбор метрики оценки качества восстановления дефокусированного изображения зависит от конкретной задачи и требований к качеству восстановленного изображения.

1.4 Постановка задачи

На рисунке 1.22 представлена IDEF0 диаграмма уровня A0 для метода восстановления дефокусированных изображений на основе определенных параметров искажения.

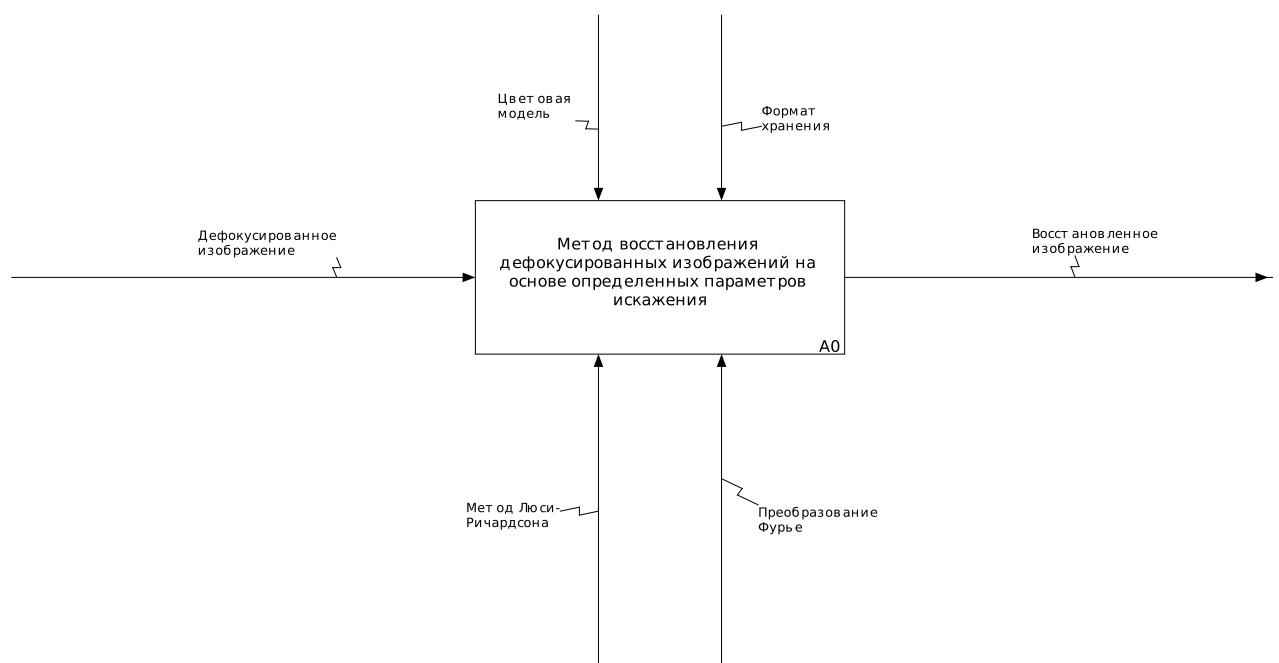


Рисунок 1.22 – IDEF0 диаграмма уровня A0

Выводы

В данном разделе были рассмотрены основные понятия предметной области, а также методы деконволюции с учетом априорной информации об искажении, а также при ее отсутствии.

В качестве цветовой модели цифрового изображения была выбрана модель RGB, т.к. разрабатываемый метод является линейным, т.е. при обработке непосредственно самого изображения нет необходимости обрабатывать какие-либо локальные подобласти.

В качестве форматов хранения выбраны расширения PNG, JPG и BMP как наиболее распространенные форматы хранения цифровых изображений, получаемых при съемке фотокамерой.

Для оценки качества восстановления будут использованы метрики PSNR (пиковое отношение сигнал – шум) и НРМ (оценка восприятия человека). В рамках рассматриваемой задачи метрика SSIM является неприменимой, т.к. по условию отсутствует априорная информация об искажении, а в частности и оригинальное изображение [22].

Метрика PSNR позволяет автоматизировать процесс оценки качества восстановления, однако эта оценка неполная, т.к. не учитывает особенности психофизиологического восприятия человеком сигнала. Наиболее точной в рамках поставленной задачи будет метрика НРМ. В рамках поставленной задачи в роли эксперта выступает автор работы.

В качестве метода классической деконволюции был выбран метод Люси – Ричардсона, т.к. фильтр Винера существенно чувствителен к шуму и статистической информации о нем, а метод регуляризации Тихонова имеет высокую вычислительную сложность и требует предварительную нетривиальную оценку параметров регуляризации для изображения. Выбранный метод является применимым, когда модель искажения может быть оценена, в частности в случае дефокусировки эта оценка возможна.

Процесс слепой деконволюции обычно включает в себя оценку неизвестной функции свертки и восстановление исходного сигнала. В качестве подхода к решению задачи «слепой» деконволюции было решено определять ФРТ на основе кепстрального анализа, который в свою очередь базируется на анализе амплитудной составляющей спектра исходного изображения.

Было выяснено, что функция рассеяния точки в случае дефокусировки

фотокамеры имеет форму диска, радиус которого равен радиусу дефокусировки, а структура кепстров содержит кольца, радиус которых зависит от степени дефокусировки.

Важно отметить, что слепая деконволюция является сложной задачей, особенно при наличии шума или других искажений в наблюдаемых данных, а также имеет ряд особенностей и ограничений, в связи с чем в дальнейшем необходимо определить область применимости предлагаемого метода восстановления.

2 Конструкторский раздел

В данном разделе сформулированы требования и ограничения к разрабатываемому методу. Разработан метод восстановления дефокусированных изображений на основе определенных параметров искажения. Описаны основные этапы разработки в виде детализированной диаграммы IDEF0 и схем алгоритмов, а также изложены особенности излагаемого метода. Спроектировано программное обеспечение для реализации разрабатываемого метода.

2.1 Требования и ограничения к разрабатываемому методу

К методу восстановления дефокусированных изображений на основе определенных параметров предъявляются следующие требования:

1. Определять параметр искажения (радиус дефокусировки).
2. Определять функцию рассеяния точки, которая была применена к искаженному изображению.
3. Восстанавливать искаженное изображение с использованием ФРТ.

Также представлен ряд ограничений для разрабатываемого метода:

1. Качество восстановления может быть неудовлетворительным, если изображение было подвергнуто существенному сжатию.
2. Качество восстановления может быть неудовлетворительным, если изображение с высоким уровнем шума и/или имеет высокочастотные детали (выбросы интенсивности).

Не предполагается обработка изображений с частичной дефокусированной, т.е. когда искажение присутствует только в определенной области, а не распространяется на всю площадь. Для обработки дефокусировки в этом случае можно выделить часть необходимую изображения. Не предполагается обработка заведомо недефокусированных изображений. Не предполагается повторное применение метода к изображению.

Также естественным ограничением семейства методов деконволюции (как «слепой», так и классической) является тот факт, что чем сильнее искажение (больше радиус дефокусировки), тем менее эффективен метод, т.к. дефокусировка является лишь частично обратимым процессом. Это связано с тем, что при увеличении силы влияния искажения теряется снижается количество информации на изображении.

2.2 Требования к разрабатываемому программному обеспечению

К разрабатываемому программному обеспечению предъявляются следующие требования:

1. Возможность загрузки изображений в формате PNG, JPG или BMP.
2. Возможность обработки как RGB – изображений, так и изображений в тонах серого.
3. Возможность просмотра результата восстановления в сравнении с исходным изображением.
4. Возможность сохранения результата в отдельный файл в формате PNG, JPG или BMP.
5. Если время выполнения программы может превышать комфортное время реакции системы для человека, необходимо обеспечить вывод предупреждающего сообщения.
6. Разрабатываемое ПО должно корректно реагировать на любые действия пользователя.

2.3 Основные этапы разрабатываемого метода

2.3.1 IDEF-0 диаграмма уровня A1

На рисунке 2.1 представлена диаграмма IDEF0 уровня A1 для разрабатываемого метода.

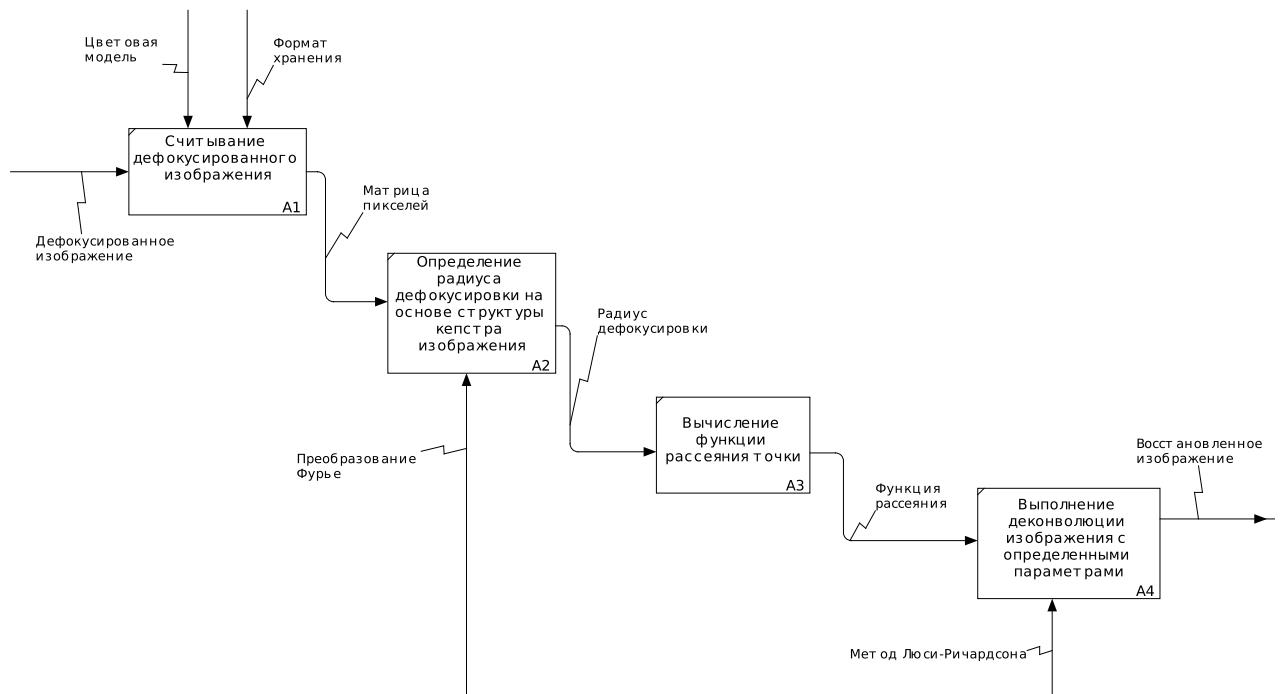


Рисунок 2.1 – IDEF0 – диаграмма уровня A1

2.3.2 Схемы алгоритмов

На первом этапе исходное дефокусированное изображение считывается в матрицу пикселей, с которой в дальнейшем будут происходить все вычисления. В соответствии с требованиями, изображение может одноканальным (grayscale) или трехканальным (RGB). В первом случае все действия производятся однократно, во втором — необходимо воспроизвести соответствующие вычисления для каждого из каналов, а затем результат объединить в трехканальное изображение.

На втором этапе происходит определение радиуса дефокусировки на основе структуры кепстера изображения. Основная идея предлагаемого метода заключается именно в этом этапе.

На рисунке 2.2 представлена схема, демонстрирующая влияние радиуса на данную структуру. В соответствии с этой закономерностью решено вычислять радиус дефокусировки как отношение радиуса ближайшего к центру

кольца к размеру радиального профиля, равного половине ширины изображения.

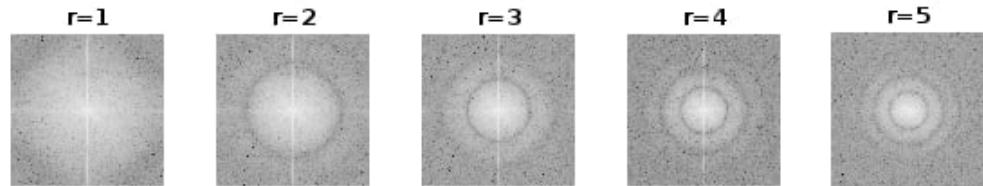


Рисунок 2.2 – Влияние радиуса дефокусировки на структуру кепстра изображения

На рисунке 2.3 представлен алгоритм вычисления кепстра.

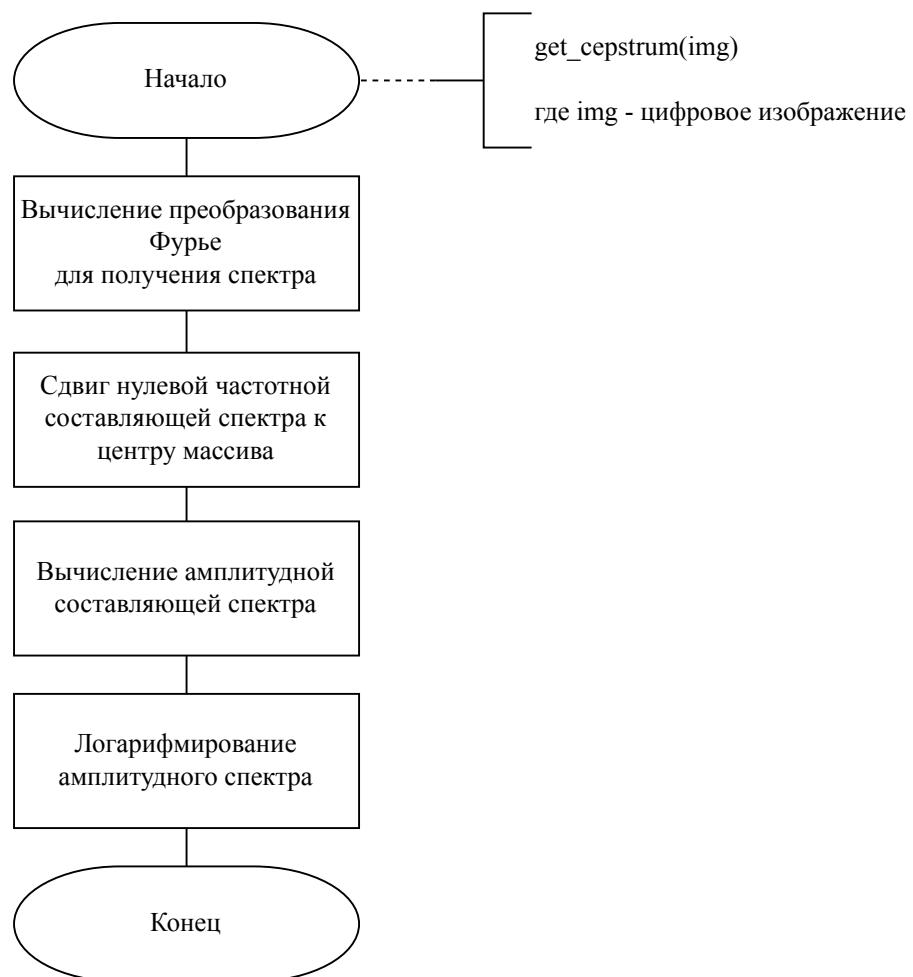


Рисунок 2.3 – Алгоритм вычисления кепстра изображения

Сдвиг нулевой частотной составляющей необходим для удобства восприятия полученной структуры. Логарифмирование кепстра производится также с целью повышения качества визуального восприятия. Для получения амплитудной составляющей необходимо вычислить модуль кепстра.

На рисунке 2.4 представлен алгоритм вычисления радиуса дефокусировки на основе кепстрального анализа.

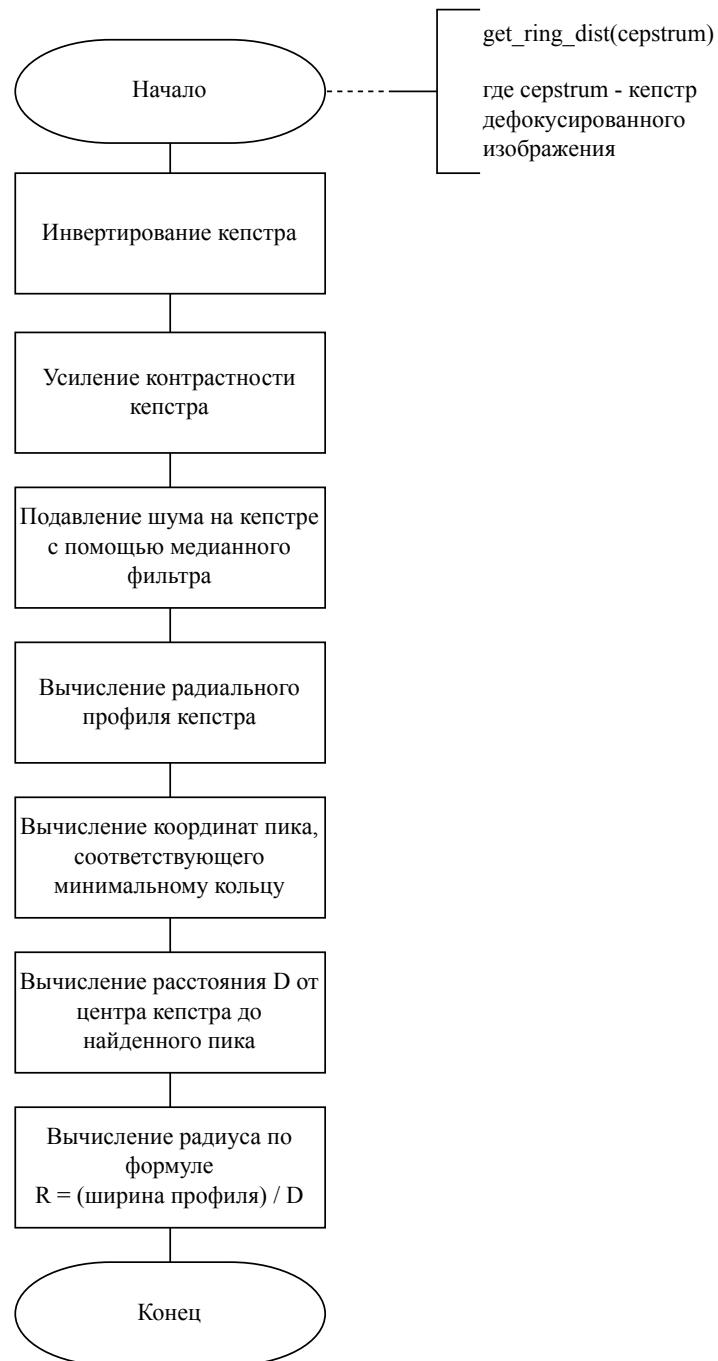


Рисунок 2.4 – Алгоритм вычисления радиуса дефокусировки

В рамках этого этапа входным параметром является матрица пикселей, выходным — радиус дефокусировки.

Инвертирование кепстра выполняется по причине того, что большинство средств обработки сигналов предоставляет широкий набор обработки пиков (максимумов) интенсивности.

Усиление контрастности проводится с целью повышения точности распознавания радиуса кольца кепстрап.

Применение медианного фильтра позволяет подавить шум на изображении, сгладив помеху типа «соль - перец» в центре кепстрапа. Размер окна выбран так, чтобы не потерять информацию о деталях наблюдаемой структуры.

На *третьем* этапе метода составляется функция рассеяния точки, описывающая процесс искажения, на основе определенного на предыдущем этапе радиуса дефокусировки. Данная матрица имеет тип квадратной матрицы размером, равным радиусу.

В данный квадрат вписан круг радиусом, также равным двойному радиусу дефокусировки. Для ячеек матрицы, принадлежащих этому кругу,

значение интенсивности вычисляется как $\frac{1}{\pi r^2}$, где r — радиус. Для остальных ячеек матрицы значение интенсивности равно 0.

На рисунке 2.5 представлен общий вид функции рассеяния точки.

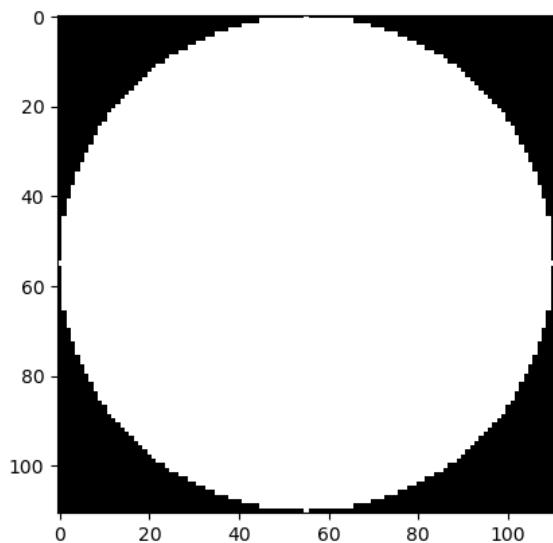


Рисунок 2.5 – Общий вид ФРТ

В рамках этого этапа входным параметром является радиус дефокусировки, выходным — квадратная матрица пикселей, описывающая ФРТ заданного радиуса.

На рисунке 2.6 представлен алгоритм вычисления ФРТ на основе радиуса.

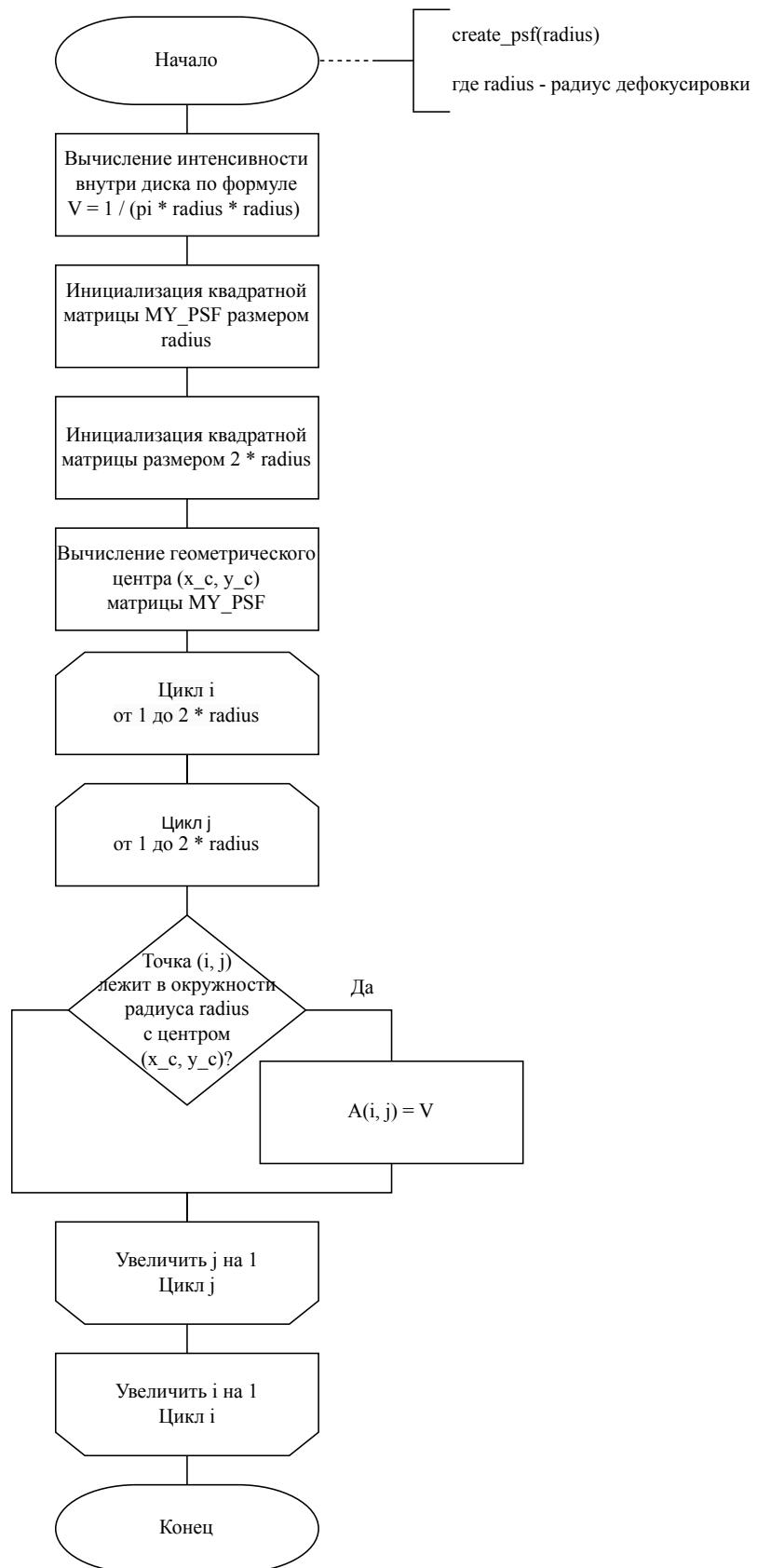


Рисунок 2.6 – Алгоритм вычисления ФРТ на основе радиуса дефокусировки

На четвертом этапе производится деконволюция с использованием априорной информации методом Люси — Ричардсона. Т.к. этот алгоритм является классическим и не является объектом исследования, было принято решение не останавливаться подробно на реализации этого алгоритма.

На рисунке 2.7 представлен алгоритм классической деконволюции на основе определенных параметров искажения.

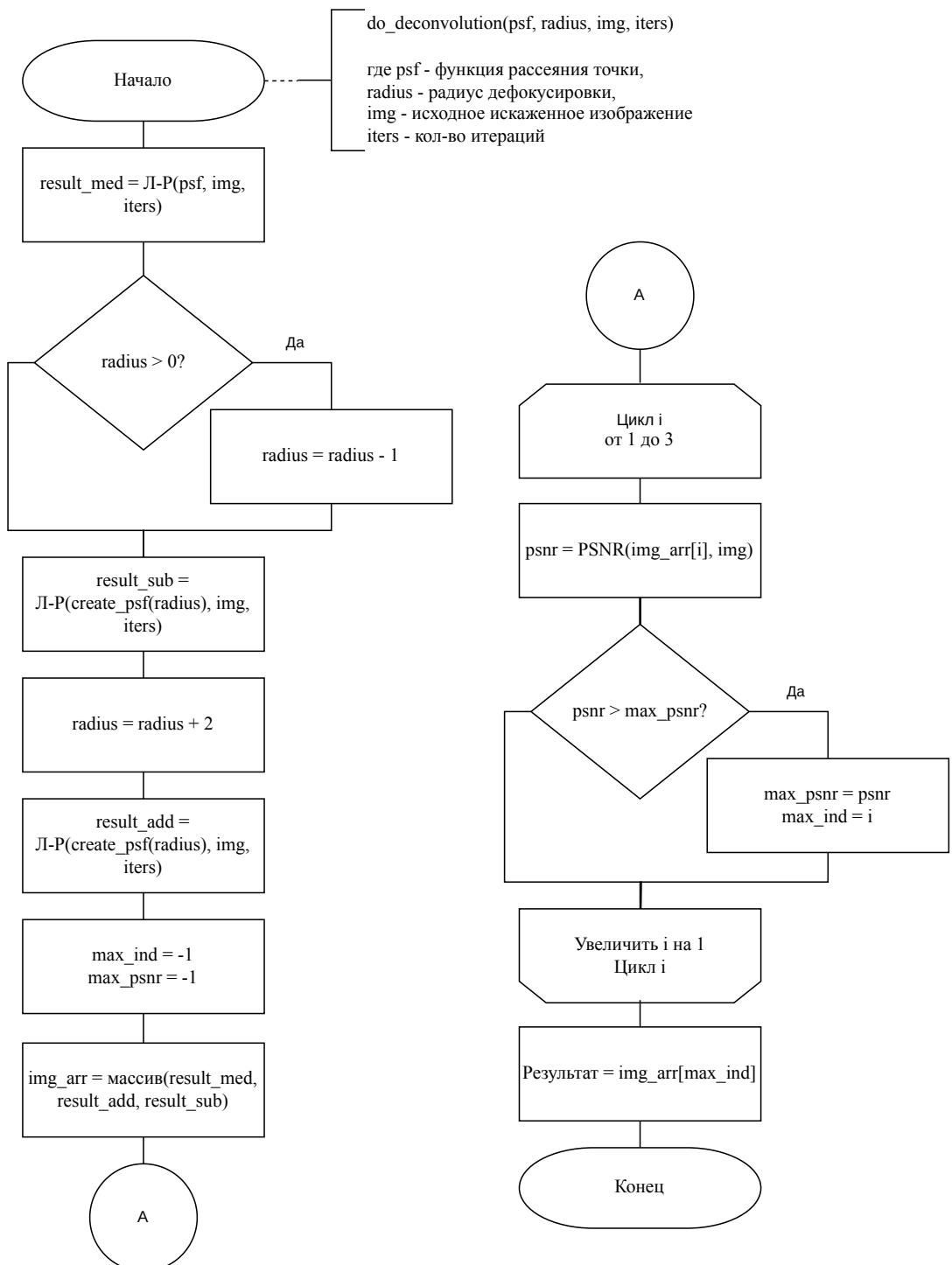


Рисунок 2.7 – Алгоритм классической деконволюции на основе определенных параметров искажения

Также было предложено варьировать на единицу в большую и меньшую сторону радиус дефокусировки, т.к. из-за наличия шума присутствует вероятность ошибиться на несколько пикселей с точностью вычисления радиуса.

Для выбора наилучшего приближения из трех было предложено использовать метрику «пиковое соотношение сигнал – шум» (англ. PSNR – Peak Signal Noise Ratio). Чем больше значение этой метрики, тем лучше произошло восстановление по предположению.

В рамках заключительного этапа входными параметрами являются радиус дефокусировки, ФРТ и исходное изображение, выходным — матрица пикселей, соответствующая восстановленному изображению.

2.3.3 Структура разрабатываемого программного обеспечения

На рисунке 2.8 представлена диаграмма компонентов разрабатываемого программного обеспечения.

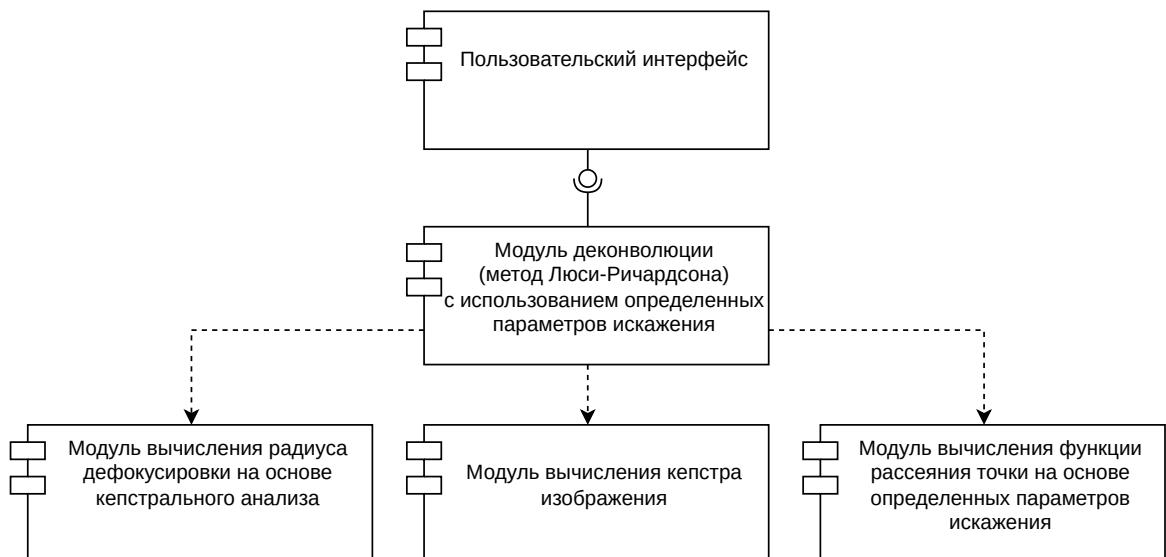


Рисунок 2.8 – Диаграмма компонентов разрабатываемого ПО

Выводы

В данном разделе были сформулированы требования и ограничения к разрабатываемому методу и соответствующему ПО, рассмотрены основные этапы разрабатываемого метода в виде детализированной диаграммы IDEF0 и схем алгоритмов, а также спроектирована структура разрабатываемого ПО.

3 Технологический раздел

В данном разделе обоснован выбор средств программной реализации предлагаемого метода, описан формат входных и выходных данных. Представлены детали реализации метода, а также описано взаимодействия пользователя с программным обеспечением.

3.1 Выбор средств реализации программного обеспечения

Выбор языка программирования

В качестве языка программирования для реализации предлагаемого метода был выбран MatLAB [23].

Выбор обусловлен тем, что данный язык имеет широкие возможности для работы с сигналами, для расчета и проектирования аналоговых и цифровых фильтров, для построения их частотных, импульсных и переходных характеристик.

В наличии также имеются средства для спектрального анализа и синтеза, в частности, для реализации прямого и обратного преобразования Фурье. [24]

3.1.1 Выбор среды разработки

Matlab предоставляет интегрированную среду разработки MatLAB IDE, предоставляющую множество инструментов и функций для разработки, отладки и выполнения кода на MatLAB. Данная среда разработки предоставляет такие компоненты как текстовый редактор, панель переменных окружения, история команд, интеграция файловой системы и др.

3.1.2 Используемые расширения

Для работы с цифровыми изображениями было использовано расширение Image Processing ToolBox [25], предоставляющее полный набор стандартных алгоритмов и приложений для обработки изображений, анализа, визуализации и разработки новых алгоритмов. Для обработки сигналов — расширение Signal Processing ToolBox [26], предоставляющее функции и интерактивные приложения для анализа, предобработки и выделения признаков из сигналов. Для компиляции приложения в независимый исполняемый файл

использовалось расширение MatLAB Compiler [27].

3.2 Формат входных и выходных данных

Входные данные для разрабатываемого метода:

- дефокусированное цифровое изображение в одном из следующих расширений: PNG, JPG или BMP.

Выходные данные для разрабатываемого метода:

- восстановленное цифровое изображение в одном из следующих расширений: PNG, JPG или BMP.

3.3 Детали реализации предлагаемого метода

В листинге 3.1 представлена функция вычисления кепстера изображения.

Листинг 3.1 – Функция вычисления кепстера изображения

```
1 function img_cepstrum = cepstrum(img)
2     spectrum = fftshift(fft2(img));
3     img_cepstrum = log(1+abs(spectrum .* spectrum));
4 end
```

В листинге 3.2 представлена функция вычисления радиуса дефокусировки на основе кепстрального анализа.

Листинг 3.2 – Функция вычисления радиуса дефокусировки

```
1 function defocus_radius = radius(cepstrum)
2     max_intensity = double(max(cepstrum(:)));
3     cepstrum = max_intensity - double(cepstrum);
4     cepstrum = medfilt2(cepstrum, [2 2]);
5
6     x_c = idivide(int32(size(cepstrum, 1)), 2) + mod(size(cepstrum, 1), 2)
7         + 1;
8     central_row = cepstrum(x_c, :);
9     half_central_row = central_row(1:size(central_row, 2) / 2);
10    half_central_row = flip(half_central_row);
11
12    [peaks, peaks_locations] = findpeaks(half_central_row);
13    [~, new_peak_locations] = findpeaks(peaks, 'NPeaks', 1);
14    defocus_radius = round(x_c / peaks_locations(new_peak_locations));
```

В листинге 3.3 представлена функция определения функции рассеяния точки на основе вычисленного радиуса фокусировки.

Листинг 3.3 – Функция определения ФРТ

```
1 function defocus_psf = psf(radius)
2     value = 1 / pi / double(radius) / double(radius);
3
4     defocus_psf = zeros(radius * 2, radius * 2);
5
6     x_c = radius;
7     y_c = radius;
8
9     w = radius * 2;
10
11    for i = 1:w
12        for j = 1:w
13            if (i - x_c) * (i - x_c) + (j - y_c) * (j - y_c) <= radius *
14                radius
15                defocus_psf(i, j) = value;
16            end
17        end
18    end
19 end
```

В листинге 3.4 представлена функция одноканальной деконволюции на основе метода Люси – Ричардсона. Полный вариант функции представлен в приложении А.

Листинг 3.4 – Функция деконволюции на основе метода Люси – Ричардсона

```
1 function focused_img = my_blind_deconvolution(original_img)
2     import cepstrum.*
3     import radius.*
4     import psf.*
5
6     function img = do_gray()
7         img_cepstrum = cepstrum(original_img);
8         focus_radius = radius(img_cepstrum);
9         focus_psf = psf(focus_radius);
10        med_img = deconvlucy(original_img, focus_psf, 100);
11
12        if focus_radius - 1 > 0
13            focus_radius = focus_radius - 1;
14            focus_psf = psf(focus_radius);
15        end
16        sub_img = deconvlucy(original_img, focus_psf, 100);
17
18        focus_radius = focus_radius + 2;
19        focus_psf = psf(focus_radius);
20        add_img = deconvlucy(original_img, focus_psf, 100);
21
22        psnr_value_med = psnr(med_img, original_img);
23        psnr_value_sub = psnr(sub_img, original_img);
24        psnr_value_add = psnr(add_img, original_img);
25        [~, max_index] = max([psnr_value_med psnr_value_sub
26                               psnr_value_add]);
27
28        if max_index == 1
29            img = med_img;
30        elseif max_index == 2
31            img = sub_img;
32        else
33            img = add_img;
34        end
35    ... % RGB processing
36    if length(size(original_img)) == 2
37        focused_img = do_gray();
38    else
39        ... % RGB processing
40    end
41 end
```

3.4 Описание взаимодействия пользователя с программным обеспечением

На рисунке 3.1 представлен разработанный пользовательский интерфейс программного обеспечения.

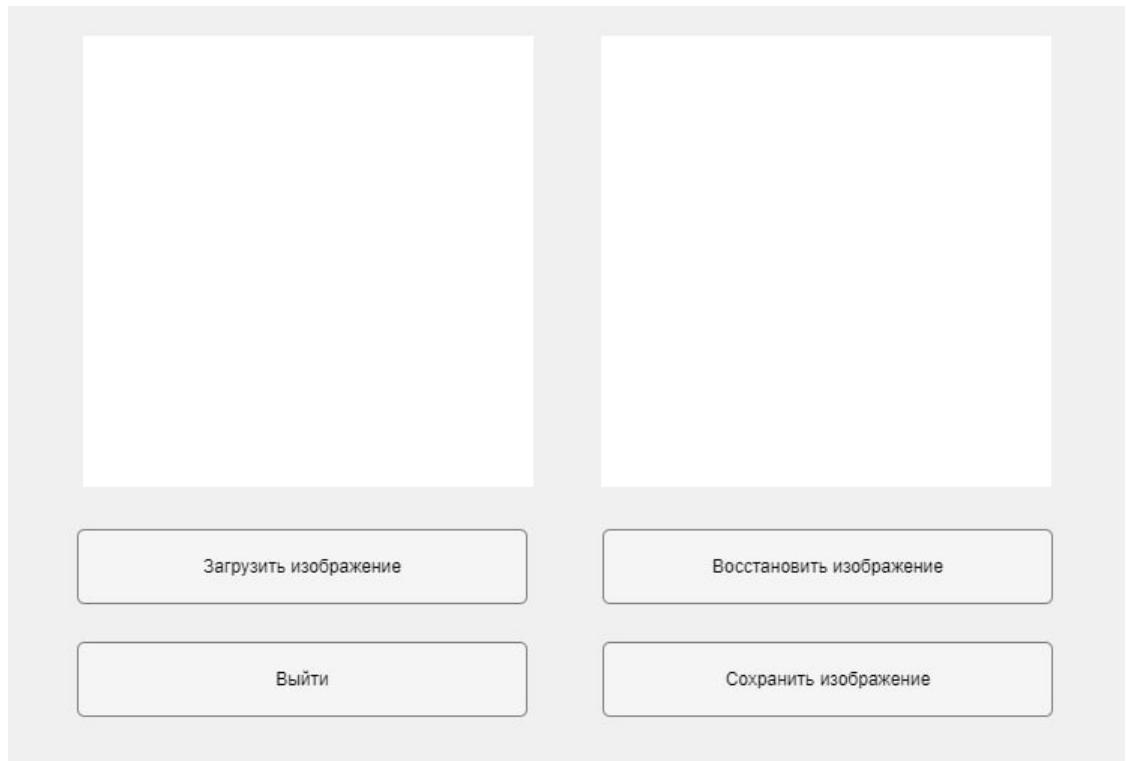


Рисунок 3.1 – Пользовательский интерфейс программного обеспечения

Разработанный интерфейс позволяет выбрать исходное цифровое изображение для обработки из файловой системы, применить операцию восстановления к загруженному изображению, а также сохранить полученный результат в отдельный файл.

Процесс выполнения операции восстановления может превышать комфортное время реакции интерактивной системы для человека, в связи с чем при обработке выводится специальное уведомление (рисунок 3.2).

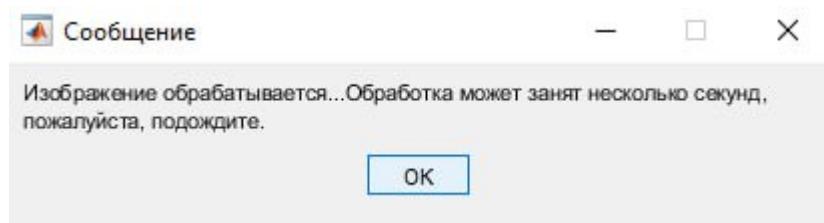


Рисунок 3.2 – Уведомление о возможном превышении времени реакции системы

При попытке восстановить изображение без предварительной загрузки исходного изображения также выводится специальное уведомление для пользователя (рисунок 3.3).

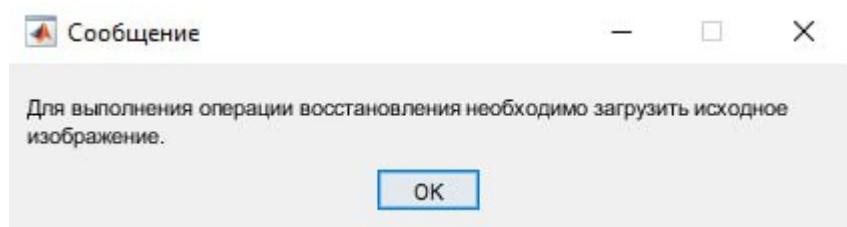


Рисунок 3.3 – Уведомление о необходимости предварительной загрузки изображения

3.5 Сборка и запуск проекта

В случае, если на персональный компьютер пользователя уже установлен MatLAB, то для сборки и запуска разработанного ПО необходимо выполнить следующие шаги:

- проверить наличие компонентов Image Processing Toolbox и Signal Processing Toolbox;
- проверить, что все необходимые файлы и функции, связанные с проектом, находятся в одной папке или в структурированной иерархии папок;
- запустить программу MatLAB на компьютере;
- открыть папку проекта;
- запустить ПО либо через консольный интерфейс (gui.m), либо через соответствующую кнопку «Run» в панели инструментов.

В случае, если MatLAB не установлен, то необходимо выполнить следующие шаги:

- проверить, что исполняемый файл и другие файлы, связанные с проектом, находятся на целевой машине пользователя;
- запустить исполняемый файл;

- пройти инсталляцию компонента MatLab Runtime [28], предлагаемую при первом запуске проекта, с целью загрузки минимально необходимого программного обеспечения для запуска проекта без лицензионного экземпляра Matlab;
- запустить проект.

4 Исследовательский раздел

В данном разделе приведены примеры идеального случая восстановления, среднего и плохого. Произведены исследования для определения времени обработки изображения в зависимости от радиуса дефокусировки и цветовой модели, а также полноты решения поставленной задачи. На основе полученных экспериментальных данных сделаны соответствующие выводы.

4.1 Технические характеристики

Технические характеристики машины, на которой производились исследования:

- операционная система: Windows 10 64-bit;
- оперативная память: 16 Gb;
- процессор: AMD(R) Ryzen(TM) 5 4500U CPU @ 2.3 GHz;
- количество ядер: 8.

4.2 Исследование времени работы предложенного метода в зависимости от размера изображения и цветовой модели

Постановка исследования

Исследование заключается в определении времени обработки изображения в зависимости от двух факторов: его размера и цветовой модели (трехканальная цветовая RGB – модель или одноканальная серая (англ. Grayscale) модель).

Для замеров должны быть использованы дефокусированные изображения различных размеров и пропорций. Замеры необходимо провести несколько раз для минимизации влияния случайных факторов.

Гипотеза заключается в том, что графики зависимостей будут иметь линейный вид, а также время, затраченное на обработку цветного изображения, будет превышать время обработки одноканального изображения примерно в 3 раза.

Результаты исследования

В таблице 4.1 представлены результаты замеров времени выполнения алгоритма в зависимости от размера исходного изображения и цветовой модели (RGB – модель или Grayscale). Размер изображения измеряется в пикселях, а время — в секундах. Для замеров времени выполнения использовались команды *tic* и *toc*, представленные в MatLAB.

Таблица 4.1 – Зависимость времени обработки изображения от его размера и цветовой модели

Размер изображения, (Ш × В, пиксели)	RGB, сек.	Grayscale, сек.
300 × 300	4.45	1.5
300 × 500	10.29	3.26
400 × 400	10.23	2.94
500 × 500	16.96	4.70
300 × 900	17.07	5.84
600 × 600	20.48	6.87
500 × 900	29.76	10.06
700 × 700	28.07	9.74
500 × 1200	40.29	13.39
700 × 900	42.53	14.06
800 × 800	37.09	13.07
900 × 900	49.67	16.52
700 × 1200	57.75	19.35
1000 × 1000	62.09	20.59
1100 × 1100	77.00	27.06
1200 × 1200	93.28	32.32

На рисунке 4.1 представлена зависимость времени обработки изображения от количества пикселей и используемой цветовой модели.

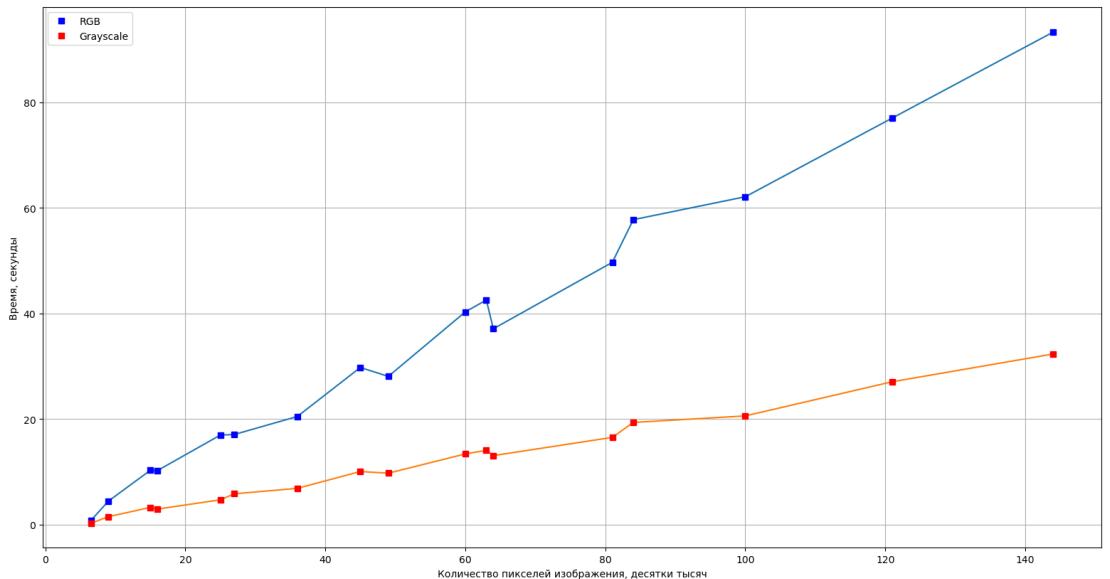


Рисунок 4.1 – Зависимость времени обработки изображения от количества пикселей и цветовой модели

Гипотеза подвердилась — график зависимости имеет линейный вид, а также обработка цветных изображений требует примерно в 3 раза больше времени, чем обработка одноканальных изображений, что подтверждает линейность работы алгоритма относительно аддитивной цветовой модели.

Таким образом, если цветовая информация об изображении не является критической для рассмотрения, то рекомендуется использовать изображения в тонах серого, т.к. с точки зрения структуры изображения они содержат ту же информацию, однако обработка такого изображения может быть значительно быстрее.

4.3 Исследование полноты решения поставленной задачи

Постановка исследования

Исследование заключается в определении точности вычисления (в процентах) радиуса дефокусировки в зависимости от его величины и цветовой модели (RGB — модель или одноканальная). Для замеров должны быть исполь-

зованы дефокусированные изображения различных размеров и пропорций.

Гипотеза заключается в том, что при радиусе размытия, равном 0 или 1, точность будет низкая, т.к. на кепстре структуры практически не будут наблюдаваться.

При достижении определенного порогового значения радиуса, которое необходимо установить в результате эксперимента, точность восстановления снова станет низкой, т.к. кольца на структуре кепстра будут сконцентрированы ближе к центру, и, соответственно, будет сложно отличить границу кольца от шума вокруг центра. Между описанными двумя областями значений радиусов точность, по предположению, должна быть удовлетворительной.

Также предполагается, что точность вычислений для RGB – модели будет ниже, чем для серой, т.к. цветные изображения могут быть более подвержены шумам, а также ввиду корреляции между каналами и дополнительной сложности вычислений может происходить накопление ошибки.

Результаты исследования

На рисунке 4.2 представлена зависимость точности вычисления радиуса дефокусировки от его величины и цветовой модели. Для каждого радиуса размытия обрабатывался ряд изображений различного размера и пропорций, а результат усреднялся.

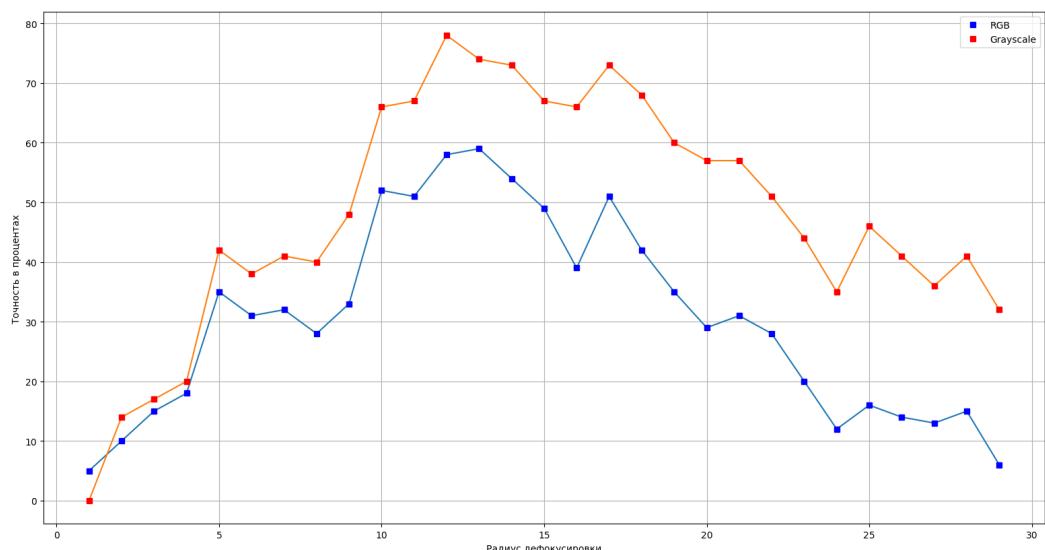


Рисунок 4.2 – Зависимость точности вычисления радиуса дефокусировки от его величины и цветовой модели

В таблице 4.2 представлены результаты исследования полноты полученного решения.

Для определения точности восстановления в случае одноканальной модели была предложена следующая метрика:

$$clean_value = \frac{1 - |R_{calc} - R|}{R}, \quad (4.1)$$

$$accuracy = \begin{cases} 0, & \text{если } clean_value < 0 \text{ или } |clean_value| > 1, \\ clean_value \cdot 100, & \text{иначе.} \end{cases} \quad (4.2)$$

где $accuracy$ — точность в процентах, R_{calc} — вычисленный радиус дефокусировки, R — реальный радиус. Для вычисления среднего значения для каждого радиуса использовалось значение $\frac{\sum_{i=1}^N accuracy_i}{N}$, где N — количество замеров.

С случае оценки точности восстановления цветного изображения метрика $accuracy$ вычислялась для каждого канала независимо, а для получения среднего результата использовалось значение $\frac{\sum_{i=1}^N r_accuracy_i \cdot g_accuracy_i \cdot b_accuracy_i}{N}$, где $r_accuracy_i$, $g_accuracy_i$, $b_accuracy_i$ — значение метрики i -го замера для красного, зеленого и синего цветовых каналов соответственно.

Таким образом, была определена область применимости предложенного метода: изображения с радиусом дефокусировки в диапазоне от 10 до 20 единиц.

Если цветовая информация об изображении не является критической для рассмотрения, то рекомендуется использовать изображения в тонах серого, т.к. для этой цветовой модели точность вычислений выше.

Таблица 4.2 – Зависимость точности вычисления радиуса дефокусировки от его величины и цветовой модели

Радиус дефокусировки, пиксели	RGB – модель	Grayscale – модель
0	0 %	0 %
1	5 %	0 %
2	10 %	14 %
3	15 %	17 %
4	18 %	20 %
5	35 %	42 %
6	31 %	38 %
7	32 %	41 %
8	28 %	40 %
9	33 %	48 %
10	52 %	66 %
11	51 %	67 %
12	58 %	78 %
13	59 %	74 %
14	54 %	73 %
15	49 %	67 %
16	39 %	66 %
17	51 %	73 %
18	42 %	68 %
19	35 %	60 %
20	29 %	57 %
21	31 %	57 %
22	28 %	51 %
23	20 %	44 %
24	12 %	35 %
≥ 25	$\leq 20\%$	$\leq 40\%$

4.4 Демонстрация вариантов работы метода в зависимости от точности вычислений

В связи с тем, что разрабатываемый метод основан на методе максимального правдоподобия, возможны несколько классов результатов восстановления: идеальный случай (реальный и вычисленный радиусы совпадают), средний (вычисление радиуса произошло с погрешностью) и плохой случай, когда реальный и вычисленный радиусы существенно не совпадают.

На рисунках 4.3 и 4.4 представлены примеры высокого (идеального) качества восстановления. Для проведения эксперимента были использованы изображения номера автомобиля [29] и кота [30].

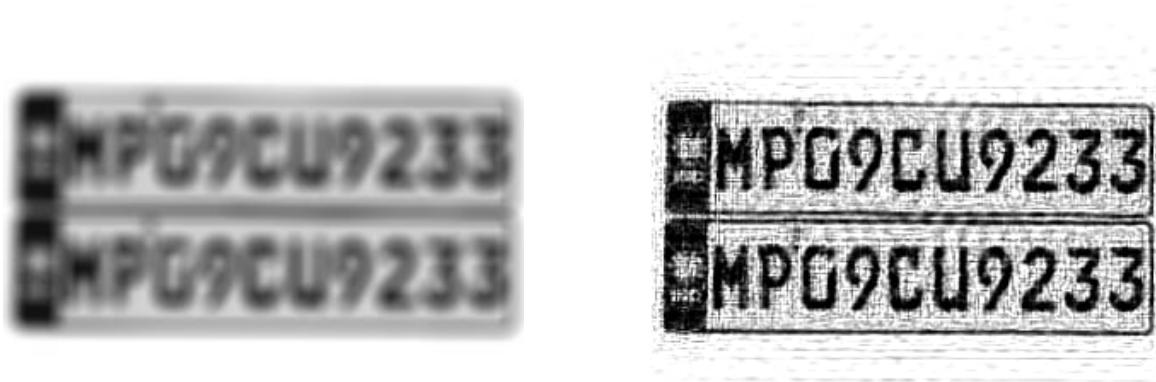


Рисунок 4.3 – Пример восстановления высокого качества

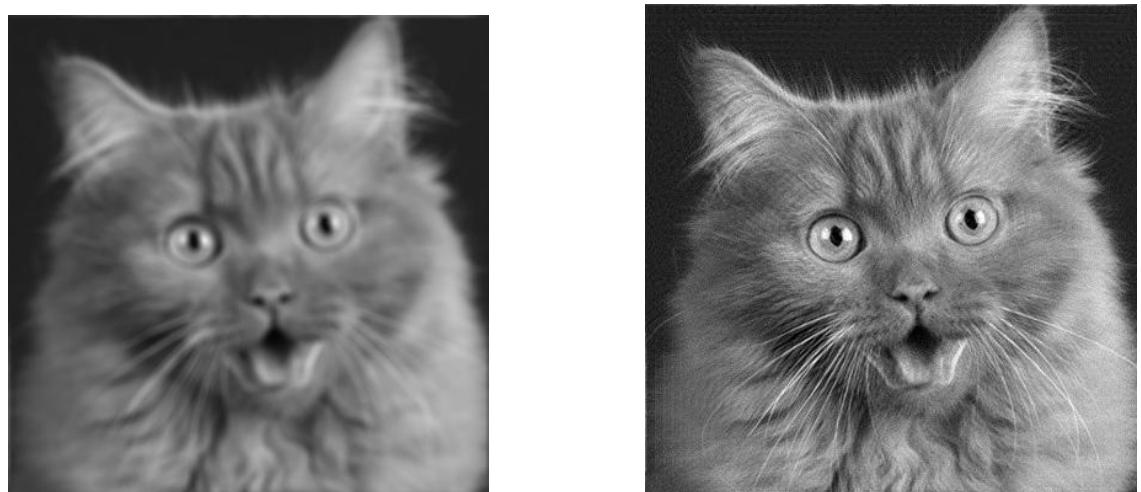


Рисунок 4.4 – Пример восстановления высокого качества

Пиковое отношение сигнал – шум для первой пары изображений составляет 14.7 дб, для второй – 23 дб.

На рисунках 4.5 и 4.6 представлены примеры среднего качества восстановления. Точность вычисления радиуса дефокусировки составляет ≈ 80 процентов.

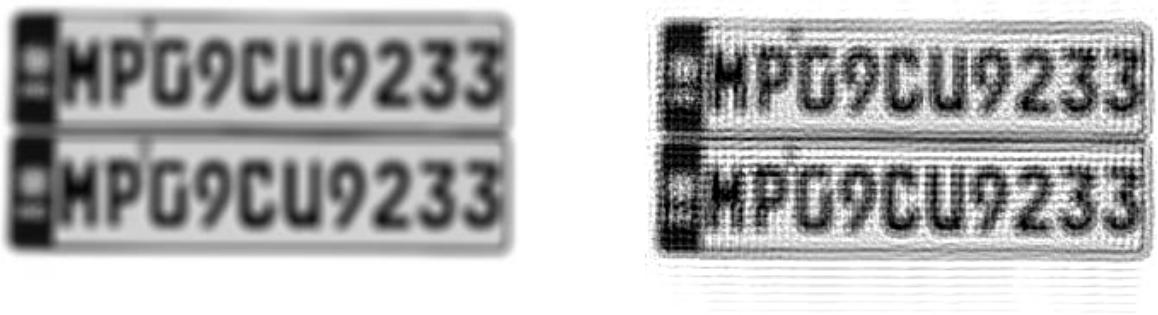


Рисунок 4.5 – Пример восстановления среднего качества

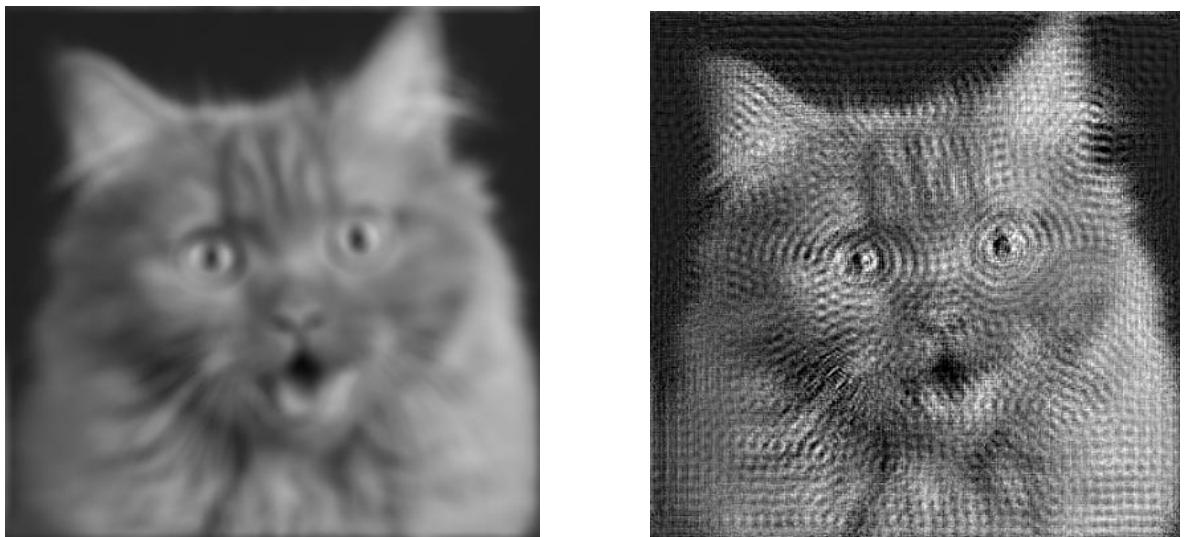


Рисунок 4.6 – Пример восстановления среднего качества

Пиковое отношение сигнал – шум для первой пары изображений составляет 19.5 дб, для второй – 20.22 дб.

На рисунках 4.7 и 4.8 представлены примеры низкого качества восстановления. Точность вычисления радиуса дефокусировки составляет ≈ 70 процентов.

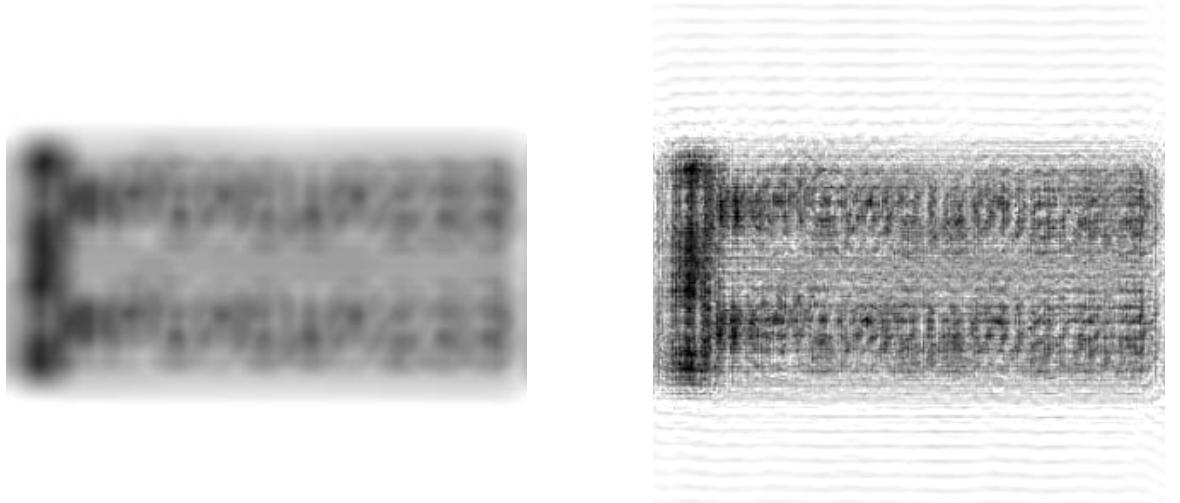


Рисунок 4.7 – Пример восстановления низкого качества

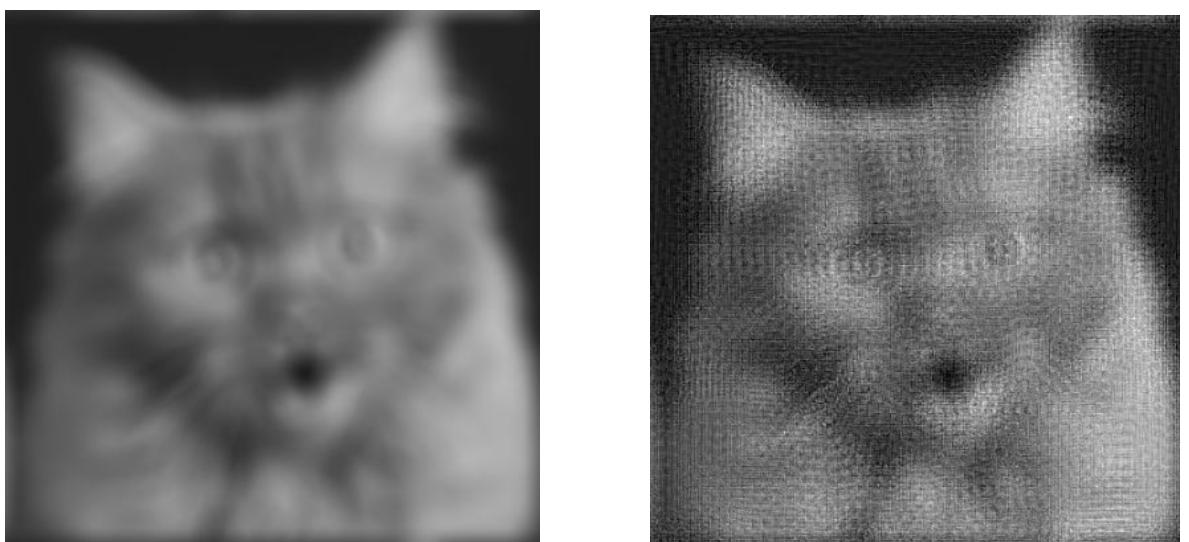


Рисунок 4.8 – Пример восстановления низкого качества

Пиковое отношение сигнал – шум для первой пары изображений составляет 21.9 дБ, для второй — 23.8 дБ.

Выводы

Было выявлено, что зависимость времени обработки изображения от его размера и цветовой модели имеет линейный вид, а обработка цветного изображения в среднем требует примерно в 3 раза больше времени, чем обработка серого.

Была определена область применимости предложенного метода: в диапазоне радиуса дефокусировки от 10 до 20 единиц точность восстановления является удовлетворительной, а в случае обработки серого изображения этот диапазон еще шире, поэтому если учет информации о цвете изображения некритичен, то рекомендуется использовать серое изображение вместо цветного в целях сокращения времени обработки и повышения качества результата. Была описана метрика, используемая для оценки точности.

Были рассмотрены варианты работы программы: идеальный случай, нормальный и плохой, а также проведен анализ значения пикового отношения сигнал – шум для каждого варианта. Было выявлено, что метрика PSNR не всегда отражает действительное качество восстановления, т.к. не учитывает психофизиологические особенности восприятия изображения человеком.

ЗАКЛЮЧЕНИЕ

В данной работе была рассмотрена задача восстановления дефокусированных изображений на основе определенных параметров искажения.

Для достижения поставленной задачи был проведен анализ предметной области дефокусированных изображений, а также существующих методов восстановления без учета априорной информации об искажении. Было выяснено, что функция рассеяния точки в случае дефокусировки фотокамеры имеет форму диска.

Для решения задачи «слепой» деконволюции был предложен метод, согласно которому ФРТ, описывающая искажающий процесс, определялась на основе радиуса колец, которые наблюдаются на структуре кепстра дефокусированных изображений. Также были рассмотрены критерии оценки качества восстановления, основным из которых является пиковое соотношение сигнал – шум.

В результате выполнения данной работы было спроектировано и реализовано соответствующее программное обеспечение, позволяющее пользователю загружать искаженное изображение, применять восстановление к нему и сохранять результат. Таким образом, цель работы достигнута.

Разработанный метод имеет ряд ограничений, однако может быть применим даже в условиях отсутствия априорной информации об искажающем процессе. Согласно исследованиям, область применимости предложенного метода — изображения с радиусом дефокусировки от 10 до 20 пикселей. Также было выяснено, что в случае, когда пренебрежение цветовой информацией является некритичным, то рекомендуется обрабатывать изображение в тонах серого.

В качестве перспектив дальнейшего развития можно рассмотреть несколько направлений, таких как учет сложной модели искажения, возникающего в реальных ситуациях помимо моделирования известных ФРТ; улучшение обработки шума на изображении; применение «слепой» деконволюции к видеофайлам; развитие комбинированных методов деконволюции, охватывающих более широкую область применения; учет частичной дефокусировки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Раков Д. Л. Морфологический анализ и синтез невозможных объектов //Невозможные объекты и оптические иллюзии в современном искусстве и дизайне (Традиционные и компьютерные технологии). – 2014. – С. 16-19.
2. Констандогло А. В. Методы восстановления расфокусированных и смязгленных изображений //Политехнический молодежный журнал. – 2020. – №. 5. – С. 5-5.
3. Малыхина Г. Ф., Меркушева А. В. Сеть с симметричной функцией преобразования нейронов для подавления искажений и восстановления изображения //Научное приборостроение. – 2008. – Т. 18. – №. 2. – С. 73-80.
4. Фотоаппараты и съемочные фотографические объективы //Государственный стандарт союза ССР. - 1983.
5. Бадюк А. А. АНАЛИЗ ИННОВАЦИОННОГО РОСТА НА РЫНКЕ ЦИФРОВЫХ ФОТОАППАРАТОВ //Экономика и социум. – 2019. – №. 12 (67). – С. 236-242.
6. Фомин А. В. Общий курс фотографии //М.:«Лёгкая индустрия. – 1975.
7. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – Litres, 2022.
8. Грищенцев А. Ю., Коробейников А. Г. Методы и модели цифровой обработки изображений. – 2014.
9. Шемплинер В. В. Восстановление дефокусированных изображений методом двумерного преобразования Фурье и регуляризации Тихонова //Научно-технический вестник информационных технологий, механики и оптики. – 2008. – №. 48. – С. 56-66.
10. Чочича, П. А. Определение вида и параметров искажений изображения по Фурье-спектру сигнала //Институт проблем передачи информации. - 2019.
11. Бойко Б. П. и др. Спектр сигнала.

12. A bunch of pink flowers that are blooming [Электронный ресурс]. Режим доступа: <https://unsplash.com/photos/OBLADaipjv4> (дата обращения 24.05.2023).
13. Соловьев В. Е. Неэталонная оценка качества в задачах фильтрации, восстановления и сжатия изображений : дис. – Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых, 2013.
14. Ахмад Х. М. Сравнительное исследование эффективности различных методов кепстрального описания речевых сигналов в задачах распознавания //Вестник Тамбовского государственного технического университета. – 2007. – Т. 13. – №. 4. – С. 887-891.
15. Dosselmann R. W., Yang X. D. No-reference noise and blur detection via the fourier transform //Dept. of Computer Science, Univ. of Regina. Regina, SK (Saskatchewan), Canada. – 2012.
16. Воропаева Н. В. и др. Дискретное преобразование Фурье в обработке сигналов //Самара: Изд-во"Самар. ун. – 2015. – Т. 2015.
17. Convolution [Электронный ресурс]. Режим доступа: <https://paperswithcode.com/method/convolution> (дата обращения 22.05.2023).
18. Набиев М., Дорогиницкий М. М. Восстановление размытых изображений //Математическое моделирование и информационные технологии. – 2018. – С. 34-35.
19. Сизиков В. С., Лавров А. В. Устойчивые методы математико-компьютерной обработки изображений и спектров //СПб: Университет ИТМО. – 2018.
20. Чурсин М. А. ВОССТАНОВЛЕНИЕ РАСФОКУСИРОВАННЫХ И СМАЗАННЫХ ИЗОБРАЖЕНИЙ //Россия молодая: передовые технологии – в промышленность!. – 2013. – №. 2. – С. 099-101.

21. Переславцева Е. Е., Филиппов М. В. Метод ускоренного восстановления изображений, смазанных при движении //Машиностроение и компьютерные технологии. – 2012. – №. 02. – С. 30.
22. Hore A., Ziou D. Image quality metrics: PSNR vs. SSIM //2010 20th international conference on pattern recognition. – IEEE, 2010. – С. 2366-2369.
23. Math. Graphics. Programming. [Электронный ресурс]. Режим доступа: <https://www.mathworks.com/products/matlab.html> (дата обращения 22.05.2023).
24. Лазарев Ю. Ф. Начала программирования в среде MatLAB: Учебное пособие //К.: НТУУ «КПИ. – 2003.
25. Image Processing Toolbox [Электронный ресурс]. Режим доступа: <https://www.mathworks.com/products/image.html> (дата обращения 22.05.2023).
26. Signal Processing Toolbox [Электронный ресурс]. Режим доступа: <https://www.mathworks.com/products/signal.html> (дата обращения 22.05.2023).
27. MATLAB Compiler [Электронный ресурс]. Режим доступа: <https://www.mathworks.com/products/compiler.html> (дата обращения 22.05.2023).
28. MATLAB Runtime [Электронный ресурс]. Режим доступа: <https://www.mathworks.com/products/compiler/matlab-runtime.html> (дата обращения 22.05.2023).
29. Car Number [Электронный ресурс]. Режим доступа: <https://mavink.com/post/260AB0E02ED9872BD33ED4F3575DEDF72CAM> (дата обращения 24.05.2023).
30. Cats meow to say hello [Электронный ресурс]. Режим доступа: <https://www.worldsbestcatlitter.com/wp-content/uploads/2017/05/1-hello.jpg> (дата обращения 24.05.2023).

ПРИЛОЖЕНИЕ А

Полный код функции слепой деконволюции

В листингах А.1 - А.3 представлен полный код функции слепой деконволюции.

Листинг А.1 – Функция слепой деконволюции

```
1 function focused_img = my_blind_deconvolution(original_img)
2     import cepstrum.*
3     import radius.*
4     import psf.*
5
6     function img = do_gray()
7         img_cepstrum = cepstrum(original_img);
8         focus_radius = radius(img_cepstrum);
9         focus_psf = psf(focus_radius);
10        med_img = deconvlucy(original_img, focus_psf, 100);
11
12        if focus_radius - 1 > 0
13            focus_radius = focus_radius - 1;
14            focus_psf = psf(focus_radius);
15        end
16        sub_img = deconvlucy(original_img, focus_psf, 100);
17
18        focus_radius = focus_radius + 2;
19        focus_psf = psf(focus_radius);
20        add_img = deconvlucy(original_img, focus_psf, 100);
21
22        psnr_value_med = psnr(med_img, original_img);
23        psnr_value_sub = psnr(sub_img, original_img);
24        psnr_value_add = psnr(add_img, original_img);
25
26        [~, max_index] = max([psnr_value_med psnr_value_sub
27                               psnr_value_add]);
28
29        if max_index == 1
30            img = med_img;
31        elseif max_index == 2
32            img = sub_img;
33        else
34            img = add_img;
35        end
36    end
```

Листинг А.2 – Функция слепой деконволюции (продолжение)

```
1      function channel_radius = get_radius(channel)
2          channel_cepstrum = cepstrum(channel);
3          channel_radius = radius(channel_cepstrum);
4      end
5
6      function focused_channel = apply_psf(channel, radius)
7          chanel_focus_psf = psf(radius);
8          focused_channel = deconvlucy(channel, chanel_focus_psf, 100);
9      end
10
11     function img = deconvolve_with_radiiuses(red_channel_radius,
12         green_channel_radius, blue_channel_radius)
13         red_channel_focused = apply_psf(red_channel, red_channel_radius);
14         green_channel_focused = apply_psf(green_channel,
15             green_channel_radius);
16         blue_channel_focused = apply_psf(blue_channel, blue_channel_radius);
17         img = cat(3, red_channel_focused, green_channel_focused,
18             blue_channel_focused);
19     end
20
21     function update_radiiuses_neg()
22         if (red_channel_radius - 1) ~= 0
23             red_channel_radius = red_channel_radius - 1;
24         end
25         if (green_channel_radius - 1) ~= 0
26             green_channel_radius = green_channel_radius - 1;
27         end
28         if (blue_channel_radius - 1) ~= 0
29             blue_channel_radius = blue_channel_radius - 1;
30         end
31     end
32
33     function update_radiiuses_pos()
34         red_channel_radius = red_channel_radius + 2;
35         green_channel_radius = green_channel_radius + 2;
36         blue_channel_radius = blue_channel_radius + 2;
37     end
38
39     if length(size(original_img)) == 2
40         focused_img = do_gray();
41     else
42         red_channel = original_img(:, :, 1);
43         green_channel = original_img(:, :, 2);
44         blue_channel = original_img(:, :, 3);
```

Листинг А.3 – Функция слепой деконволюции (продолжение)

```
1      red_channel_radius = get_radius(red_channel);
2      green_channel_radius = get_radius(green_channel);
3      blue_channel_radius = get_radius(blue_channel);
4
5      med_rgb_img = deconvolve_with_radiuses(red_channel_radius,
6                                              green_channel_radius, blue_channel_radius);
7
8      update_radiuses_neg();
9      sub_rgb_img = deconvolve_with_radiuses(red_channel_radius,
10                                             green_channel_radius, blue_channel_radius);
11
12     update_radiuses_pos();
13     add_rgb_img = deconvolve_with_radiuses(red_channel_radius,
14                                             green_channel_radius, blue_channel_radius);
15
16
17     [~, max_index] = max([rgb_psnr_value_med rgb_psnr_value_sub
18                           rgb_psnr_value_add]);
19
20     if max_index == 1
21         focused_img = med_rgb_img;
22     elseif max_index == 2
23         focused_img = sub_rgb_img;
24     else
25         focused_img = add_rgb_img;
26     end
27 end
```

ПРИЛОЖЕНИЕ Б