

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Г.И. Радченко  
«\_\_» \_\_\_\_\_ 2021 г.

Разработка и исследование алгоритма сглаживания цветных изображений,  
основанного на анализе градиентов

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ

Руководитель работы,  
к.т.н., доцент каф. ЭВМ  
\_\_\_\_\_ Е.С. Ярош  
«\_\_» \_\_\_\_\_ 2021 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ И.Ю. Моисеев  
«\_\_» \_\_\_\_\_ 2021 г.

Нормоконтролер,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
«\_\_» \_\_\_\_\_ 2021 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Г.И. Радченко

« \_\_\_\_ » \_\_\_\_\_ 2021 г.

### **ЗАДАНИЕ**

**на выпускную квалификационную работу бакалавра**  
студенту группы КЭ-405  
Моисееву Илье Юрьевичу  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

1. **Тема работы:** «Разработка и исследование алгоритма сглаживания цветных изображений, основанного на анализе градиентов» утверждена приказом по университету от 26 апреля 2021 г. №714-13/12
2. **Срок сдачи студентом законченной работы:** 5 июня 2021 г.
3. **Исходные данные к работе:**
  - Gonzalez, R.C. Digital Image Processing / R.C. Gonzalez, R.E. Woods – New York: Pearson, – 2018. – 1022 p.
  - Kass, M. Analyzing oriented patterns / M. Kass, A. Witkin // Computer vision, graphics, and image processing. – 1987. – № 3(37). – P. 362–385.

- Wang, D.C. Gradient inverse weighted smoothing scheme and the evaluation of its performance / D.C. Wang, A.H. Vagnucci, C.C. Li // Computer graphics and image processing. – 1981. – № 2(15). – P. 167–181.
- Zhu, F. A benchmark for edge-preserving image smoothing / F. Zhu, Z. Liang, X. Jia, L. Zhang, Y. Yu // IEEE Transactions on Image Processing. – 2019. – № 7(28). – P. 3556–3570.
- бенчмарк для тестирования: [https://github.com/zhufeida/Benchmark\\_EPS](https://github.com/zhufeida/Benchmark_EPS).

#### 4. Перечень подлежащих разработке вопросов:

- рассмотрение существующих научных работ посвященных сглаживанию изображений;
- анализ задачи сглаживания и существующих подходов к ней;
- разработка собственного метода сглаживания изображений, использующего информацию о градиентах и позволяющего сглаживать с учетом границ объектов;
- разработка программной реализации метода;
- оценка качества работы разработанного метода, сравнение результатов с аналогами.

5. Дата выдачи задания: 1 декабря 2020 г.

Руководитель работы \_\_\_\_\_ /Е.С. Ярош/

Студент \_\_\_\_\_ /И.Ю. Мусеев/

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2021	
Разработка метода	01.04.2021	
Программная реализация	01.05.2021	
Тестирование, отладка, эксперименты	15.05.2021	
Компоновка текста работы и сдача на нормоконтроль	28.05.2021	
Подготовка презентации и доклада	30.05.2021	

Руководитель работы \_\_\_\_\_ /Е.С. Ярош/

Студент \_\_\_\_\_ /И.Ю. Мусеев/

## Аннотация

И.Ю. Моисеев. Разработка и исследование алгоритма сглаживания цветных изображений, основанного на анализе градиентов. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2021, 70 с., 19 ил., библиогр. список – 36 наим.

В рамках выпускной квалификационной работы проводится исследование алгоритма сглаживания изображений на основе анализа градиентов. Данный алгоритм производит сглаживание изображения с учетом границ, используя две компоненты векторов градиента – их длины и углы. Информация, извлеченная из данных признаков, позволяет эффективно отличать границы объектов на изображении от их текстур и сглаживать их соответственно. В работе производится общий анализ задачи и обзор существующих ее решений, описываются теоретические основы, на которых базируется предлагаемый метод, а также приводятся результаты ряда экспериментов, позволяющих сравнить данный метод с аналогами в качественном и количественном измерениях и выявляющих области, в которых предлагаемый алгоритм превосходит аналоги.

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1. ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ.....	8
2. ОБЗОР ЛИТЕРАТУРЫ.....	13
КАТЕГОРИЗАЦИЯ ПОДХОДОВ .....	13
2.1 ЯВНЫЕ МЕТОДЫ СГЛАЖИВАНИЯ .....	14
2.2 НЕЯВНЫЕ МЕТОДЫ СГЛАЖИВАНИЯ.....	19
3. ОПИСАНИЕ МЕТОДА .....	23
4. РЕЗУЛЬТАТЫ .....	29
5. СРАВНЕНИЕ КАЧЕСТВА .....	34
5.1 СРАВНЕНИЕ КАЧЕСТВЕННОЕ .....	37
5.2 СРАВНЕНИЕ КОЛИЧЕСТВЕННОЕ.....	42
5.2.1 СРАВНЕНИЕ НА ЗАДАЧЕ .....	42
5.2.2 СРАВНЕНИЕ НА БЕНЧМАРКЕ.....	51
6. ОБСУЖДЕНИЕ .....	57
6.1 ГРАНИЦЫ ПРИМЕНИМОСТИ .....	58
6.2 ДАЛЬНЕЙШИЕ РАЗРАБОТКИ .....	60
ЗАКЛЮЧЕНИЕ .....	61
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	63
ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД РЕАЛИЗАЦИИ ПРЕДЛАГАЕМОГО МЕТОДА .....	68

## ВВЕДЕНИЕ

Сглаживание изображений является фундаментальной задачей в области компьютерной графики и компьютерного зрения. Его цель – улучшение качества изображения, выявление его структуры с помощью устранения незначительных деталей, и сохранения при этом общей структуры изображения. Однако всегда присутствует необходимость идти на компромисс между этими двумя задачами: сглаживая очень легко потерять структуры, несущие информацию. Сглаживание имеет широкий спектр применений в задачах компьютерной графики и обработки изображений – это низкоуровневое преобразование, входящее практически в любой процесс обработки изображения. К сферам применения относятся задачи выделения границ, сегментации, абстракции, улучшения изображений и подобные [1–3].

В этой работе предлагается новый алгоритм для решения задач сглаживания с сохранением границ, который позволяет сохранять выраженные структуры на изображении и подавлять слабозаметные текстуры, шум невысокой интенсивности и разного рода артефакты.

Он основан на анализе двух компонент векторов градиента, отражающих изменение интенсивности цвета в окрестности определенной точки. Эти компоненты – длина и угол наклона вектора градиента. Метод имеет явно выраженное правило для формирования маски весов. Предполагается, что близость углов градиента в точках окрестности говорит о принадлежности двух точек к одной границе, а значит, при сглаживании они должны иметь больший вес. Также принимается во внимание обратное значение длины вектора градиента как фактор формирования веса.

Целью данной работы является разработка алгоритма сглаживания цветных изображений для задачи сглаживания с учетом границ на основе анализа компонент градиента.

В рамках данной работы поставлены задачи: провести обзор родственных разработок – работ, предлагающих решения задачи сглаживания с учетом границ; разработать такой дескриптор, который позволил бы настраивать веса фильтра в зависимости от вида границы в окрестности; провести эксперименты, определив тем самым область применимости метода.

К методам исследования относятся: проведение обзора, постановка вычислительных экспериментов и оценка их результатов количественная и качественная.

## **1. ПОСТАНОВКА И АНАЛИЗ ЗАДАЧИ**

Сглаживание изображений берет свое начало в области сглаживания сигналов. Сгладить сигнал – значит устранить незначительные колебания в виде шума и незначительных флуктуаций, чтобы оставить только сигнал, несущий передаваемую информацию.

В терминах сглаживания сигналов задача сглаживания изображения может формулироваться как сглаживание дискретной функции двух переменных (хотя допустимо и использование предположения непрерывности). Она может формулироваться также в области окрестностей пикселей, в пространстве изображения или быть переведена в частотную область преобразованием Фурье.

Вне зависимости от пространства, в котором производится решение задачи сглаживания, основной компромисс, на который приходится идти при ее решении. – это противоречие между сглаживанием незначительных деталей и сохранением сигнала. Можно сгладить детали с максимальной силой, но при

этом потерять ценную информацию, которую несет в себе сигнал и наоборот, можно пытаться максимально сохранить сигнал, но при этом плохо сгладить детали.

Основной информацией на изображении, или «сигналом» являются крупные объекты, их положение, границы и цвет. Эта информация играет ключевую роль в процессе последующей обработки изображений, будь то извлечение границ, сегментация или даже просто просмотр изображения человеком. Следовательно, сглаживание с учетом границ должно стремиться сохранять границы и цветовую составляющую значимых объектов на изображении и в то же время сглаживать текстуры вне и внутри объектов и незначительные детали.

Алгоритм, представленный в данной работе, был разработан для решения задачи сглаживания цветных изображений с учетом границ. Существуют также другие, более специфические задачи сглаживания, которые остаются за границами этой работы. Это, например, задача сглаживания карт глубин [4]. Карты глубин из-за специфики процесса получения состоят из константных областей и резких перепадов яркости. Некоторые перепады означают границы между объектами, а некоторые образуются на рельефе самого объекта. Соответственно «сигналом» в такой задаче являются объекты, их границы и относительная глубина, а сгладить необходимо резкие ступенчатые перепады на самих объектах. Такая задача не может быть решена обычными алгоритмами для сглаживания цветных изображений. Еще одним примером может служить задача сглаживания поля потоков при обработке изображений отпечатков пальцев. Направления градиентов в каждой из точек изображения отпечатка пальца могут дать большое количество информации о рисунке отпечатка. Учитывая, что градиент сам по себе является дескриптором высокочувствительным к шуму, и, следовательно, порождающим шум, значения

градиента необходимо сгладить каким-то образом. Для решения такой задачи используются специальные алгоритмы сглаживания поля потоков. Пример усреднения градиентов при работе с полем потоков приведен не зря: в основу описываемого подхода лег принцип усреднения градиентов, использующийся в задачах подобного рода. Более подробно этот принцип будет рассмотрен в разделе с описанием метода.

Как уже было упомянуто, задача сглаживания решается в большом количестве сценариев обработки изображения. Наиболее распространенная цель использования сглаживания это удаление слабого шума. На всех этапах обработки изображения начиная с его получения с помощью камеры, заканчивая многократным сжатием и передачей по сети, изображение подвергается воздействию различного вида шума. Такой шум (особенно на изображениях высокого разрешения) может быть невиден невооруженным глазом, но при этом оказывать влияние на различные алгоритмы обработки изображения. Например, процедура извлечения границ объектов – оператор Кэнни [5] включает в себя обработку изображения гауссовым фильтром перед извлечением границ. Такая процедура очень чувствительна к шуму – перепады яркости могут порождать ложные границы в силу локальной природы оператора, а, следовательно, требует предварительного сглаживания для получения более точных границ. Аналогичная ситуация возникает при решении задачи сегментации изображения – шум может порождать ложные сегменты либо изменять форму истинных. Здесь сглаживание может быть применено, например, перед бинаризацией по пороговому значению, устраняя возможные ложные срабатывания вызванные шумом. В целом задачи выделения границ и сегментации являются лучшим, но не единственным применением алгоритмов сглаживания, учитывающих границы. Как уже упоминалось ранее, изображение перед передачей его по каналам связи зачастую множество раз подвергается

операциям сжатия, которое, в свою очередь, порождает артефакты различного рода. Для улучшения таких, подверженных артефактам изображений также могут быть использованы алгоритмы сглаживания.

На данный момент существует большое множество алгоритмов сглаживания, но является ли задача сглаживания решенной?

Во-первых, стоит взглянуть на саму постановку задачи – задача является некорректно поставленной. Существует три критерия корректности постановки задачи, установленные Жаком Адамаром [6]: ее решение существует, оно единственно и устойчиво. Для задачи сглаживания цифрового изображения как дискретной функции двух переменных, как минимум не выполняется второе условие: для любого изображения существует бесконечное множество его сглаженных версий. Соответственно, сама задача не может быть решена окончательно, и число способов, которыми можно ее решить, бесконечно. Помимо этого, некорректность постановки порождает другую проблему – как сравнить между собой разные алгоритмы сглаживания? Можно было бы разработать численные метрики (так же, как это делается, например, в задачах удаления шума), если бы решение было известно, то есть если бы существовало идеально сглаженное изображение, с которым можно было бы сравнивать результаты различных подходов. Но его не существует, и это значит, что исследователям в области сглаживания приходится опираться на экспертную оценку (что чаще всего и происходит) или на то, как качественно работают алгоритмы более высокого уровня, опирающиеся на сглаживание, результаты которых возможно численно оценить. Сами критерии для оценки качества сглаживания с учетом границ можно было бы задать следующим образом: на выходе должно получиться изображение, в котором для значимых объектов точно сохранены их границы, и при этом мелкие объекты и текстуры сглажены.

Во-вторых, можно рассмотреть существующие на данный момент решения задачи сглаживания. Каковы недостатки этих решений, исходя из критериев оценки качества сглаживания?

Во-первых, высокая вычислительная сложность. Даже в современных условиях, когда высокопроизводительная техника становится все более доступной, не уменьшается роль экономичных оптимизированных вычислений. В особенности это значимо для программ, выполняющихся на встроенных или конечных устройствах, где условия не позволяют иметь высокую вычислительную мощность. Например, для работы в пайплайнах вычислительной фотографии, алгоритмы сглаживания должны выполняться за приемлемое для таких задач время, но ситуация на сегодняшний день такова, что имеется крайне мало высокоскоростных и качественных решений задачи сглаживания.

Во-вторых, неэффективное сглаживание текстур. Алгоритмы сглаживания, основанные на локальных признаках, имеют чаще всего подобную особенность. Границы объектов могут оставаться сохраненными, но и текстуры сохраняются вместе с ними. Возможны также ситуации, когда в отдельных регионах текстура сглаживается, и детали исчезают, а в отдельных нет, как, например, у подходов, основанных на применении нейронных сетей.

Однако возможна иная ситуация. Текстуры могут сглаживаться эффективно, но при этом размываться границы объектов. Такая проблема зачастую характерна для алгоритмов, веса, маски которых не формируются адаптивно, но фиксированы. Такие алгоритмы, как правило, быстрее, но склонны к подобным проблемам, не различая текстуры и структуры на изображении.

В-третьих, неточная локализация границ после сглаживания. Многие алгоритмы, показывающие при зрительной оценке высокие результаты

сглаживания (хорошо сглаженные текстуры и сохраненные границы), могут иметь склонность смещать границу объекта так, что ее локализация будет хоть и возможной, но неточной, либо при слишком сильном сглаживании границы ее локализация может быть затруднена. Такое свойство может оказывать сильное влияние на алгоритмы выделения границ и сегментации.

Уточним теперь, пользуясь вышеописанным, определение задачи сглаживания изображения с учетом границ. Сглаживание изображения с учетом границ – операция преобразования входного изображения в другое, называемое выходным или сглаженным. Такая операция должна сохранять границы объектов на сглаженном изображении и в то же время сглаживать текстуры и незначительные детали. То есть в границах объектов должна уменьшаться вариация цветов, при этом сами границы должны оставаться нетронутыми.

Подводя итог, задачу сглаживания в целом можно описать как математически некорректно поставленную, но, несмотря на это, хорошо разработанную в силу своей широкой применимости. Алгоритмы сглаживания в целом обладают рядом недостатков [7], которыми можно пренебрегать в зависимости от тех или иных условий приложения.

## **2. ОБЗОР ЛИТЕРАТУРЫ**

В данном разделе проводится обзор работ из области сглаживания с учетом границ. В разделе проводится разделение различных подходов в области на две большие категории. В нем будут рассмотрены наиболее зарекомендовавшие себя подходы к задаче, их достоинства и недостатки.

### **КАТЕГОРИЗАЦИЯ ПОДХОДОВ**

В зависимости от рассмотрения задачи подходы к сглаживанию изображений с учетом границ делятся на явные и неявные [8]. *Явные методы* –

это такие, в которых правило сглаживания задается явным образом. Чаще всего они формулируются с помощью пространственной свертки, но существуют и исключения: к ним относится, например фильтр, использующий взвешенную медиану. *Неявные методы* – такие, которые чаще всего формируются в виде задачи минимизации (взвешенной) суммы двух слагаемых: слагаемого данных и слагаемого сглаживания. Такая формулировка дается им также не всегда: методы, основанные на машинном обучении, также полагаются неявными.

## 2.1 ЯВНЫЕ МЕТОДЫ СГЛАЖИВАНИЯ

Методы данной категории являются как исторически первыми, так и менее затратными вычислительно, а потому более распространенными. Как уже было замечено ранее, чаще всего эти фильтры формулируются как пространственная свертка.

Двумерную пространственную свертку изображения  $f$  ядром  $w$  можно описать выражением:

$$(w * f)(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t),$$

где  $x, y$  – пространственные координаты изображения;  $s, t$  – пространственные координаты ядра,  $a, b$  – половина высоты и ширины изображения в пикселях соответственно.

Это выражение является взвешенной суммой значений пикселей окрестности и может также называться линейной пространственной фильтрацией [9]. Исходное изображение расширяется (классический подход – нулями) и для каждого пикселя вычисляется указанная взвешенная сумма. Важно отметить, что веса  $w(s, t)$  положительны и в сумме дают единицу, а высота и ширина изображения предполагаются четными числами.

Самые простые из явных подходов, будучи наиболее быстрыми, часто приводят к потере деталей изображения, размытию структуры. Примером здесь может служить наиболее часто применяющийся в стадии предобработки гауссов фильтр.

Перечисляя подходы к сглаживанию, нельзя не упомянуть билатеральный фильтр, который был представлен в 1998 году в работе С. Tomasi и R. Manduchi [10]. Этот фильтр заменяет значение каждого пикселя средним значением среди наиболее похожих и ближайших пикселей. Для достижения данного эффекта используется комбинация пространственного и яркостного ядра. Это позволяет ему сглаживать небольшие колебания цвета в окрестности пикселя, но при этом сохранять жесткие переходы контуров, так как отличные друг от друга пиксели практически не влияют друг на друга. Билатеральный фильтр выражается формулой:

$$BLF(I) = \frac{1}{\sum_{q \in N(p)} W_{BLF_{pq}}(I)} \sum_{q \in N(p)} W_{BL_{pq}}(I) I_q,$$

где функция весов  $W_{BL_{pq}}(I)$  выражается как:

$$W_{BLF_{pq}}(I) = \exp\left(\frac{-\|p - q\|^2}{2\sigma_s^2}\right) \exp\left(\frac{-|I_p - I_q|^2}{2\sigma_r^2}\right).$$

Стоит отметить, что формула билатерального фильтра является ничем иным, как упрощенной записью формулы двумерной пространственной свертки, где  $p$  – центральный пиксель окрестности  $N$ ,  $q$  – текущий пиксель.

Первый множитель является функцией расстояния от пикселя  $p$  до  $q$ , где пиксель  $p$  лежит в центре окна фильтра. В то время как второй множитель – функция разности значений интенсивности пикселей. Обе они выражаются гауссовой функцией. Первый множитель дает больший вес пикселям, которые находятся близко к исходному пикселю  $p$ , а второй пикселям, которые близки к

нему по интенсивности. Параметры  $\sigma_s$  и  $\sigma_r$  позволяют регулировать форму двух гауссианов, а именно «тяжесть» их хвостов. Большие значения будут означать более тяжелые хвосты, что в свою очередь означает большую восприимчивость к отличающимся значениям и соответственно больший уровень сглаживания.

Несмотря на простоту формулировки, алгоритм является довольно сложным вычислительно. Чтобы решить эту проблему, было разработано множество подходов. Среди них можно выделить реализацию, в которой авторы достигли сложности  $O(1)$  с помощью тригонометрических ядер [11]. Благодаря своей фундаментальной природе, билатеральный фильтр стал фундаментом для дальнейшего развития, продуктом которого стал совместный билатеральный фильтр (joint bilateral filter) [12]. Его вклад заключается в использовании изображения дескриптора текстур вместо изначального изображения при вычислении пространственного ядра. Еще одним из расширений идеи билатерального фильтра стал направляемый фильтр (guided filter) – метод сглаживания изображений, подразумевающий использование направляющего изображения, был предложен в 2013 году [13]. Его функция может быть явно описана формулой подобной той, которая уже была приведена при рассмотрении билатерального фильтра. Приведем только формулу расчета весов фильтра, определив направляющее изображение  $G$  и входное изображение  $I$ .  $G_p$  – значение интенсивности пикселя  $p$ .

$$W_{GI_{pq}}(G) = \frac{1}{|\omega|^2} \sum_{k:(p,q) \in N(p)} \left( 1 + \frac{(G_p - \mu_p)(G_q - \mu_p)}{\sigma_p^2 + \epsilon} \right)$$

Здесь  $\mu_p$  – среднее значение пикселей окрестности,  $\sigma_p^2$  – дисперсия,  $\epsilon$  – параметр, позволяющий регулировать сглаживание, а  $|\omega|^2$  – коэффициент для нормировки. Идея этого фильтра подобна той, что используется в совместном билатеральном фильтре, но она основана на другой модели. Авторы полагают

локальную линейную зависимость между входным и выходным изображениями, из чего в итоге выводят функцию весов.

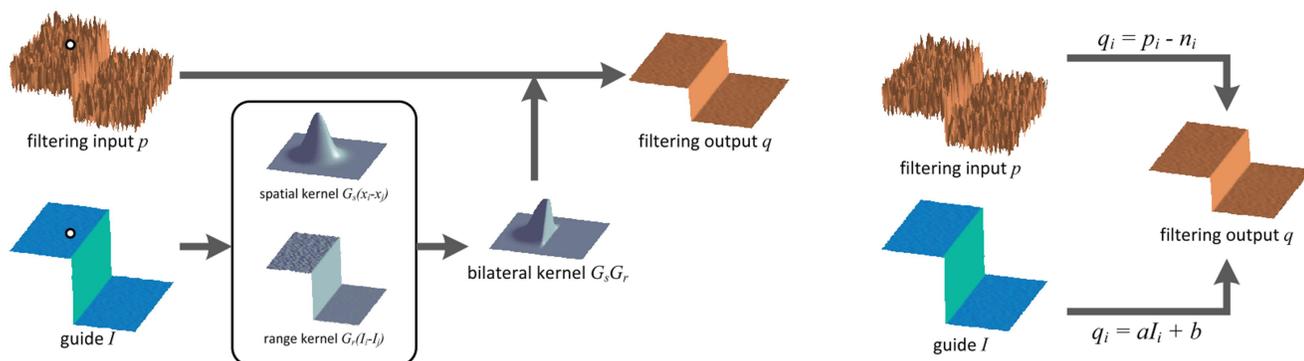


Рисунок 1 – Схема работы билатерального фильтра (слева) и принцип работы направляемого фильтра (справа).

Немного более необычным может показаться следующий пример, но, несмотря на это, можно отнести его к категории явных. Скользящий направляемый фильтр (rolling guidance filter) можно описать как процесс, состоящий из двух шагов: размытие текстур с помощью гауссовой функции, а затем итеративное восстановление структуры [14].

Функция для первого шага выражается как:

$$G(p) = \frac{1}{K} \sum_{q \in N(p)} \exp\left(\frac{-\|p - q\|^2}{2\sigma_s^2}\right) I(p),$$

$$K = \sum_{q \in N(p)} \exp\left(\frac{-\|p - q\|^2}{2\sigma_s^2}\right).$$

После обработки гауссовой функцией происходит итеративный процесс восстановления структуры изображения, который является основной частью предлагаемого метода. Исследователи заметили, что гауссов фильтр размывает слабые и небольшие изменения в яркости, но большие структуры сохраняются в размытом состоянии. Задача состоит в том, чтобы восстановить не до конца

утерянные крупные детали изображения. Процесс функционирования метода показан на рисунке 2.

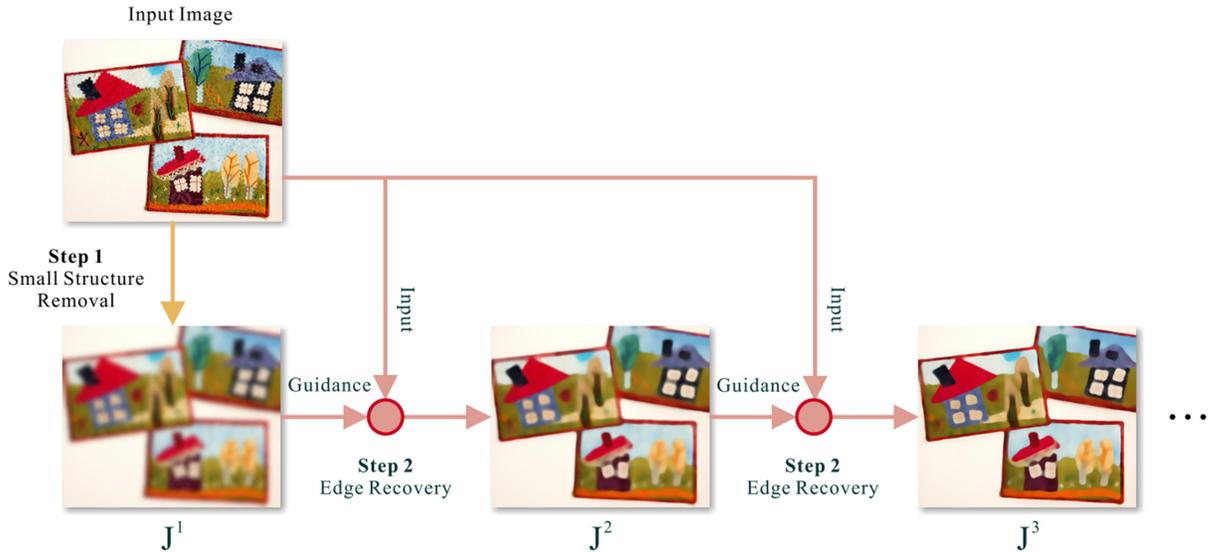


Рисунок 2 – Схема работы скользящего фильтра

Для восстановления структуры используется итеративный процесс, который можно описать формулой совместного билатерального фильтра, использующего два изображения:  $J^t$  – изображения, полученного на  $t$ -й итерации и  $I$  – входного изображения.

$$J^{t+1}(p) = \frac{1}{K_p} \sum_{q \in N(p)} \exp\left(-\frac{\|p - q\|^2}{2\sigma_s^2} - \frac{\|J^t(p) - J^t(q)\|^2}{2\sigma_r^2}\right) I(q),$$

$$K_p = \sum_{q \in N(p)} \exp\left(-\frac{\|p - q\|^2}{2\sigma_s^2} - \frac{\|J^t(p) - J^t(q)\|^2}{2\sigma_r^2}\right).$$

В качестве дополнения идеи скользящего направляемого фильтра была разработана анизотропная его версия, улучшающая качество работы [15]. Анизотропии добиваются, обобщая исходные гауссовы функции. Функция расстояния дорабатывается с помощью специальной матрицы, детали получения которой выходят за рамки данного обзора. Прodelывая такие преобразования, авторы утверждают, что в итоге достигается наиболее высокое

качество сглаживания, чем при использовании (изотропного) скользящего фильтра.

## 2.2 НЕЯВНЫЕ МЕТОДЫ СГЛАЖИВАНИЯ

Эта группа методов отличается многообразием формулировок подходов и зачастую большой вычислительной сложностью. Задача чаще всего формулируется в виде минимизации суммы [8]:

$$E(S) = E_d(S, I) + \lambda E_v(S),$$

где  $E_d(S, I)$  – компонента данных, мера разницы между исходным изображением  $I$  и тем, что получено на выходе  $S$ ;

$E_v(S)$  – компонента сглаживания, которая заставляет снижать вариабельность цветов в окрестности пикселя;

$\lambda$  – параметр, балансирующий эти два слагаемых.

Функция для минимизации формулируется так в общем виде, но часто подвергается модификациям, добавлению новых компонент и т.д. Здесь указана общая формулировка, но к неявным методам предлагается также относить методы, использующие машинное обучение.

Описывая метод сглаживания, основанный на минимизации  $L_0$  градиента, исследователи предлагают разреженную схему подсчета градиентов в оптимизационном фреймворке [16]. Основное нововведение подхода – это новая стратегия, подсчитывающая дискретное число изменений интенсивности цвета в окрестности пикселя, что, как указывают авторы, математически ведет к норме  $L_0$ . При использовании метода стремятся сделать выделяющиеся границы тоньше, что улучшает их различимость. Сформулируем задачу минимизации, предложенную в данной работе, в двумерном виде. Мера для градиентов выражается как:

$$C(S) = \#\{p \mid |\partial_x S_p| + |\partial_y S_p| \neq 0\},$$

где  $\#\{\cdot\}$  – оператор, возвращающий количество элементов, для которых выполняется условие в скобках. В данном случае он подсчитывает число всех пикселей, для которых градиенты по осям  $x$  и  $y$  не равны нулю.

При этом всем задача минимизации формулируется как:

$$\min_S \left\{ \sum_p (S_p - I_p)^2 + \lambda C(S) \right\}.$$

Форма функции для минимизации напоминает уже описанную в начале раздела. Две компоненты: первая отвечает за сохранение структуры, а вторая за сглаживание текстур. Задача минимизации этой функции нетривиальна, и не поддаётся решению классическими градиентными методами, так как сама функция содержит параметр  $C(S)$ , являющийся дискретной недифференцируемой функцией. Исходя из этого, авторы подхода разработали собственный алгоритм для решения данной задачи, который предполагает введение новых переменных в целевую функцию и нахождение результатов нескольких подзадач.

Среди последних разработок популярностью пользуются подходы, использующие машинное обучение, позволяющие достигать результатов, сопоставимых с результатами традиционных методов. Одной из проблем использования машинного обучения в задаче сглаживания является некорректность постановки задачи, а если быть более точным – отсутствие решения. Алгоритмам машинного обучения требуются входные данные, истинные значения (в данном случае сглаженные изображения) и определенная функция потерь. Для задачи сглаживания существуют некоторые проблемы при определении последних двух. В первом случае требуется задать эталонные образцы сглаживания, что само по себе является нетривиальной задачей, а во

втором случае проблема, как правило, заключается в том, чтобы функция потерь отражала именно критерии сглаживания. То есть, чтобы изображение с меньшим значением этой функции было сглаженным более качественно, чем изображение, у которого это значение больше.

Одно из решений приведенных проблем описано в работе [17]. В ней решение для задачи сглаживания определяется как взвешенная сумма наиболее выбираемых добровольцами результатов. Для решения задачи в качестве модели была предложена очень глубокая сверточная сеть (Very Deep Convolutional Neural Network – VDCNN). Были вручную выбраны 500 изображений из двух наборов данных, один из которых ранее использовался для сравнения моделей качества изображений [18], а другой в задаче сегментации [19]. Чтобы определить наилучшие для сглаживания методы, для тестирования были взяты наиболее значимые подходы из области сглаживания, ими обработали изображения и предложили добровольцам выбрать лучшие варианты. Выбирать предлагалось как среди разных подходов, так и среди различных параметров в рамках одного подхода. В итоге на основе полученных данных была определена функция потерь, являющаяся суммой трех слагаемых. Первое и второе представляют собой L1 и L2 нормы между эталонным и предсказанным изображениями, а третье измеряет локальные колебания интенсивности в окрестности точки. Эталонным изображением было выбрано не одно, а пять лучших, чтобы не использовать только один метод, а взять что-то от каждого часто выбираемого. Соответственно, каждое из слагаемых является взвешенной суммой пяти лучших результатов. Используя полученные данные и выведенную функцию потерь, авторы, обучив модель, получают качественные результаты при меньшем времени работы, если запускать модель на видеокарте. Другим примером решения задачи с помощью машинного обучения может служить работа [20]. Авторы в ней решают приведенные

проблемы иным путем: они предполагают, что текстуры на изображении могут быть и высоких амплитуд и различных масштабов. Это предположение само по себе нетипично для сглаживания, где, как правило, предполагается обратное. Исходя из этого предположения, авторы предлагают подход, основанный на разделении текстур и структур на изображении. Для достижения этой цели необходимо получить данные для обучения, что само по себе является нетривиальной задачей: разметка текстур и структур на изображении очень трудоемка, учитывая также точность и объемы данных, которые требуются для эффективного обучения модели. Авторы пошли другим путем: они объединили изображения, на которых присутствует только текстуры с изображениями, на которых присутствуют только структуры. Такими изображениями оказались мультипликационные кадры, на которых объекты представляют собой контуры и сплошные цвета. Три различных сверточных сети использовались при решении задачи. Structure Prediction Network (SPN) – сеть предсказания структуры изображения; Texture Prediction Network (TPN) – сеть предсказания текстур на изображении; Texture and Structure Aware Filtering (TSAFN) – сеть, производящая сглаживание с учетом текстуры и структуры, полученных на предыдущих шагах. В итоге получился подход, реализующий сглаживание на уровне, который далеко не всегда достижим с помощью традиционных подходов. Хотя его результаты зачастую непредсказуемы и непохожи на результаты других методов. В отличие от методов, рассмотренных ранее, методы, использующие обучение без учителя, решают проблему того, что решение не определено другим способом. Они не нуждаются в размеченных данных [21]. Для достижения требуемых свойств, тем не менее, необходимо определить функцию потерь, то есть критерии сглаживания. Целевая функция формулируется как взвешенная сумма компоненты данных, компоненты сглаживания и компоненты сохранения границ. Первая компонента выражается

квадратом  $L_2$  нормы между исходным и сглаженным изображением. Вторая выражается как взвешенная сумма  $L_p$  норм пикселей окрестности, где веса рассчитываются подобно весам билатерального фильтра. Третья выражается как  $L_2$  норма разности отклика градиентов исходного и сглаженного изображения. При этом берутся не все градиенты, а только те, которые принадлежат контурам, которые были найдены с помощью детектора контуров. Данная архитектура позволяет решать задачи не только сглаживания, но и, например, генерации изображений, имитирующих карандашный рисунок или задачи улучшения изображения.

Таким образом, задача сглаживания изображений не является новой и за многие годы существования стала неплохо разработанной. Она имеет устоявшиеся подходы и продолжает развиваться до сих пор.

### **3. ОПИСАНИЕ МЕТОДА**

Как было указано выше, предлагаемый метод принадлежит к категории явных. Он подобно другим классическим методам использует пространственную свертку, механизм которой уже был упомянут в данной работе. Исходя из этого, для формулирования описываемого фильтра необходимо задать правило, по которому будут рассчитываться веса. Но перед этим необходимо объяснить теорию, которая стоит за правилом их формирования. Стоит отметить, что само правило формирования весов, по сути, является ключевым вкладом данной разработки.

При формировании предлагаемого подхода рассматриваются два типа границ объектов, которые могут попадать в окрестность точки: это регулярные и нерегулярные границы. Пример, поясняющий эти две концепции, показан на рисунке 3.

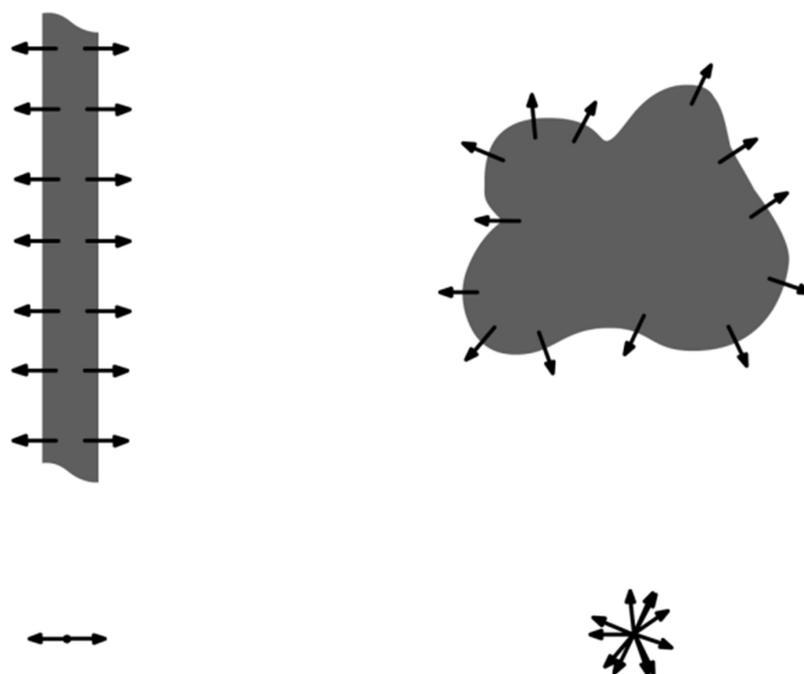


Рисунок 3 – Два типа границ в окрестности точки и их градиенты

Предполагается, что регулярные границы принадлежат границам объектов, а нерегулярные принадлежат текстурам или объектам незначительным. Исходя из этого предположения, веса должны формироваться так, чтобы сглаживать эти два типа границ по-разному: сохраняя регулярные границы объектов и сглаживая нерегулярные границы текстур. Стоит отметить, что на разных масштабах одни и те же границы могут восприниматься как границы разных типов. Эта особенность классификации границ позволяет определять пространственный масштаб деталей, которые признаются незначительными.

Для того, чтобы отличить один тип границ от другого, в данном подходе используется информация о векторах градиента. Градиент на цифровом изображении  $f$  рассчитывается для каждой точки с координатами  $(x, y)$  как двумерный вектор столбец [9]:

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}.$$

Такой вектор имеет своей особенностью то, что своим направлением он указывает в каждой точке на направление наибольшего изменения интенсивности, а длиной на силу этого изменения. Для простой окрестности в цифровом изображении как на рисунке 4 градиент будет рассчитываться приближенно как конечные разности. Формула градиента будет выглядеть следующим образом.

$$\nabla f \approx \begin{bmatrix} z_6 - z_4 \\ z_8 - z_2 \end{bmatrix}$$

В рассматриваемом подходе приведенные особенности векторов градиента используются для работы с границами объектов на изображении.

$z_1$	$z_2$	$z_3$
$z_4$	$z_5$	$z_6$
$z_7$	$z_8$	$z_9$

Рисунок 4 – Простая окрестность пикселя

При этом направления векторов градиента рассчитываются по следующей формуле.

$$\vartheta = \arctan(g_y/g_x)$$

А длины векторов по следующей формуле.

$$\alpha = \sqrt{g_x^2 + g_y^2}$$

Рассмотрим направления векторов градиентов, расположенных на границах объектов. Градиенты на границах на рисунке 3 изображены в двух видах: на самом изображении и сведенные в одну точку. Можно заметить, насколько сильно отличаются распределения направлений градиентов относительно друг друга у двух типов границ. Регулярные границы порождают окрестность такую, в которой направления градиентов распределены с минимальным отклонением друг от друга. Нерегулярные же отличаются хаотичным распределением направлений. При этом не всякие распределения направлений градиентов указывают на границы объектов. Чтобы определить выделяющуюся границу, используются также длины векторов градиента, показывающие силу перехода, а значит и предположительно наличие границы. При формировании весов в данном подходе используется именно эти свойства векторов градиента.

Если просуммировать все, что было дано ранее, веса для фильтра рассчитываются таким образом, чтобы при применении он позволял сглаживать незначительные детали и оставлять границы объектов и общую структуру изображения нетронутыми посредством учета границ на изображении. Для достижения этих свойств используются углы градиента и его длины, которые, по предположению, отличаются для тех областей, которые необходимо сгладить и тех, которые необходимо сохранить. Используя все выше сказанное, можно перейти к математическому формулированию метода.

Во-первых, чтобы сформулировать подход с описанными ранее свойствами, необходимо рассчитать, насколько углы градиентов в пикселях  $p$  и  $q$  близки друг к другу, то есть разницу между углами градиентов в центральном пикселе и пикселе окрестности. Близость углов рассчитывается как косинус удвоенного угла между векторами. При расчете углов векторов градиентов получаются значения от  $-\pi/2$  до  $\pi/2$ . При этом удвоенная разница между

векторами будет лежать между  $-2\pi$  и  $2\pi$ . Разность углов удваивается для того, чтобы сблизить углы, разница между которыми близка к  $\pi$ . Это делается из предположения, что такие векторы принадлежат к одной границе – один находится с одной стороны от границы, а другой соответственно с противоположной. Они направлены в разные стороны, но принадлежат одной границе, а значит, не должны взаимно уничтожать друг друга при усреднении. Пример можно увидеть на рисунке 1 у левой границы, но подробнее процесс удвоения углов продемонстрирован на рисунке 5.

Градиенты в окрестности    Разница углов    Разница углов после удвоения

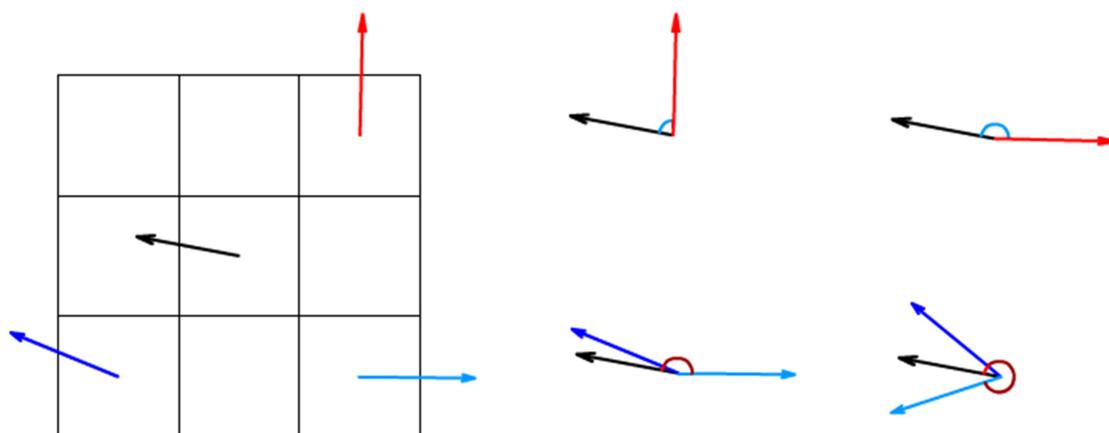


Рисунок 5 – Причина удвоения градиентов

Видно, как углы градиентов некоторой окрестности, изображенные голубым, после удвоения становятся ближе друг к другу, а разница для тех градиентов, которые по предположению не принадлежат той же границе, остается большой. Идея использования удвоенной разности углов векторов не нова и, более того, имеет широкое применение в другой области обработки изображений: распознавании отпечатков пальцев. При оценке поля потоков изображения отпечатка пальца необходимо выполнить усреднение направлений градиентов в окрестности. Учитывая, что изображение отпечатка пальца

состоит из узора чередующихся линий, на краях каждой из линий существуют градиенты двух противоположных направлений, которые при суммировании взаимно уничтожатся, давая нулевые значения углов там, где на самом деле существует линия. Чтобы избежать этого эффекта, было предложено использовать удвоенные значения углов вместо обычных [22].

Помимо направлений векторов градиента, как уже было упомянуто, используются также обратные значения их длин. Предполагается, что пиксели с большим значением модуля градиента должны сглаживаться в меньшей степени, так как они представляют собой границы объектов. Пиксели с меньшими значениями скорее всего принадлежат константным областям, которые необходимо сгладить. Таким образом, чтобы достичь требуемых свойств, в расчете весов также принимает участие компонента градиента в виде обратного значения длины. Обратное значение градиента уже применялось для сглаживания [23]. Подход, описанный в данной работе, используется для сглаживания изображения путем использования в качестве весов маски нормализованные значения обратного градиента.

Теперь можно определить формулу для формирования весов  $w(p, q)$ :

$$w(p, q) = (\cos 2(\vartheta_p - \vartheta_q) + 1) / \alpha,$$

где  $\alpha$  – это модуль градиента в точке  $q$ , а  $\vartheta_p$  и  $\vartheta_q$  – углы векторов градиентов в точках  $p$  и  $q$ . К значениям косинуса необходимо прибавлять единицу, чтобы соблюсти требование о положительных весах для взвешенной суммы.

Таким образом, было получено правило для формирования весов адаптивного фильтра, которое позволяет учитывать направления градиентов в окрестности пикселя, что, в свою очередь, дает описываемому подходу все его свойства. Такое правило позволяет производить сглаживание с учетом границ по-разному для регулярных и нерегулярных границ – для текстур и структур на

изображении. Результаты, демонстрирующие описанные свойства приводятся в следующем разделе.

## 4. РЕЗУЛЬТАТЫ

Для получения экспериментальных результатов, приведенных в следующем разделе, метод был реализован с использованием языка C++. Исходный код реализации приведен в листинге А.1 приложения А.

Реализация по умолчанию производит сглаживание как трехканальных, так и одноканальных изображений. Сглаживание цветных изображений в данной реализации происходит аналогично сглаживанию одноканальных: три цветовых плоскости сглаживаются независимо как одноканальные изображения. Сами изображения представлены в виде массивов однобайтового беззнакового целочисленного типа и загружаются при помощи библиотеки `opencv` версии 4.3.0. Программная реализация основана на шаблонах и может обеспечивать работу с типами данных различной ширины. Для работы самого подхода требуются числа с плавающей точкой, но выходное изображение все равно приводится к диапазону  $[0, 255]$  в целых числах. Из особенностей реализации стоит отметить, что если в окрестности пикселя не меняется интенсивность цвета, то его значение остается без изменений.

Продемонстрируем результаты работы приведенного фильтра на одном из классических изображений. Метод предусматривает использование по умолчанию лишь одного параметра, который влияет на сглаживание – размера ядра. Это объединяет его со многими подобными методами, в которых за основу берется свертка. Увеличение размера окна свертки приводит к усилению сглаживания при константном размере изображения, но после определенного момента качество начинает падать. Больше деталей входит в окно, следовательно, больше деталей влияет на результирующее значение каждого

пикселя, что при определенном размере окна по отношению к деталям может привести к ухудшению качества.

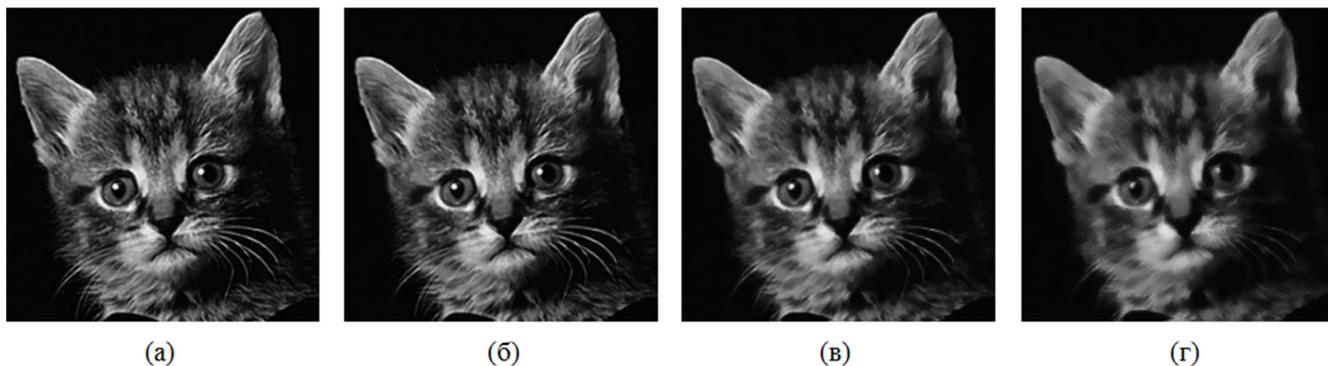


Рисунок 6 – Результат сглаживания при изменяющемся размере окна

На рисунке 6 представлены результаты сглаживания одноканального изображения с размерами  $241 \times 241$  пиксель с помощью предлагаемого подхода и различных значениях параметра размера окна. Приводить размер исходного изображения важно, так как это позволяет оценить размер сглаживаемых деталей – алгоритмы сглаживания в основной массе очень чувствительны к изменению масштаба деталей, а, следовательно, и размеров изображения. Исходя из этого, здесь и далее размеры исходных изображений будут приводиться. На рисунке 6 под буквой (а) – исходное. Второе, под буквой (б) сглажено с помощью окна  $3 \times 3$ , третье, под буквой (в) с помощью окна  $5 \times 5$ , а четвертое  $7 \times 7$ . Отметим сначала свойства описываемого подхода на примере этих результатов. Особенно ярко они заметны на примере последнего изображения. Можно заметить, как проявляются заявленные ранее свойства: мелкие детали на изображении сглаживаются в том время как четко выраженные границы остаются явными. Видна разница между тем, как сглаживаются границы с более однородным распределением градиентов и с менее однородным. Теперь можно обсудить влияние растущего размера окна на сглаживание. Здесь видно, что сила сглаживания растет с размером окна вместе с масштабом деталей, которые сохраняются. Можно увидеть, как некоторые

детали, присутствующие в изображении, сглаженном ядром с размером  $5 \times 5$ , перестают существовать на результате сглаживания ядром  $7 \times 7$ .

Результаты сглаживания одноканального изображения не могут дать полной картины для описания алгоритма сглаживания цветных изображений. Как уже упоминалось, для сглаживания цветного изображения цветовые плоскости сглаживаются независимо, что, в свою очередь, позволяет более точно учитывать границы предметов определенных цветов, но и приводит к определенным последствиям, которые будут описаны позднее. На рисунке 7 приведено множество изображений аналогичное тому, что было на рисунке 6 – под буквой (а) оригинал и под буквами (б), (в) и (г) изображения, сглаженные с ядрами  $3 \times 3$ ,  $5 \times 5$  и  $7 \times 7$  соответственно. Исходное изображение размером  $640 \times 426$  пикселей не является классическим в практике обработки изображений в отличие от первого, но было выбрано по другой причине. Оно было выбрано как содержащее множество деталей различных масштабов и форм, а также как содержащее детали различных цветов, чтобы можно было оценить качество сглаживания именно цветных изображений. Помимо уже отмеченных на предыдущем примере особенностей сглаживания можно увидеть цветовые особенности сглаживания данным методом. На глобальном уровне распределение цветов не меняется, но локально происходят небольшие изменения, которые видны при увеличении силы сглаживания. Причины таких особенностей будут обсуждаться в последующих разделах данной работы.



(а)



(б)



(в)



(г)

Рисунок 7 – Сглаживание цветного изображения

В целом же результаты работы данного алгоритма на цветных изображениях не сильно отличаются от результатов на черно-белых: можно видеть все те же самые свойства. Помимо однократного применения фильтра возможен также сценарий многократного его применения с теми же параметрами окна. Такой подход характерен для подобных методов. В работе [10] авторы предполагают итеративное использование билатерального фильтра для усиления сглаживания. Точно так же в более родственной работе [23], в которой используются значения обратных градиентов для сглаживания с учетом границ, для улучшения качества сглаживания используется многократное повторение сглаживания окном  $3 \times 3$ . Рассмотрим результат последовательного

многократного сглаживания с использованием уже приведенных изображений и размеров окон.

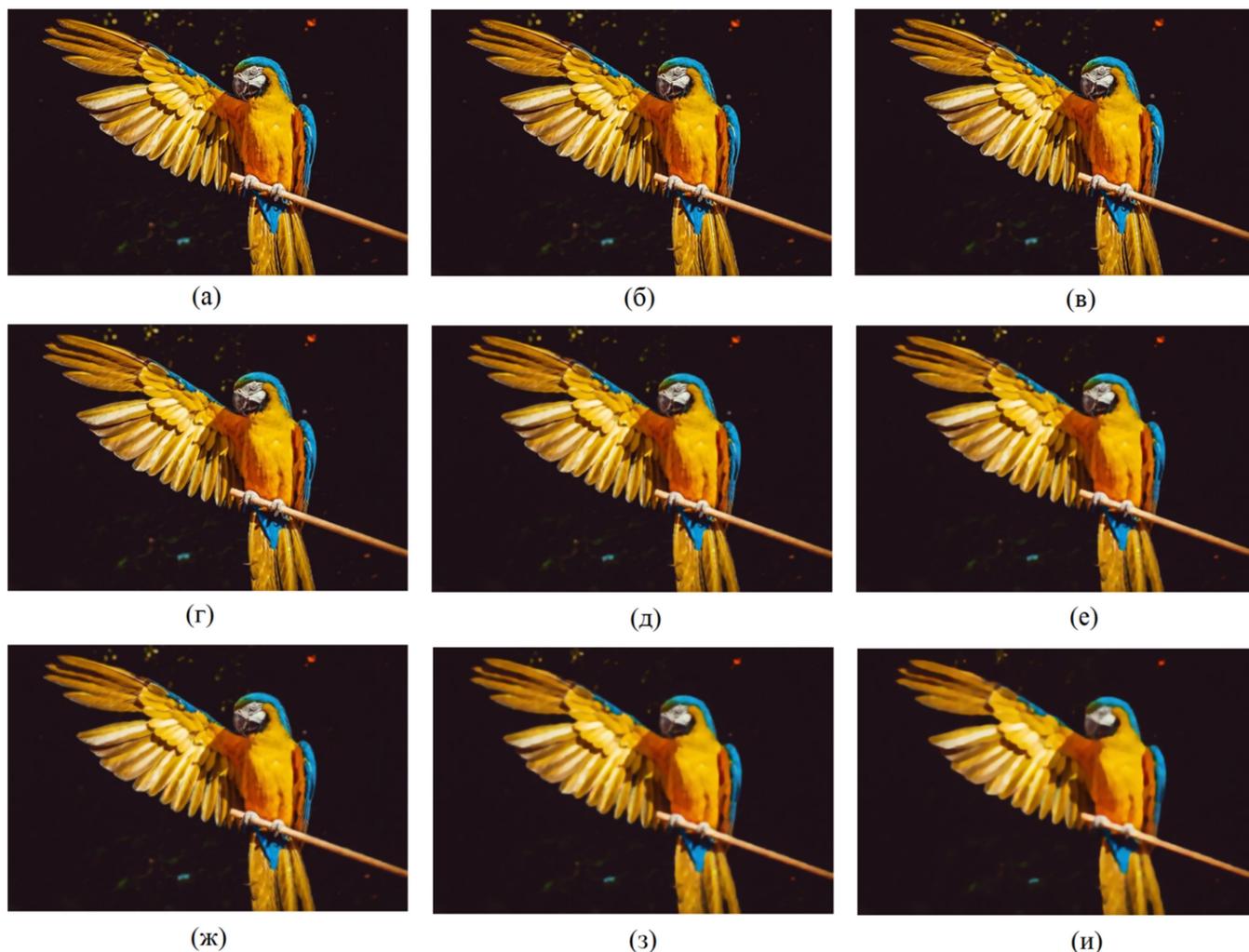


Рисунок 8 – Многократное сглаживание одним и тем же окном

На рисунке 8 приведен результат многократного сглаживания одним и тем же окном. В левом столбце под буквами (а), (г) и (ж) изображения, сглаженные один раз окнами с размером  $3 \times 3$ ,  $5 \times 5$  и  $7 \times 7$  пикселей. Последующие два столбца демонстрируют изображения, сглаженные теми же окнами два раза под буквами (б), (д) и (з) и три раза под буквами (в), (е) и (и) соответственно. Видно, как при многократном применении метода усиливается сглаживание. Шум, присущий градиенту, а в особенности его углам, также последовательно

снижается, следовательно, значения пикселей становятся более однородными, но границы при этом не теряют резкости, что имело бы место при использовании любого простого метода сглаживания, который не учитывает границы.

Последовательное применение не ограничивается сглаживанием с окном константного размера. Возможно также применение последовательного сглаживания с различным размером окна. Было выявлено, что при правильном подборе параметров можно достичь качественных результатов. Подбор параметров и границы применимости будут освещены в последующих разделах.

Подводя итог рассмотрению полученных результатов, можно заключить, что описанные в предыдущем разделе свойства, которые были обоснованы теоретически, были выявлены на практике. Но приведенных результатов недостаточно для обоснования качества представленного подхода. Необходимо также провести сравнение с существующими алгоритмами на задачах, которые, как правило, решаются при помощи сглаживания.

## **5. СРАВНЕНИЕ КАЧЕСТВА**

Учитывая математически некорректную постановку задачи сглаживания изображений, сравнение методов между собой крайне затруднено. Эта трудность в основном возникает по той же причине, что и некорректность: задача сглаживания не имеет определенного решения. Исходя из того, что решение не определено, невозможно создать метрику, по которой можно было бы измерять качество того или иного метода сглаживания. Если бы, например, как в задаче удаления шума, решение существовало бы, то можно было бы измерять качество в близости решения, выдаваемого алгоритмом к эталонному решению. При удалении шума эталоном выступает незашумленное изображение, то есть идеальный алгоритм удаления шума получает на вход

зашумленное изображение, а на выходе выдает исходное, без шума. По близости полученного решения к идеальному в данном случае определяется качество. Но допустим, решение задачи сглаживания было бы определено: имелся бы для каждого исходного изображения строго один его идеально сглаженный вариант. Возможно ли, аналогично примеру борьбы с шумом, измеряя расстояние до эталона, получить оценку качества сглаживания? Зависит от того, какие метрики использовать. Чаще всего в подобных задачах, где входными и выходными данными являются изображения, в качестве метрики используются попиксельные меры по типу корня из среднеквадратической ошибки (Root Mean Square Error, RMSE) или средней относительной ошибки (Mean Relative Error, MRE). Как пример такой задачи можно привести задачу оценки глубины на изображениях. Для такой задачи на входе цветное изображение сцены, а на выходе изображение, называемое картой глубины, где интенсивность каждого пикселя означает расстояние от камеры до этого объекта. Несмотря на то, что это классические для статистики метрики, они часто подвергаются критике в таких типах задач как оценка глубины или сглаживание изображений. Например, в работе [24] приводится критика подобных метрик именно из-за их обобщающей попиксельной природы. Они учитывают близость изображений в общем, в каждой точке отдельно и равноправно, что уводит новые алгоритмы от решения практической задачи, заставляя их максимизировать метрику, не отвечающую критериям задачи. В примере оценки глубины это локальная геометрия объектов и их границы. Для сглаживания это жесткость границ и сглаженность текстур. Получается, что даже если бы существовало определенное решение задачи сглаживания, отдельной и по своей сути такой же сложной задачей было бы сконструировать такую метрику, которая отвечала бы критериям сглаживания.

Несмотря на приведенные сложности при сравнении алгоритмов сглаживания, практика их сравнения все же существует и это в определенной степени все же возможно. Возможные подходы к сравнению можно разделить на две категории: это качественное и количественное сравнение. В предыдущем параграфе речь шла о строгом количественном сравнении, таком, которое есть в других подобных задачах. Качественное сравнение же производится человеком и по-другому называется экспертной оценкой.

Экспертная оценка основана на том, что человек обладает априорной информацией о том, какие критерии существуют в задаче сглаживания и на основе этих критериев может дать оценку качеству, сравнивая два и более результата сглаживания между собой. Такая оценка, хоть и отвечающая критериям сглаживания, имеет свои недостатки, среди которых, например, низкая скорость оценки. Она не позволяет оценивать качество на большом количестве данных в разумное время. Еще одним недостатком является ее субъективность: несмотря на то, что критерии одни и те же, восприятие качества среди экспертов может различаться.

Количественная оценка обладает куда большим количеством различных вариантов. Среди них: оценка напрямую при определенном решении, оценка на задаче (например, сглаживание шума, артефактов), косвенная оценка через другую задачу. Оценка напрямую обсуждалась выше. Ее недостатки были описаны, но из преимуществ можно отметить объективность оценки относительно разных людей, проводящих измерения, скорость работы и объемы данных, позволяющие очень точно оценить качество в различных контекстах, а не в ограниченном количестве, что характерно для экспертной оценки. На такой оценке основан, например, бенчмарк для сглаживания изображений, на котором была произведена оценка и этого алгоритма [17]. Интересно отметить, что при формировании эталонных решений бенчмарка применялись именно экспертные

оценки. Вторым вариантом оценки предполагалась оценка алгоритмов сглаживания на таких задачах, для которых метрики определены и устоялись. Примером может служить сглаживание шума. Алгоритмы для сглаживания не приспособлены для удаления шума, но могут справляться с небольшими его амплитудами, для чего зачастую используются. Более естественным применением для них может быть, например, сглаживанием артефактов сжатия, что тоже является задачей с определенными метриками качества. Последними были перечислены методы косвенного сравнения, основанные на какой-то комплексной задаче, решаемой, в том числе с использованием сглаживания на каком-либо из этапов. Во множестве различных пайплайнов обработки изображений на каком-то этапе используется сглаживание, качество которого в некоторых случаях может оказаться критичным для качества работы всей системе, так как, как правило, сглаживание находится в самом начале обработки изображения. Но бывает и иная ситуация, когда сглаживание находится в конце для улучшения качества выходного изображения. В обоих случаях, оценивая качество конечной системы с различными алгоритмами сглаживания, можно на практике судить об их качестве.

Подводя итог, отметим, что, хотя сравнение алгоритмов сглаживания изображений затруднено, на практике оно осуществляется набором различных способов, некоторые из которых будут в частности применены в данной работе в последующих разделах.

## **5.1 СРАВНЕНИЕ КАЧЕСТВЕННОЕ**

Для хорошей качественной оценки результатов работы алгоритма необходимо воспроизвести не только его результаты, но и результаты большого количества других подходов в одном и том же контексте, то есть на одном и том же изображении. Учитывая высокую трудоемкость подобной задачи, для

данной работы был выбран другой способ достижения тех же условий. В качестве данных для тестирования были выбраны исходные данные уже не раз упомянутого ранее бенчмарка для сглаживания изображений. Авторы, собственно говоря, уже выполнили данную работу по получению результатов работы множества алгоритмов и, чтобы произвести сравнение, теперь достаточно произвести только расчет результатов одного рассматриваемого подхода на тех же данных.

Для сравнения среди данных, предоставленных авторами, представлены алгоритмы: FGS[25], L0[16], L1[26], LLF[27]. SDF[4], Tree filter[28], фильтрация взвешенной медианой [29]. Учитывая, что каждый из алгоритмов сглаживания имеет свой собственный набор параметров, результаты сглаживания снимались авторами при наилучших наборах параметров. То есть были взяты результаты обработки с такими параметрами, при которых качество максимизируется. В случае с результатами подхода, представленного в данной работе, подбор параметров производился аналогичным образом. Оптимальными значениями размера ядра и количества итераций оказались 5 и 3 соответственно.

На рисунке 9 верхнее изображение в левом углу – исходное, а на последующих изображениях приведены результаты перечисленных алгоритмов. Результаты предлагаемого метода здесь и далее выделены красной рамкой для удобства восприятия. Подбор параметров производился эмпирически для улучшения результатов, предлагаемый алгоритм применялся также с лучшими параметрами для такого типа изображений.

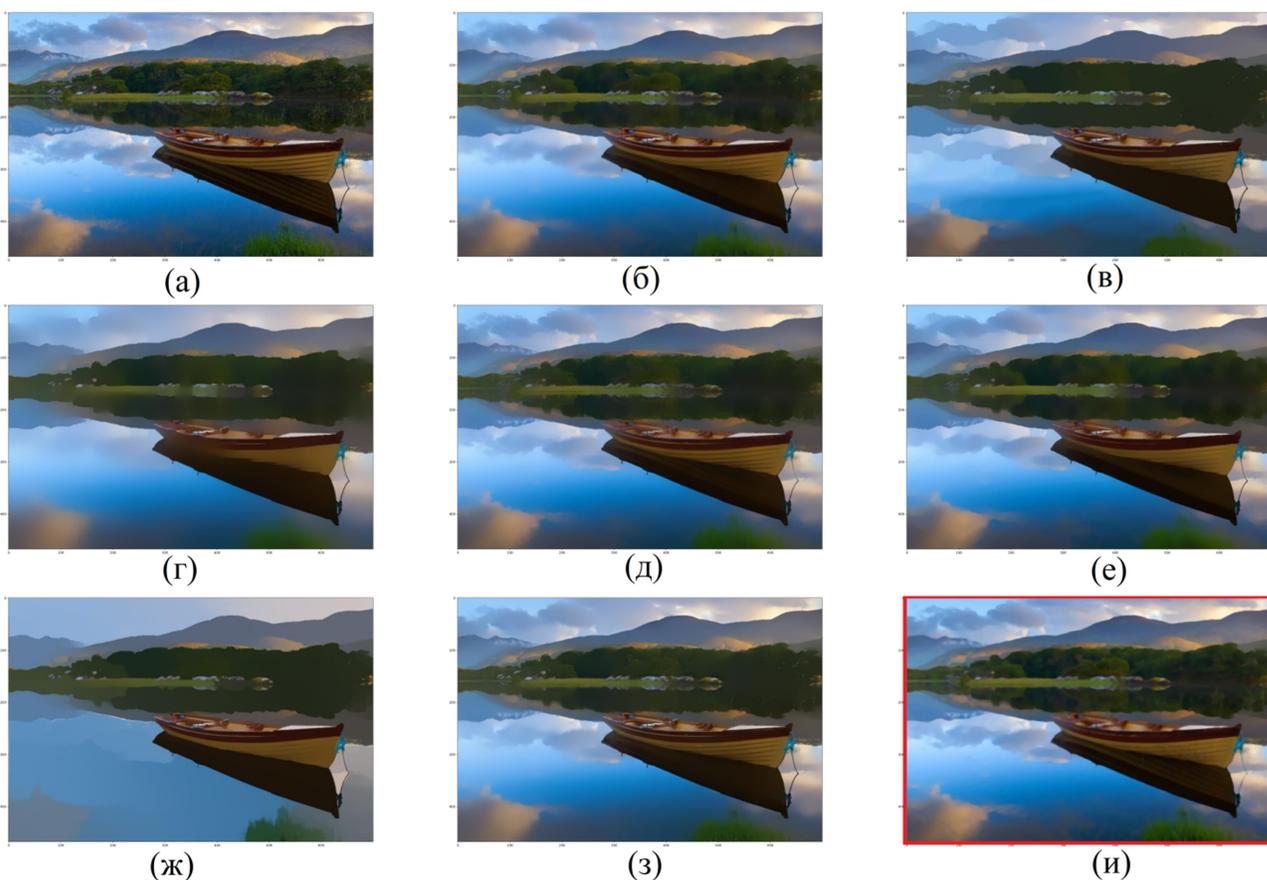


Рисунок 9 – Результаты сглаживания другими существующими фильтрами и предлагаемым

Анализируя пример, данный на рисунке 10, можно провести подробное сравнение свойств различных алгоритмов. Под буквой (б) приведены результаты алгоритма SDF, а под буквой (в) находится метод, использующий минимизацию L0 градиента. Видно, как они оба, сглаживая изображение на хорошем уровне, производят артефакты в виде ступенчатости на границах цветов. Также можно отметить общий недостаток всех приведенных здесь аналогичных методов: отсутствие сглаживания мелких деталей, если те имеют достаточно выраженные границы. Под буквой (г) результаты сглаживания FGS. Помимо ступенчатости некоторые области расплылись, и границы полностью исчезли. Обратная ситуация наблюдается у метода Tree filter под буквой (д).

Детали практически не тронуты, даже мелкие. Некоторые более равномерные области все же сглажены при этом. Под буквой (е) взвешенный медианный фильтр. Здесь сглаживание на хорошем уровне, но не сгладились некоторые мелкие детали, а некоторые сегменты также поплыли. Два предпоследних метода под буквами (ж) и (з) – L1 и LLF. L1 сглаживает очень хорошо, но все же, оставляя некоторые мельчайшие по сути детали нетронутыми и производя на некоторых границах артефакты. Стоит также отметить сильное повреждение цветов по сравнению с оригинальным изображением. LLF в свою очередь ведет себя наоборот: сглаживание произошло только на однородных областях, детали остались нетронутыми.

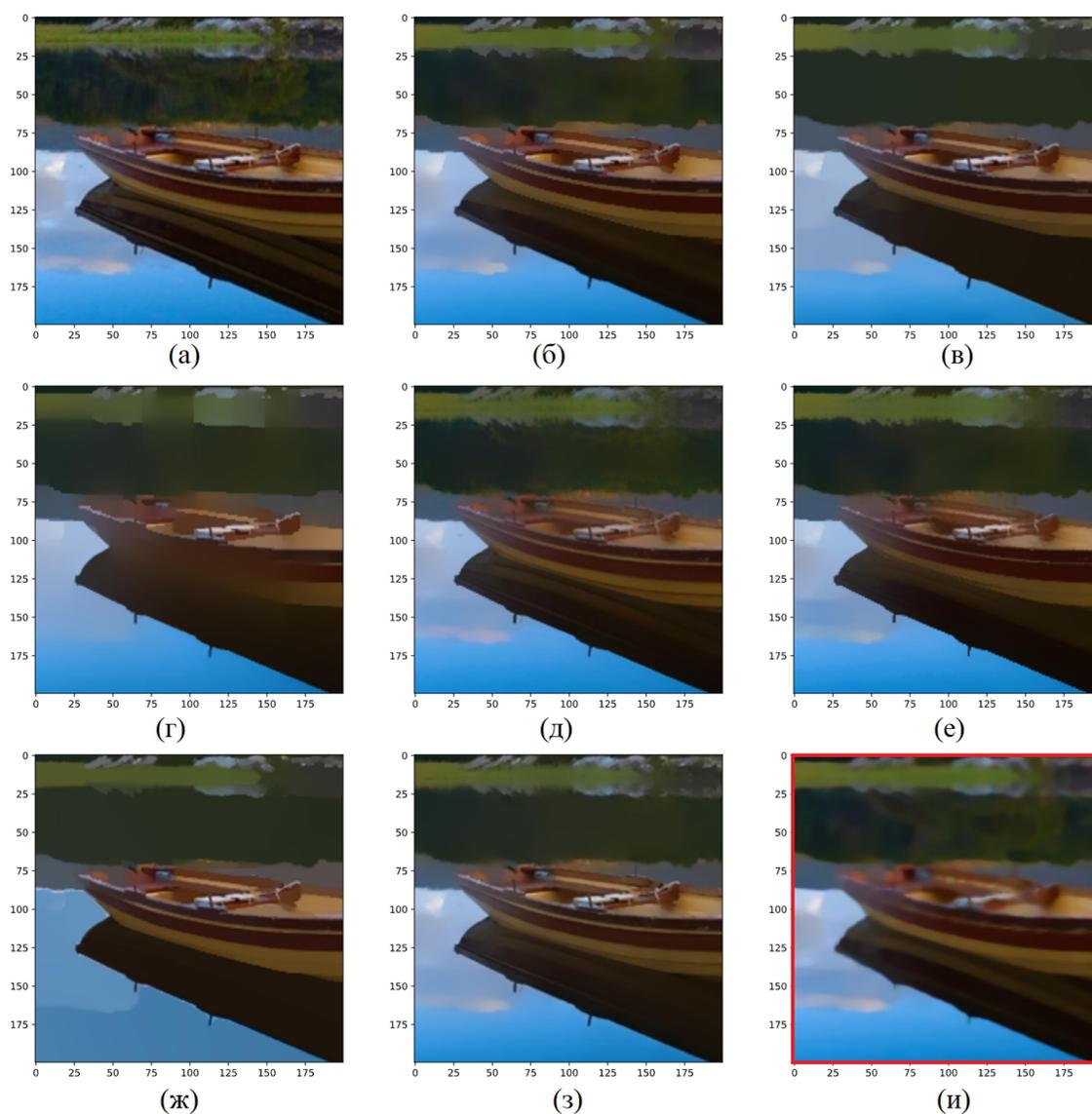


Рисунок 10 – Увеличенные фрагменты изображений с рисунка 9

Общими проблемами являются недостаточное сглаживание мелких деталей, протекание некоторых областей, полная потеря некоторых границ вследствие этого. Некоторые из методов дают артефакты: ступенчатость или изменение исходных цветов. Указанные недостатки в первую очередь связаны с природой данных подходов: практически все они являются неявными и имеют подобные недостатки, являющиеся общими для данной группы фильтров.

Предлагаемый подход позволяет производить сглаживание однородных и неоднородных областей с одинаковой силой, что дает ему определенную предсказуемость результатов по сравнению с другими рассмотренными методами. Сегменты не могут «протечь» и границы не могут быть полностью уничтожены. Помимо этого, он не дает глобального изменения цветов и ступенчатости, производя при этом изображение, сглаженное с достаточной силой. Многие мелкие несущественные детали, оставшиеся при сглаживании другими методами, также исчезают, что безусловно является критерием успешного сглаживания.

## **5.2 СРАВНЕНИЕ КОЛИЧЕСТВЕННОЕ**

В данном разделе будут представлены результаты количественного сравнения представленного в этой работе алгоритма с наиболее влиятельными подходами в области сглаживания. Будет производиться сравнение двух типов из трех описанных: сравнение на задаче с определенными метриками и сравнение напрямую, при условии, что решение задачи сглаживания задано.

### **5.2.1 СРАВНЕНИЕ НА ЗАДАЧЕ**

Решение задачи сглаживания не может быть объективно определено, но сглаживание чаще всего не используется само по себе – его используют или как предобработку для последующих стадий или в попытках устранить нежелательные дефекты на изображении. В качестве примера таких задач можно рассмотреть задачи сглаживания шума и артефактов сжатия, относящиеся ко второму типу. Подобные по своей сути, эти задачи имеют определенное решение и устоявшийся набор метрик. Сами же задачи были выбраны из-за их широкой распространенности в практической области. В большом количестве приложений сглаживание используется на ранних стадиях обработки для устранения шума и незначительных деталей.

Что более важно, у этих задач в свою очередь имеются собственные определенные решения и метрики для измерения качества. Такими метриками являются Peak Signal to Noise Ratio или PSNR и Structural Similarity или SSIM. Эти две метрики отражают сходство между двумя изображениями – между эталонным решением и результатом алгоритма. Эталонным решением в случае приведенных задач является исходное, не подверженное шуму и артефактам изображение. Таким образом, чем ближе результат сглаживания к исходному изображению, тем лучше им выполняется борьба с шумом или артефактами сжатия. Учитывая невозможность избежать потери деталей, не являющихся шумом, алгоритмы сглаживания не могут в своих результатах сравниться со специализированными алгоритмами, разработанными специально для максимизации данных метрик. Но, благодаря своей простоте и присутствию реализаций в большинстве библиотек обработки изображений, алгоритмы сглаживания зачастую используются на практике для подобных задач вместо более сложных специализированных алгоритмов.

Переходя к более подробному описанию применяемых метрик, приведем их математическую формулировку. Пусть дано незашумленное изображение  $I$  и его зашумленное приближение  $K$ . При том, что оба они имеют одинаковый размер  $m \times n$ , среднеквадратическая ошибка или Mean Squared Error (MSE) формулируется как в следующей формуле:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i, j) - K(i, j)|^2.$$

Тогда PSNR в децибелах определяется согласно следующему правилу, где  $MAX_I$  является максимальным значением пикселя. При восьмибитном представлении это число равно 255.

$$PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right).$$

Такая метрика является отношением максимально возможного сигнала к мощности шума. Чем она больше, тем ближе друг к другу изображения I и K. Учитывая такую формулировку, можно также отметить, что метрика PSNR не определена для случаев I=K, так как MSE=0.

Индекс структурного сходства или SSIM является развитием исходных методов (MSE, PSNR) в сторону приближения модели ошибки к человеческому восприятию изображений. SSIM в отличие от PSNR рассчитывается не для всего изображения, а в среднем для отдельных его окрестностей. Индекс структурного сходства для двух квадратных окрестностей x и y одинаковых размеров выражается:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + (0.01 * MAX)^2)(2\sigma_{xy} + (0.03 * MAX)^2)}{(\mu_x^2 + \mu_y^2 + (0.01 * MAX)^2)(\sigma_x^2 + \sigma_y^2 + (0.03 * MAX)^2)},$$

где аналогично расчету PSNR, MAX означает наибольшее возможное значение яркости пикселя. Обе данные метрики формулируются для монохромных изображений. В случае цветных изображений с тремя каналами значения метрик для отдельных каналов усредняются.

Для сравнения было подобрано несколько упомянутых в обзоре алгоритмов, являющихся наиболее популярными и применимыми: билатеральный фильтр, скользящий направляемый фильтр, гауссов фильтр и L0 фильтр. Гауссов фильтр, единственный неадаптивный подход, был добавлен для сравнения из-за своей большой распространенности в прикладной области. При применении аппроксимации он является довольно быстрым для вычисления и будет задавать, таким образом, базовый результат для остальных более сложных методов.

Корректное сравнение работы алгоритмов описанными метриками возможно лишь при равенстве контекстов. Это означает, что изображение, на котором производится сравнение, должно быть одним и тем же для разных

алгоритмов, чтобы применение данных метрик имело смысл [30]. Для сравнения на задаче сглаживания шума в этой работе было выбрано уже использованное изображение, которое было представлено на рисунке 6. Предварительно оно было подвержено аддитивному гауссовому шуму с нулевым средним и среднеквадратическим отклонением, изменяющимся от 10 до 100 с шагом 10, чтобы проследить качество работы алгоритмов при изменяющейся силе шума. После этого каждое из изображений было обработано каждым из алгоритмов для получения результатов сглаживания. Учитывая, что каждый из методов имеет уникальный набор параметров, влияющих на сглаживание и качество результата, все приведенные методы были применены с обширным диапазоном различных параметров. Такой метод позволяет оценить все распределение результатов, которых можно достичь с определенным фильтром, вместо того чтобы сравнивать результаты с каким-то наперед заданным значением. За счет этого он также позволяет исключить предвзятость оценки, которая может иметь место при ручном подборе параметров.

Для воспроизводимости результатов в данной работе приводятся значения параметров для каждого из использованных алгоритмов: билатеральный фильтр  $\sigma_s$  [30, 50, 100, 200, 300, 500], билатеральный фильтр  $\sigma_r$  [3, 5, 7, 9, 15, 25], размер окна гауссового фильтра [3, 5, 7, 9, 11, 13, 15, 17], L0 фильтр параметр  $\lambda$  [1/75, 1/65, 1/55, 1/45, 1/35, 1/25, 1/15, 1/10, 1/5], скользящий направляемый фильтр (СНФ)  $\sigma_s$  [1, 2, 2.5, 3, 3.5, 4, 5], СНФ  $\sigma_r$  [0.1, 0.3], СНФ число итераций 4, СНФ гауссова точность 0.05, размер окна предлагаемого фильтра [3, 5, 7, 9, 11, 13] число итераций предлагаемого фильтра [1, 2, 3, 4]. При вычислении результатов для подходов, имеющих несколько параметров, расчет производился полным перебором каждого с каждым. Расчет SSIM производился

с размером окна 7. На рисунках 11 и 12 приведены результаты вычисления метрик PSNR и SSIM соответственно.

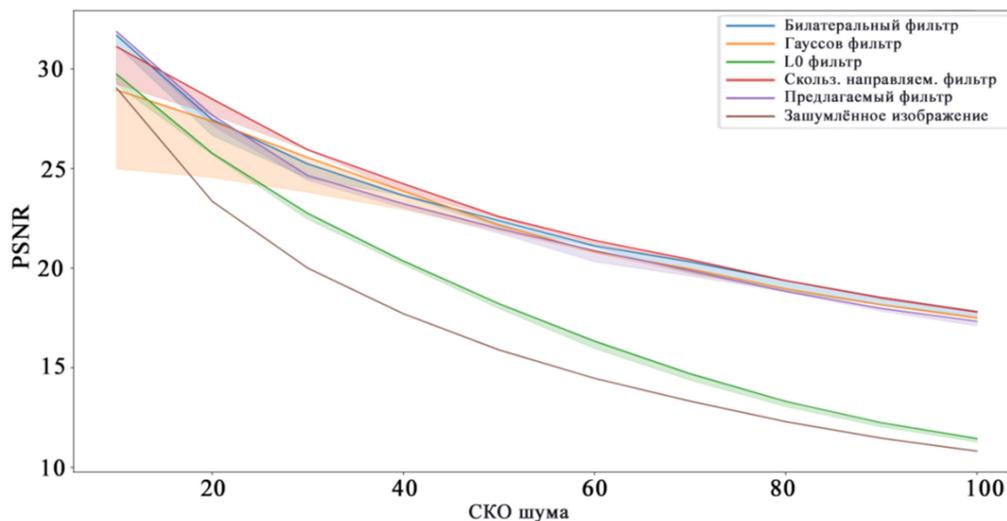


Рисунок 11 – PSNR в зависимости от СКО шума

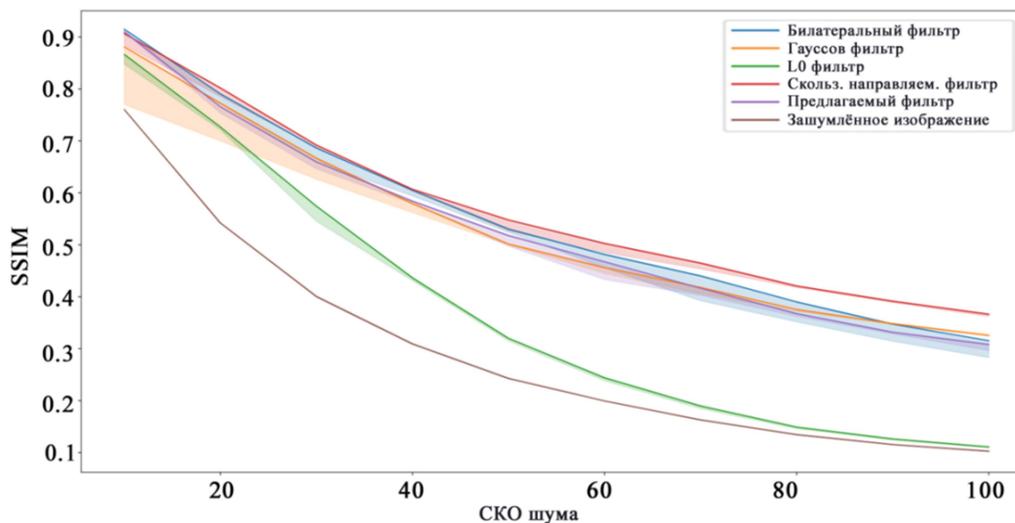


Рисунок 12 – SSIM в зависимости от СКО шума

Помимо результатов алгоритмов сглаживания, на графиках приведены также значения метрик для исходного, не сглаженного изображения с шумом. Это сделано с той же целью, что и использование гауссового фильтра, а то есть для задания базовой линии. В данном случае зашумленное изображение задает минимальное значение метрики, ниже которого использовать алгоритм сглаживания не имеет смысла. Иными словами, если алгоритм сглаживания

усиливает шум вместо того, чтобы его сглаживать, то значения метрик будут меньше, чем значения для исходного изображения. Эти значения также дают понять, насколько каждый из алгоритмов улучшает качество изображения в терминах используемой метрики. Яркие линии на графиках означают максимальное значение метрики, полученное с использованием определенного набора параметров. Для каждого из значений ширина полосы под линией соответствует стандартному отклонению метрики. Это позволяет оценить разброс распределения результатов при каждом значении СКО шума.

Переходя к осмыслению полученных результатов, можно отметить, что все они невелики по сравнению с результатами специализированных алгоритмов для борьбы с шумом, но все же улучшают качество изображения, как видно из графика. Основное расхождение между результатами наблюдается в области небольших значений СКО шума, что показывает определенные границы применимости рассматриваемых подходов. Предлагаемый фильтр на такой задаче показывает себя лучшего всего на минимальных значениях шума, в целом не уступая наиболее популярным алгоритмам сглаживания и на более высоких значениях. Полученные результаты также позволяют сделать вывод о целесообразности применения адаптивных алгоритмов сглаживания в общем. Видно, что на больших уровнях шума те показывают результаты, не существенно превосходящие неадаптивный гауссов фильтр, который, в свою очередь, может превосходить их в скорости.

Задача устранения артефактов сжатия устроена подобным же образом. Алгоритмы сжатия с потерями порождают при применении дефекты различного вида, зачастую ухудшающие зрительное восприятие изображения или качество его последующей обработки. Для получения результатов была использована Matlab-реализация алгоритма сжатия JPEG как наиболее популярного и применимого для сжатия изображений с потерями перед передачей их по сети

или для хранения на диске. Аналогично процессу с шумом, были сгенерированы 10 изображений со значением качества сжатия от 10 до 100 с шагом в 10. К ним были применены алгоритмы с теми же параметрами что и раньше, после чего были получены подобные предыдущим графики PSNR и SSIM, за исключением того факта, что по оси абсцисс теперь откладывается не СКО шума, а качество сжатия. На рисунках 13 и 14 приведены результаты вычисления метрик PSNR и SSIM соответственно.

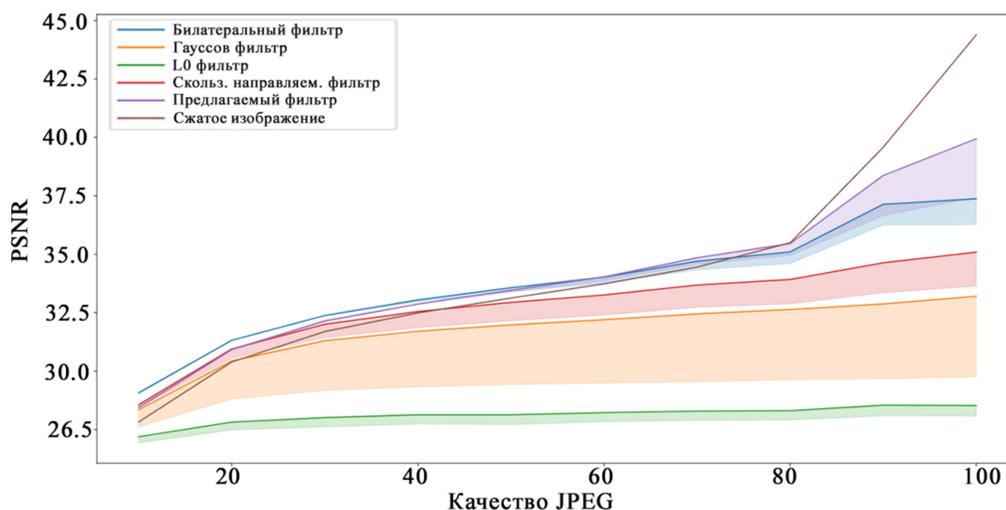


Рисунок 13 – PSNR в зависимости от качества сжатия

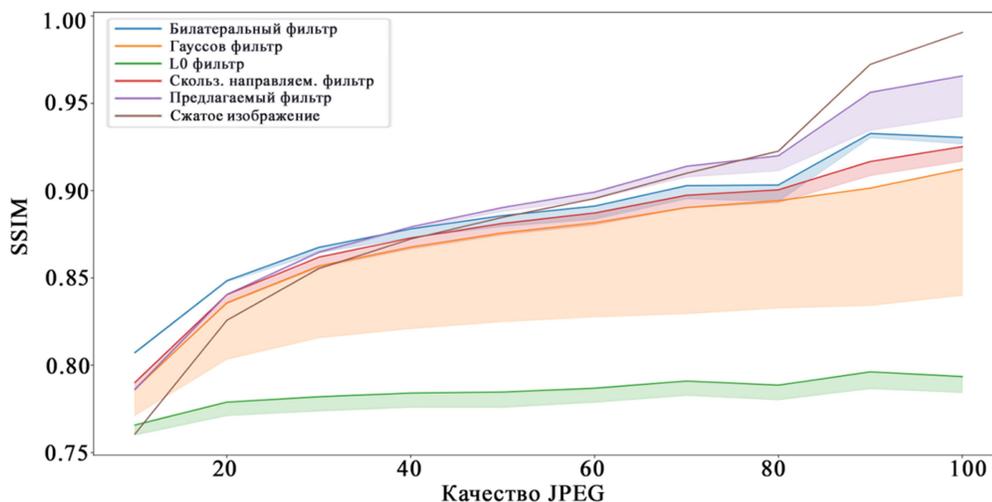


Рисунок 14 – SSIM в зависимости от качества сжатия

Анализируя количественные результаты экспериментов на задаче устранения артефактов сжатия, можно отметить, что описываемый подход на всей области определения превосходит многие, а на некоторых отрезках и все из приведенных алгоритмов сглаживания. Основное сравнение, по сути, происходит с билатеральным фильтром, отставание от которого сокращается при падении уровня качества ниже восьмидесяти. Учитывая, что только количественные метрики не могут все же обеспечить полноты оценки качества работы, а в особенности, когда речь идет о такой задаче как работа с артефактами сжатия, требуется также привести некоторые качественные результаты. Приведенные на рисунке 15 изображения дают понимание того, как метрики отражают качество работы алгоритмов. Единственное изображение в первом ряду – исходное. Во втором ряду первое изображение является сжатым с качеством 20. Далее за ним слева направо представлены результаты с максимальными значениями SSIM (обозначаемые яркой линией на рисунке 14) билатерального фильтра, гауссового фильтра. В третьем ряду аналогично: L0 фильтра, скользящего направляемого фильтра и предлагаемого.



Рисунок 15 – Сопоставление количественных метрик и качественных результатов

Можно наблюдать, как растет качество сжатого изображения после сглаживания различными методами. Также в целом видно соответствие метрики и визуального качества сглаженного изображения.

## 5.2.2 СРАВНЕНИЕ НА БЕНЧМАРКЕ

Одним из подходов к сравнению различных алгоритмов решения задачи сглаживания является задание определенного решения как эталонного, чтобы далее можно было применять классические метрики для измерения качества. Таким образом поступили авторы работы, посвященной бенчмарку для сглаживания изображений с учетом границ [17].

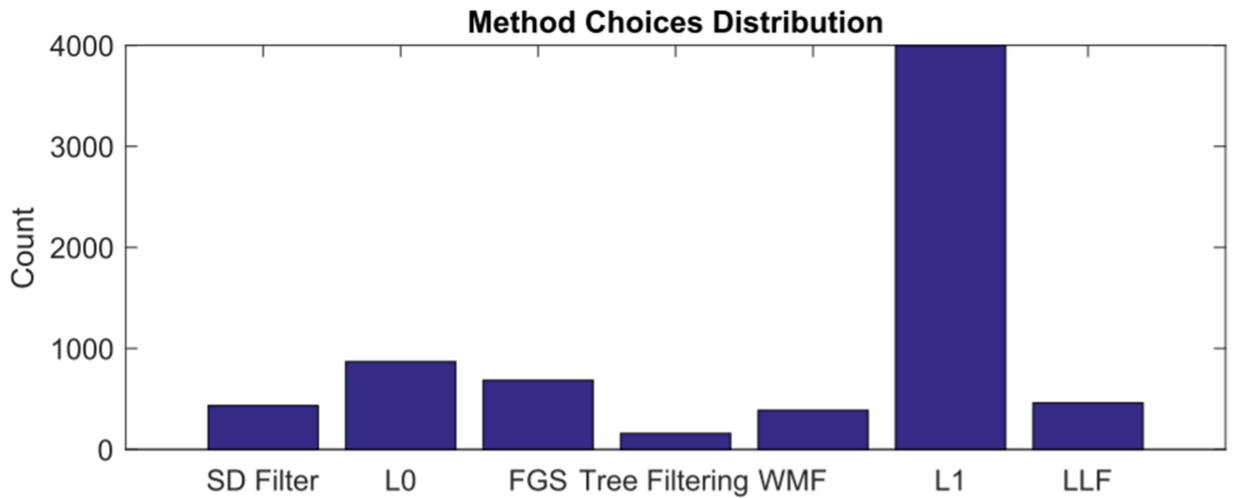
Для того, чтобы задать эталонное решение задачи сглаживания с учетом границ, авторы провели обзор работ, посвященных этой теме, и выбрали семь наиболее влиятельных классических методов, то есть таких, которые являются наиболее цитируемыми, выбирая при этом примерно равное число локальных (три) и глобальных (четыре) методов. Такими методами оказались уже упомянутые FGS, L0, L1, LLF, SDF, Tree filter и фильтрация взвешенной медианой. После этого они собрали набор данных для тестирования, состоящий из пятисот изображений, сто из которых были использованы в качестве тестовой выборки. Изображения для набора данных были отобраны вручную из двух баз изображений, ранее использовавшихся для задачи сегментации [19] и сравнения моделей качества изображений [18]. Сами изображения включают в себя множество различных сцен и объектов как представлено на рисунке 16. Отобранные изображения были обработаны выбранными алгоритмами сглаживания с различными наборами параметров. Таким образом были подготовлены данные для определения эталонного решения. Однако оценка качества сглаживания является нетривиальной задачей, а потому авторы прибегли к использованию экспертной оценки.



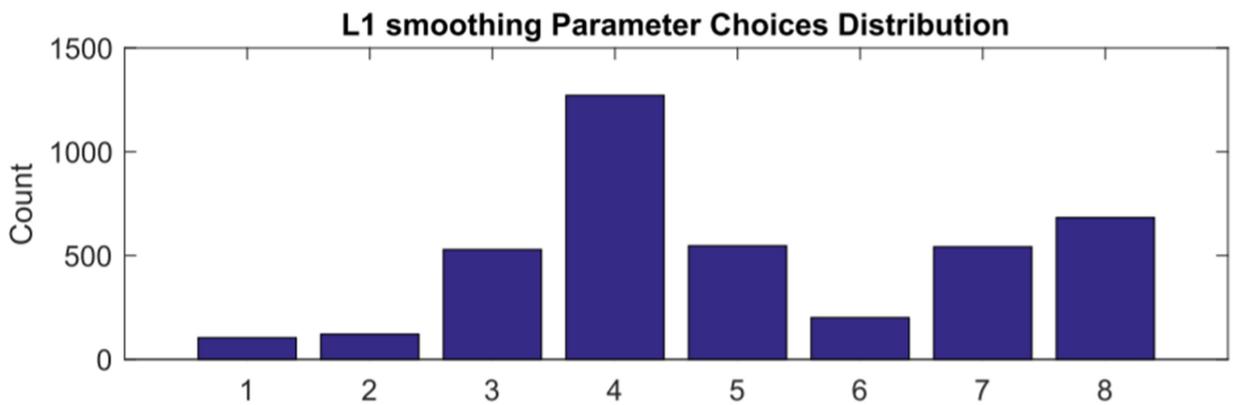
Рисунок 16 – Пример изображений бенчмарка

Для определения наилучшего качества сглаживания была сформирована группа из 26 добровольцев из числа студентов-магистров. Большинство из них не имело ранее опыта, связанного со сглаживанием изображений. Им было поручено выбрать наилучшее сглаженное изображение из набора представленных. Сначала из числа результатов работы одного алгоритма с различными параметрами, затем среди различных алгоритмов. Добровольцам для выбора подходящих вариантов давалось три ключевых инструкции. Первая: значительные края должны быть сохранены и их размытие недопустимо. Вторая: цвета сглаженного изображения должны быть максимально близко к цветам оригинала. И третья: для изображений, удовлетворяющих первым двум инструкциям, более сглаженное лучше.

На основе такой процедуры были собраны данные о том, какой из алгоритмов сглаживания (и с каким набором параметров) оказался наиболее выбираемым среди семи отобранных.



(a) Vote distribution among different algorithms



(b) Vote distribution among different parameter settings of  $L_1$  smoothing

Рисунок 17 – Распределение голосов добровольцев

На основе распределения голосов, изображенного на рисунке 17, авторами были сформулированы метрики качества сглаживания. Для того, чтобы не выбирать какой-то один метод сглаживания в качестве эталонного, было решено использовать в качестве решения взвешенную сумму пяти наиболее выбираемых методов. Используя распределение голосов, для измерения качества сглаживания были предложены следующие две метрики.

$$WRMSE = \left( \frac{\sum_t \sum_{i,j} \sum_{k=1}^5 w_{t,k} \|F(x^t)_{i,j} - Y_{i,j}^{t,k}\|^2}{\sum_t \sum_{i,j}} \right)^{\frac{1}{2}}$$

$$WMAE = \frac{\sum_t \sum_{i,j} \sum_{k=1}^5 w_{t,k} \|F(x^t)_{i,j} - Y_{i,j}^{t,k}\|_1}{\sum_t \sum_{i,j}}$$

$$w_{t,k} = \frac{\text{count}(Y^{t,k})}{\sum_{k=1}^5 Y^{t,k}}$$

Первая – взвешенный корень из среднеквадратичного отклонения, а вторая – взвешенное среднее абсолютное отклонение. Эти метрики вычисляются для каждого пикселя  $F(x^t)_{i,j}$   $t$ -го входного изображения с координатами  $i, j$ .  $Y_{i,j}^{t,k}$  обозначается, в свою очередь, соответствующий пиксель  $k$ -го из пяти эталонного изображения. Данные метрики не являются чем-то уникальным и используются в целом довольно часто с изображениями. Основным нововведением здесь считается расчет их взвешенно для пяти эталонных изображений.

Используя полученные оценки, авторы воспользовались открывшейся возможностью провести обучение сверточной нейронной сети. Было выбрано две архитектуры для такой задачи: очень глубокая сверточная нейронная сеть (VDCNN) на 20 слоев и ResNet [31]. Эти две модели и несколько других были оценены, используя предложенные метрики. Описываемый в данной работе алгоритм, в свою очередь, также был оценен, используя данные и метрики, опубликованные авторами. Результаты данных сравнений приведены в таблице 1.

Таблица 1 – Результаты оценки алгоритмов на бенчмарке

Алгоритм	Оригинал	Предлагаемый	SD Filter	L0	FGS	Tree Filter	WMF
WMAE	9.025	8.804	7.650	6.930	6.820	7.654	7.960
WRMSE	13.476	13.006	11.570	10.640	10.670	14.310	11.830
Алгоритм	L1	LLF	VDCNN	ResNet	Xu и др.[32]	Liu и др. [33]	Li и др. [20]
WMAE	5.760	7.290	6.150	5.550	9.270	7.300	6.570
WRMSE	9.890	11.060	9.780	9.030	12.490	10.080	10.180

Здесь, в дополнение к тому, что было сделано авторами бенчмарка, помимо оценки описываемого алгоритма была проведена оценка необработанных изображений. Это было сделано с целью оценить степень влияния алгоритмов сглаживания на оригинальное изображение, а также определить состоятельность предлагаемых авторами бенчмарка метрик. Метрика сглаживания изображений может считаться состоятельной, если сглаженное изображение по ней имеет лучшую оценку, чем несглаженное или сглаженное в меньшей степени. Для лучшей наглядности отобразим значения из таблицы на графике на рисунке 18, отложив одну метрику по оси абсцисс, а другую по оси ординат.

Как видно из рисунка 18, метрики, предложенные авторами работы о бенчмарке, не могут являться состоятельными в полном смысле, так как по каждой из них имеется пример, когда результаты после применения метода сглаживания оказываются хуже результатов до его применения. Описываемый метод при этом, уступая большинству оцененных, все же показывает себя лучше, чем необработанное изображение. Подобное распределение значений метрик объясняется несколькими факторами. Во-первых, результаты

нескольких алгоритмов из выборки участвуют в формировании эталонного решения с определенным весом, что предоставляет им определенный перевес по сравнению с другими алгоритмами.

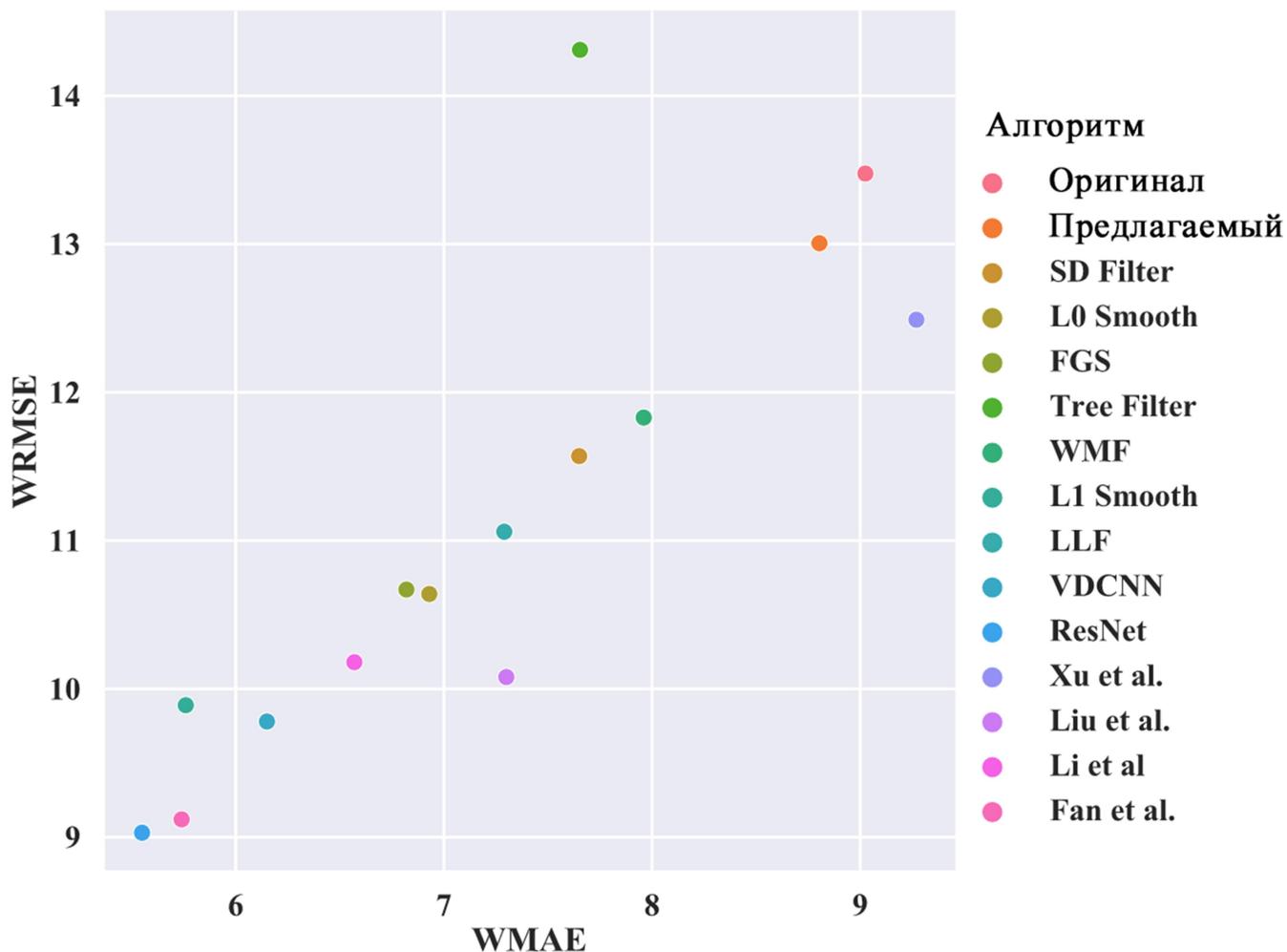


Рисунок 18 – Результаты оценки на бенчмарке

Во-вторых, метрики, использованные для оценки, имеют ряд недостатков, среди которых, в первую очередь, их точечное вычисление. Каждая из выбранных метрик использует попиксельную разницу между изображениями, что по своей сути не соотносится со смыслом задачи. Подобные метрики не учитывают изменения на границах объектов и в их текстуре, что может

привести к случаям, когда, например, изображение имеет в среднем небольшое отклонение от эталонного, но при этом на нем образуются размытые границы.

Описываемый подход к задаче сглаживания изображений с учетом границ занимает свое место среди других подобных ему подходов. В данном разделе было показано, что результаты, получаемые при его использовании, могут сравниться, а на некоторых задачах и превзойти существующие аналогичные подходы к данной задаче.

## **6. ОБСУЖДЕНИЕ**

В данном разделе приведено обсуждение предлагаемого метода, его достоинств, недостатков и границ применимости. В конце раздела также обсуждаются возможные дальнейшие разработки в предлагаемом направлении.

Достоинства предлагаемого метода уже были упомянуты косвенно, но стоит привести их все же отдельно. Поговорим сначала о самом качестве сглаживания. Как уже было показано при качественном и количественном сравнении результатов, результаты сглаживания сравнимы с существующими методами, и, более того, в некоторых областях превосходят их. Вычислительная сложность данного метода также сравнима с текущими методами. В целом она примерно равняется сложности любого метода, основанного на операторе свертки, и не может быть уменьшена увеличением окна, как, например, в случае медианного фильтра, так как веса всех пикселей в окне зависят от значения центрального пикселя и должны будут полностью подвергнуться пересчету при сдвиге окна.

Говоря о недостатках описываемого метода, нельзя не упомянуть ключевую компоненту, которая используется при расчете его весов. Градиент является подверженным шуму дескриптором, а использование обратного значения его длины усиливает данное свойство. Возьмем для примера

окрестность с почти нулевой, но не равной нулю разностью значений пикселей. Центральный пиксель в такой окрестности имеет градиент определенного случайного направления и очень маленькой длины, что, в свою очередь, делает его вес непропорционально большим, что может приводить к артефактам. Такой пиксель будет влиять на любое окно, в которое он попадает, образуя вокруг себя квадратный артефакт размером с окно сглаживания. Те же соображения могут быть применены и в отношении углов.

## **6.1 ГРАНИЦЫ ПРИМЕНИМОСТИ**

Рассмотрим теперь область применимости описываемого метода сглаживания. В разделе количественного сравнения область применимости была определена в смысле задачи, на которой лучше всего применять данный метод. В этом разделе будут определены границы применимости в смысле наилучшего размера изображения для применения сглаживания. Чтобы определить это, было взято естественное изображение размером  $4608 \times 3456$  пикселей с различными масштабами текстур и деталей. Его размер был последовательно сокращен в два раза пять раз подряд. Так были получены данные для эксперимента с масштабами. Каждое из этих изображений было обработано фильтром с различными наборами параметров.

Исходя из оценки результатов такого эксперимента, было оценено влияние масштаба на сглаживание. Недостаток информации об окрестности (из-за невозможности увеличить ее размер не теряя качество), а также подверженность шуму ограничивают область применимости данного подхода более крупными изображениями. Это связано с тем, что больший размер ядра и большее количество пикселей окрестности дают статистически больше информации для фильтра, чтобы вычислить веса. Различия между регулярными и нерегулярными границами лучше проявляются на больших изображениях, где

большое окно позволяет проводить наиболее качественное сглаживание за счет минимизации влияния шума. На рисунке 19 приведены результаты работы на различных масштабах. Для удобства отображения все изображения были приведены к размеру наибольшего. Под буквой (а) изображение с изначальным разрешением, под буквой (б) изображение в два раза меньше, то есть  $2304 \times 1728$ , под буквой (в) изображение с изначальным размером  $1152 \times 864$ , под буквой (г) изображение еще в два раза меньше  $576 \times 432$ , и наконец, под буквами (д) и (е) изображения с оригинальными размерами  $288 \times 216$  и  $144 \times 108$  соответственно.

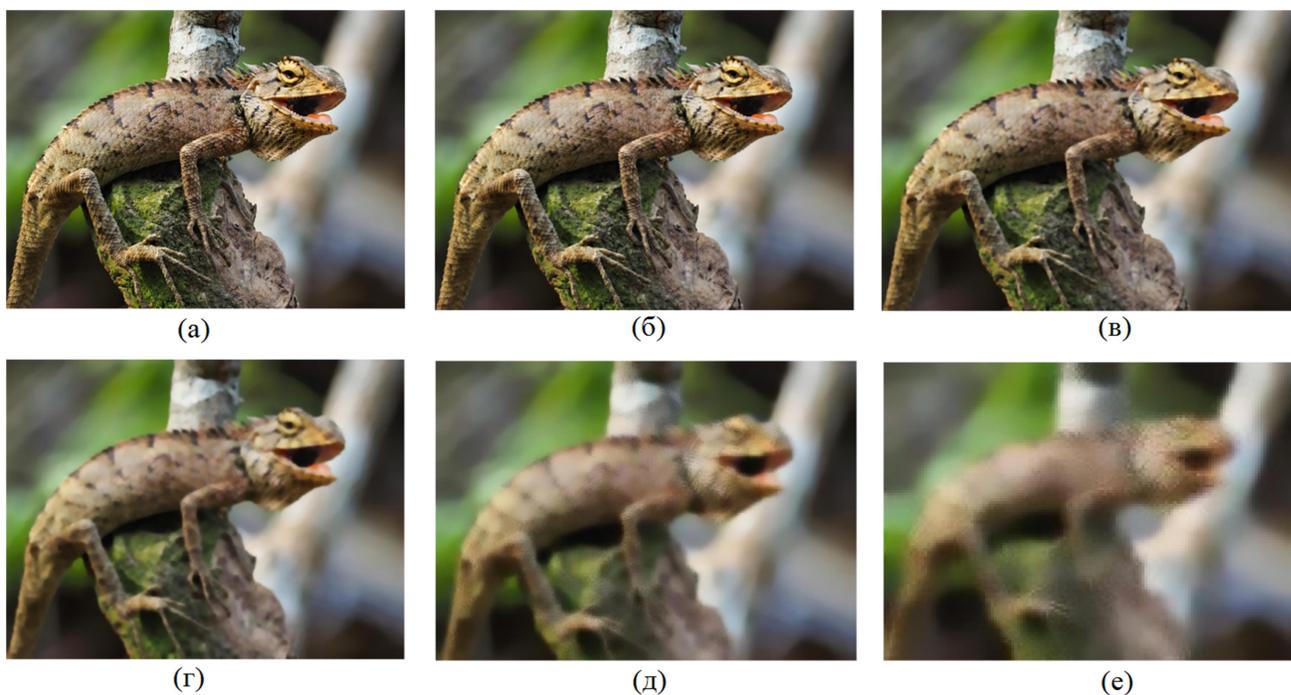


Рисунок 19 – Результат работы с ядром  $7 \times 7$  и 3 итерациями на различных масштабах

Таким образом, предположение о том, что использование описываемого подхода оптимально на более крупных изображениях, которое было сделано основываясь на природе поведения градиента, было подтверждено экспериментально.

## 6.2 ДАЛЬНЕЙШИЕ РАЗРАБОТКИ

Касательно дальнейших разработок в направлении данного подхода, предлагается несколько векторов развития. Исходя из некоторых перечисленных недостатков, можно предложить исследование применения альтернативной метрики, подобной градиенту, с его способностью выделять границы объектов, но при этом более устойчивой к локальному шуму.

Описываемый метод также открыт для разработки менее вычислительно сложной реализации. У него существует большой потенциал для распараллеливания вычислений, так как каждый из цветовых каналов, например, может обрабатываться независимо. Точно так же индивидуально могут обрабатываться любые подмножества пикселей. Как уже упоминалось, в отличие от некоторых методов, значение фильтра в новой соседней точке требует пересчета всех весов, так как зависит от значений вектора градиента центрального пикселя окна. Возможен также и поиск более быстрой приближенной реализации по примеру приближенных реализаций билатерального фильтра.

В целом простота описываемого подхода создает значительный потенциал для модификаций. Предполагаемые модификации направлены на смягчение или устранение некоторых ключевых недостатков описываемого подхода – на улучшение качества сглаживания, устранение артефактов и ускорение работы. Но не все из них непротиворечивы: могут существовать модификации (наиболее вероятно) улучшающие лишь один из данных факторов в ущерб остальным. Это естественный компромисс, на который приходится идти при разработке любого решения подобного рода.

## ЗАКЛЮЧЕНИЕ

В данной работе была поставлена следующая цель: разработать алгоритм сглаживания цветных изображений для задачи сглаживания с учетом границ на основе анализа компонент градиента. В рамках данной работы также были поставлены следующие задачи: провести обзор родственных разработок – работ предлагающих решения задачи сглаживания с учетом границ; разработать такой дескриптор, который позволил бы настраивать веса фильтра в зависимости от вида границы в окрестности; провести эксперименты, определив тем самым область применимости метода.

Подводя итог проделанной работы можно заявить, что поставленная в начале работы цель была к ее завершению достигнута. Был проведен обзор родственных разработок и выделены как исторически фундаментальные методы, так и самые последние разработки. Помимо этого, были также определены наиболее близкие подходы по виду применяемых дескрипторов, от результатов которых можно было бы оттолкнуться. Был разработан дескриптор, отвечающий требуемым свойствам, которые по предположению привели бы к качественному сглаживанию, после чего был проведен ряд экспериментов, позволяющих сравнить результаты работы метода с другими существующими, а также определить наилучшее его приложение.

Описанный в данной работе подход позволяет производить сглаживание с учетом границ на уровне, не уступающем другим аналогичным методам, при этом используя сравнительно простой в реализации дескриптор. Было показано, что лучше всего он подходит для задачи сглаживания артефактов сжатия. Исходя из этого, наилучшим его применением было бы поместить его на стадии постобработки сжатых изображений перед демонстрацией их пользователю, для улучшения их визуального качества. Применяя его с высокой степенью

параллельности, можно было бы достичь высокой производительности для приложений реального времени.

Также в рамках апробации данной работы, было опубликовано несколько научных работ. Впервые данный метод был представлен в работе под названием «Image Smoothing Algorithm Based on Gradient Analysis» в мае 2020 года на конференции USBEREIT [35]. Дальнейшие исследования и эксперименты были освещены в работе «О свойствах алгоритма сглаживания на основе анализа градиента» опубликованной в серии «Вычислительная математика и информатика» журнала «Вестник Южно-Уральского Государственного Университета» [36].

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Tripathi, A.K. Single image fog removal using bilateral filter / A. Tripathi, S. Mukhopadhyay // 2012 IEEE International conference on signal processing, computing and control. – IEEE, 2012. – P. 1–6.
2. He, K. Image segmentation on adaptive edge-preserving smoothing / K. He, D. Wang, X. Zheng // Journal of electronic imaging. – 2016. – № 5(25).
3. Meyer, F. Levelings, Image simplification filters for segmentation / F. Meyer // Journal of Mathematical Imaging and Vision. – 2004. – № 1(20). – P. 59–72.
4. Ham, B. Robust guided image filtering using nonconvex potentials / B. Ham, M. Cho, J. Ponce // IEEE transactions on pattern analysis and machine intelligence. – 2017. – № 1(40). – P. 192–207.
5. Canny, J.F. A computation approach to edge detection / J.F. Canny // IEEE transactions on pattern analysis and machine intelligence. – 1986.– № 6(8). – P. 670–700.
6. Hadamard, J. Sur les problèmes aux dérivées partielles et leur signification physique / J. Hadamard // Princeton University Bulletin. – 1902. – P. 49–52.
7. Zhao, L. Local activity-driven structural-preserving filtering for noise removal and image smoothing / L. Zhao, H. Bai, J. Liang, A. Wang, B. Zeng, Y. Zhao // Signal Processing. – 2019. №1(157). – P. 62–72.
8. Xiaonan, F. Learning Explicit Smoothing Kernels for Joint Image Filtering / F. Xiaonan, W. Miao, A. Shamir, S. Hu // Pacific Graphics. – 2019. – № 38(7). – P. 180–190.
9. Gonzalez R.C. Digital Image Processing / R.C. Gonzalez, R.E. Woods – New York: Pearson, – 2018. – 1022 p.

10. Tomasi, C. Bilateral filtering for gray and color images / C. Tomasi, R. Manduchi // Proceedings of the IEEE international conference on computer vision. – 1998. – P. 839–846.
11. Chaudhury, K.N. Fast  $O(1)$  bilateral filtering using trigonometric range kernels / K.N. Chaudhury, D. Sage, and M. Unser // IEEE Trans. Image Processing. – 2011. – № 12(20). – P. 3376–3382.
12. Cho, H. Bilateral texture filtering / H. Cho, H. Lee, H. Kang, S. Lee // ACM transactions on graphics. – 2014. – № 4(33). – P. 1–8.
13. He, K. Guided image filtering / K. He, J. Sun, X. Tang // IEEE transactions on pattern analysis and machine intelligence. – 2013. – № 6(35). – P. 1397–1409.
14. Zhang, Q. Rolling guidance filter / Q. Zhang, X. Shen, L. Xu, J. Jia // Lecture notes on computer science. (including subseries lecture notes on artificial intelligence and lecture notes on bioinformatics). – 2014. – № 8691 – P. 815–830.
15. Yoshimura, K. Structure-tensor-based anisotropic rolling filter for image smoothing / K. Yoshimura, Y. Xin, N. Xie // Proceedings of SPIE. – 2019. – № 1104904.
16. Xu, L. Image smoothing via  $L_0$  gradient minimization / L. Xu, C. Lu, Y. Xu, J. Jia // ACM Transactions on Graphics. – 2011. – № 30(6). – P. 1–12.
17. Zhu, F. A benchmark for edge-preserving image smoothing / F. Zhu, Z. Liang, X. Jia, L. Zhang, Y. Yu // IEEE Transactions on Image Processing. – 2019. – № 7(28). – P. 3556–3570.
18. Ma, K. Group MAD competition? A new methodology to compare objective image quality models / K. Ma, Q. Wu, Z. Wang, Z. Duanmu, H. Yong, H. Li, L. Zhang // In proceedings IEEE conference on computer vision and pattern recognition. – 2016. – P. 1664–1673.

19. Arbelaez, P. Contour detection and hierarchical image segmentation / F. Zhu, Z. Liang, X. Jia, L. Zhang, Y. Yu // IEEE Transactions on pattern analysis and machine intelligence. – 2011. – № 5(33). – P. 898–916.
20. Lu, K. Deep Texture and structure aware filtering network for image smoothing / K. Lu, S. You, N. Barnes // Proceedings of the European Conference on Computer Vision (ECCV). – 2018. – P. 217–233.
21. Fan, Q. Image Smoothing via Unsupervised Learning / Q. Fan, J. Yang, D. Wipf, B. Chen, X. Tong. // ACM transactions on graphics. – 2018. – № 6(37). – P. 1–14.
22. Kass, M. Analyzing oriented patterns / M. Kass, A. Witkin // Computer vision, graphics, and image processing. – 1987. – № 3(37). – P. 362–385.
23. Wang, D.C. Gradient inverse weighted smoothing scheme and the evaluation of its performance / D.C. Wang, A.H. Vagnucci, C.C. Li // Computer graphics and image processing. – 1981. – № 2(15). – P. 167–181.
24. Hu, J. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries / J. Hu, M. Ozay, Y. Zhang, T. Okatani // 2019 IEEE Winter Conference on Applications of Computer Vision. – IEEE, – 2019. – P. 1043–1051.
25. Min, D. Fast global image smoothing based on weighted least squares / D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, M. Do // IEEE transactions on image processing – 2014. – №12(23). – P. 5638–5653.
26. Bi, S. An L1 image transform for edgepreserving smoothing and scene-level intrinsic decomposition / S. Bi, X. Han, Y. Yu // ACM transactions on graphics. – 2015. – № 4(34). – P. 1–12.
27. Paris, S. Local Laplacian filters: Edgeaware image processing with a Laplacian pyramid / S. Paris, S.W. Hasinoff, J. Kautz // ACM transactions on graphics. – 2011. – № 4(30). – P. 1–68.

28. Bao, L. Tree filtering: Efficient structure-preserving smoothing with a minimum spanning tree / L. Bao, Y. Song, Q. Yang, H. Yuan, G. Wang // IEEE Transactions on pattern analysis and machine intelligence. – 2014. – № 2(23). – P. 555–569.
29. Zhang, Q. 100+ times faster weighted median filter (WMF) / Q. Zhang, L. Xu, J. Jia // In Proceedings IEEE conference on computer vision and pattern recognition. – 2014. – C. 2830–2837.
30. Huynh-Thu Q., Ghanbari M. Scope of validity of PSNR in image/video quality assessment / Q. Huynh-Thu, M. Ghanbari // Electronics letters. – 2008. – № 13(44). – P. 800–801.
31. Ledig, C. Photo-realistic single image super-resolution using a generative adversarial network / C. Ledig, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang // In proceedings of conference on computer vision and pattern recognition. – 2017. – P. 4681–4690.
32. Xu, L. Deep edge-aware filters / L. Xu, J. Ren, Q. Yan, R. Liao, J. Jia // In proceedings 32nd international conference on machine learning (ICML). – 2015. – P. 1669–1678.
33. Liu, S. Learning recursive filters for low-level vision via a hybrid neural network / S. Liu, J. Pan, and M.-H. Yang // In proceedings of european conference on computer vision. – Cham, Switzerland: Springer, 2016. – P. 560–576.
34. Li, Y. Deep joint image filtering / Y. Li, J.-B. Huang, N. Ahuja, and M.-H. Yang // In proceedings of european conference on computer vision. – Cham, Switzerland: Springer, 2016. – P. 154–169.
35. Gudkov, V. Image Smoothing Algorithm Based on Gradient Analysis. / V. Gudkov, I. Moiseev // Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology. – 2020. – P. 403–406.

36. Гудков, В.Ю. О свойствах алгоритма сглаживания цветных изображений на основе анализа градиента / В.Ю. Гудков, И.Ю. Моисеев // Вестник ЮУрГУ. Серия «Вычислительная математика и информатика». – 2020. – № 3(9). – С. 5–16.

# ПРИЛОЖЕНИЕ А

## ИСХОДНЫЙ КОД РЕАЛИЗАЦИИ ПРЕДЛАГАЕМОГО МЕТОДА

### Листинг А.1 - Исходный код реализации

```
#include "stdint.h"
#include <cmath>

#define MIN(a,b) ((a) < (b) ? (a) : (b))

template<typename Tf, typename Ts>
Tf angle(Ts* vector) { return atan2(vector[1], vector[0]); }

template<typename Tf, typename Ts>
Tf module(Ts* vector) { return sqrt(vector[0] * vector[0] + vector[1] * vector[1]); }

template<typename Tu, typename Ts>
void grad(int x, int y, int c, Tu*** image, Ts* grad)
{
    grad[0] = Ts(image[y][x - 1][c] - image[y][x + 1][c]);
    grad[1] = Ts(image[y + 1][x][c] - image[y - 1][x][c]);
}

template<typename Tu, typename Ts>
void computeGrads(Tu*** src, Ts**** dst, uint32_t height, uint32_t width, uint32_t colors)
{
    /*
     * The function to compute the gradient vectors.
     * Takes the image from src and leaves gradient vectors as [x, y] in dst.
     *
     * Recieves:
     * Tu*** src - source image, array with the shape (height x width x colors)
     * Tf*** dst - destination array, where angles will be stored; has the same shape as src
     * uint32_t height, width, colors - dimensions of the src image
     */
    for (uint32_t c = 0; c < colors; c++)
        for (uint32_t i = 1; i < height - 1; i++)
            for (uint32_t j = 1; j < width - 1; j++)
                grad(j, i, c, src, dst[i][j][c]);
}

template<typename Tf, typename Tu, typename Ts>
void computeAngles(Tu*** src, Tf*** dst, uint32_t height, uint32_t width, uint32_t colors)
{
    /*
     * The function to compute the angles of the gradients.
     * Takes the image from src and leaves angles in dst. Computes gradients.
     *
     * The angles are given by standard atan2 function, evaluated from gradient vector.
     */
}
```

```

    Recieves:
    Tu*** src - source image, array with the shape (height x width x colors)
    Tf*** dst - destination array, where angles will be stored; has the same shape as src
    uint32_t height, width, colors - dimensions of the src image
*/
for (uint32_t c = 0; c < colors; c++)
    for (uint32_t i = 1; i < height - 1; i++)
        for (uint32_t j = 1; j < width - 1; j++)
            {
                Ts* gradient = new Ts[2];
                grad<Tu, Ts>(j, i, c, src, gradient);
                dst[i][j][c] = angle<Tf, Ts>(gradient);
                delete gradient;
            }
}

template<typename Tf, typename Tu, typename Ts>
void computeModules(Tu*** src, Tf*** dst, uint32_t height, uint32_t width, uint32_t colors)
{
    /*
    The function to compute the modules.
    Takes the image from src and leaves modules in dst. Computes gradients.

    The modules are euclidean norm of gradients.

    Recieves:
    Tu*** src - source image, array with the shape (height x width x colors)
    Tf*** dst - destination array, where modules will be stored; has the same shape as src
    uint32_t height, width, colors - dimensions of the src image
*/
    for (uint32_t c = 0; c < colors; c++)
        for (uint32_t i = 1; i < height - 1; i++)
            for (uint32_t j = 1; j < width - 1; j++)
                {
                    Ts* gradient = new Ts[2];
                    grad<Tu, Ts>(j, i, c, src, gradient);
                    dst[i][j][c] = module<Tf, Ts>(gradient);
                    delete gradient;
                }
}

void filter(Tu*** src, Tf*** dst, Tf*** modules, Tf*** angles,
           uint32_t ksize, uint32_t height, uint32_t width, uint32_t colors)
{
    for (uint32_t c = 0; c < colors; c++)
        for (uint32_t i = 0; i < height; i++)
            for (uint32_t j = 0; j < width; j++)
                {
                    int up = i - ksize / 2;
                    int left = j - ksize / 2;
                    int down = i + ksize / 2 + 1;
                    int right = j + ksize / 2 + 1;
                    double sumWeights = .0;

```

```

double result = .0;
for (int k = up; k < down; k++)
{
    if (k < 0 || k >= height) continue;
    for (int l = left; l < right; l++)
    {
        if (l < 0 || l >= width) continue;
        if (modules[k][l][c] == .0) continue;

        Tf weight = .0;
        if ((k != i) || (l != j))
        {
            Tf a = 1. / modules[k][l][c];
            Tf beta = 2. * (angles[i][j][c] - angles[k][l][c]);
            weight = (cos(beta) + 1.) * a;
        }
        else
            //weight of central pixel
            weight = 1.;

        result += weight * src[k][l][c];
        sumWeights += weight;
    }
}
if (sumWeights != 0)
    dst[i][j][c] = Tf(result / sumWeights);
else
    //pixel remains without changes if sum of weights = 0
    dst[i][j][c] = Tf(src[i][j][c]);
}
}

```