

Институт ИТКН

Кафедра инженерной кибернетики

Направление подготовки: 01.03.04 прикладная математика

Квалификация (степень): **бакалавр**

Курс: **3**

Семестр: **6**

КУРСОВАЯ РАБОТА

на тему:

Создание спиральных раскрасок на основе произвольных изображений

учебная дисциплина

«Методы и средства обработки изображений»

2023 – 2024 уч. год

Команда учащихся: Матюшков А. Н.

Преподаватель: Полевой Д. В.

Оценка: _____

Дата защиты: _____

Москва
2024

Содержание отчета

Введение	3
1. Постановка задачи.....	4
2. Используемые средства разработки и системные требования.....	5
3. Описание алгоритма.....	6
4. Результаты запусков.....	7
5. Выводы.....	13

Введение

Современные методы и средства обработки изображений предоставляют широкие возможности для создания различных визуальных эффектов и художественных трансформаций. Одним из таких творческих подходов является генерация спиральных раскрасок на основе произвольных изображений. Реализация таких раскрасок вызывает творческий интерес, поэтому я выбрал эту задачу для своей курсовой работы.

В данной курсовой работе рассматривается разработка программы на языке C++ с использованием библиотеки OpenCV, предназначенной для преобразования входного изображения в формат спиральной раскраски. Программа принимает на вход произвольное изображение и генерирует два спиральных контура, которые накладываются на изображение и изменяют свою толщину в зависимости от контраста цвета в исходном изображении. В результате получается уникальное изображение, готовое для дальнейшей раскраски.

Использование OpenCV позволяет эффективно обрабатывать изображения, применять различные фильтры и проводить анализ контраста. В данной работе подробно рассматриваются методы обработки изображений, алгоритмы создания спиральных контуров и принципы изменения их толщины на основе контрастности. Также обсуждаются технические аспекты реализации программы и приводятся примеры полученных изображений.

Целью данной курсовой работы является изучение и применение методов обработки изображений для создания спиральных раскрасок, а также демонстрация возможностей библиотеки OpenCV в реализации творческих и художественных задач.

Постановка задачи

В рамках курсовой работы по предмету «Методы и средства обработки изображений» необходимо разработать программное обеспечение, которое будет преобразовывать произвольные изображения в формат спиральной раскраски. Основные требования и этапы выполнения задачи следующие:

1. Анализ и предварительная обработка изображения:

- Программа должна принимать на вход изображение в популярных форматах (например, JPEG, PNG, BMP).
- Выполнить предварительную обработку изображения для улучшения качества анализа

2. Создание спиральных контуров:

- Разработать алгоритм для генерации двух спиральных контуров, которые будут наложены на изображение.

3. Изменение толщины контуров в зависимости от контрастности изображения:

- Толщина спиральных контуров должна изменяться в зависимости от контрастности цвета в соответствующих областях исходного изображения.
- Использовать методы анализа изображения для определения контраста.

4. Формирование окончательного изображения:

- Отрисовать спиральные контуры на новом изображении
- Сформировать итоговое изображение, пригодное для раскраски, и сохранить его

5. Тестирование и оптимизация:

- Провести тестирование программы на различных изображениях.
- Оптимизировать алгоритмы для повышения качества изображений.

Таким образом, основная цель данной курсовой работы заключается в создании программы, способной преобразовывать произвольные изображения в художественные спиральные раскраски, используя методы анализа и обработки изображений на базе библиотеки OpenCV. Программа должна не только генерировать спиральные контуры, но и адаптировать их толщину в зависимости от контрастности изображения, что придаст раскраскам уникальный и выразительный вид.

Использованные средства разработки и системные требования

Для разработки программного обеспечения, реализующего преобразование изображений в спиральные раскраски на основе произвольных изображений, были использованы следующие технологии и среда разработки:

1. **Язык программирования C++:** Производительность и широкое применение в области компьютерного зрения и обработки изображений.
2. **Библиотека OpenCV (Open Source Computer Vision Library):** OpenCV предоставляет мощные инструменты для обработки изображений, анализа контраста, детекции границ и многих других операций, необходимых для реализации поставленной задачи.
3. **Среда разработки Visual Studio 2022:** в качестве среды разработки был выбран Visual Studio 2022 в связи с его удобным интерфейсом, поддержкой языка C++ и возможностью интеграции с Git для управления версиями.
4. **GitHub:** для контроля версий технология Git. Это обеспечило возможность отслеживать изменения в коде, управлять ветками
5. **CMake:** автоматически генерирует файлы проекта для различных интегрированных сред разработки (IDE), таких как Visual Studio, Xcode и другие, что делает сборку и запуск проекта более удобным.
6. **Операционная система Windows 11:** для запуска среды разработки и выполнения программы использовалась операционная система Windows 11, совместимая с использованием Visual Studio 2022 и OpenCV.

Использование этих технологий позволило эффективно реализовать поставленную задачу, обеспечив высокую производительность и качество обработки изображений в рамках проекта.

Описание алгоритма

1. Загрузка изображения:

- Программа начинает с загрузки произвольного изображения в память. Изображение может быть в любом популярном формате, таком как JPEG, PNG или BMP.

2. Предварительная обработка изображения:

- Прежде чем приступить к созданию спиральных раскрасок, необходимо выполнить предварительную обработку изображения для улучшения качества анализа, например нормализация контраста.

3. Создание спиральных контуров:

- На этом этапе программа создает два спиральных контура, которые будут наложены на изображение.

$$x = r * \cos(a)$$

$$y = r * \sin(a)$$

4. Определение толщины контуров:

- Толщина спиральных контуров изменяется в зависимости от контрастности цвета в соответствующих областях исходного изображения. Это реализовано путем анализа яркости пикселей вдоль контуров и регулировки толщины в соответствии с полученными данными.

$$\text{thickness} = 1 + ((255 - \text{averageColor}) / 255) * (\text{spiralWidth} - 1)$$

5. Наложение контуров на изображение:

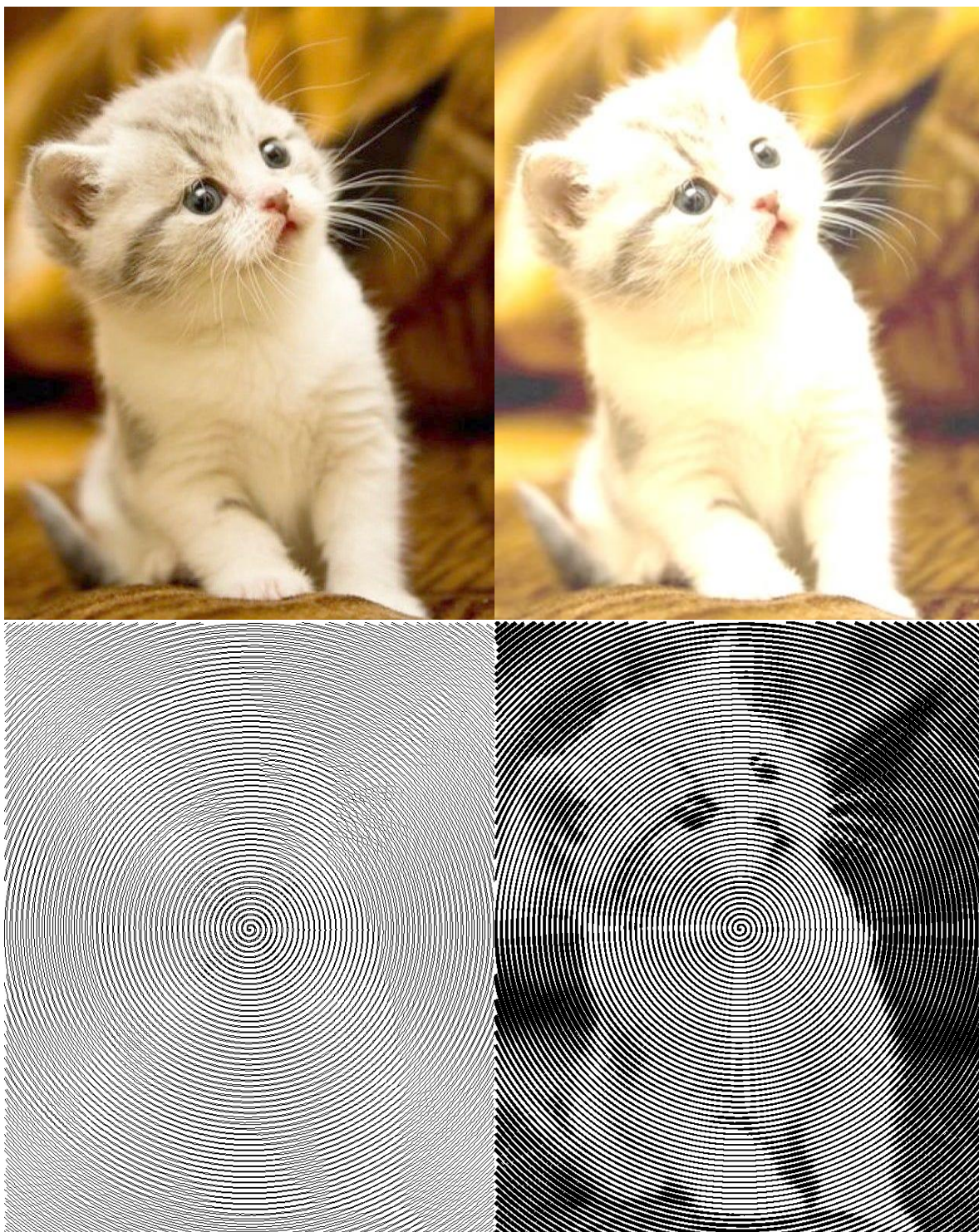
- Полученные спиральные контуры наложены на исходное изображение с учетом измененной толщины. Это позволяет создать эффект спиральной раскраски, где контуры соответствуют различным уровням контраста на изображении.

6. Сохранение и отображение результатов:

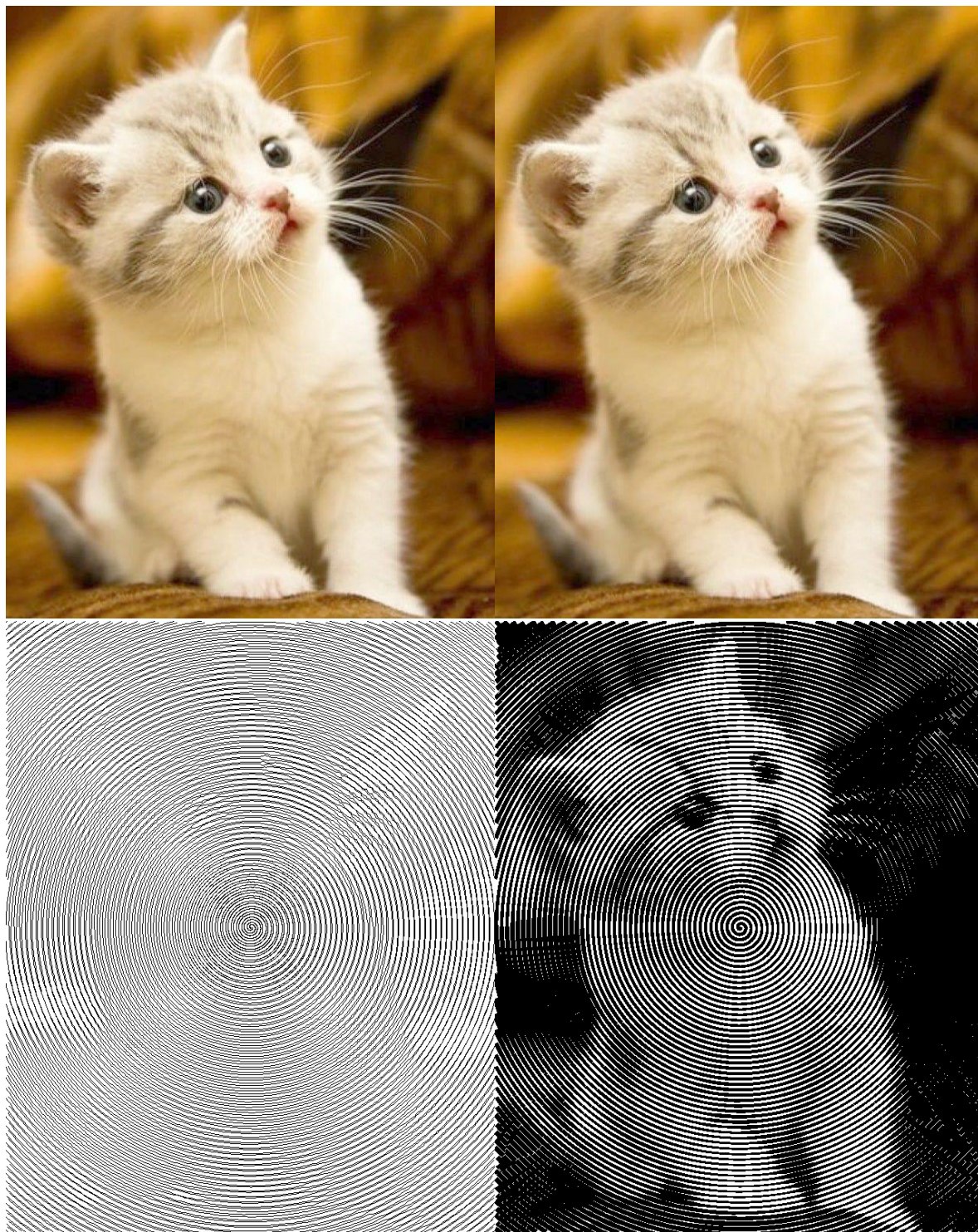
- Полученное спиральное изображение может быть сохранено в файле и отображено пользователю для просмотра. Программа также может предоставить возможность дополнительной настройки параметров и сохранения результатов.

Результаты запусков

Ширина спирали 7px, контрастирование – да

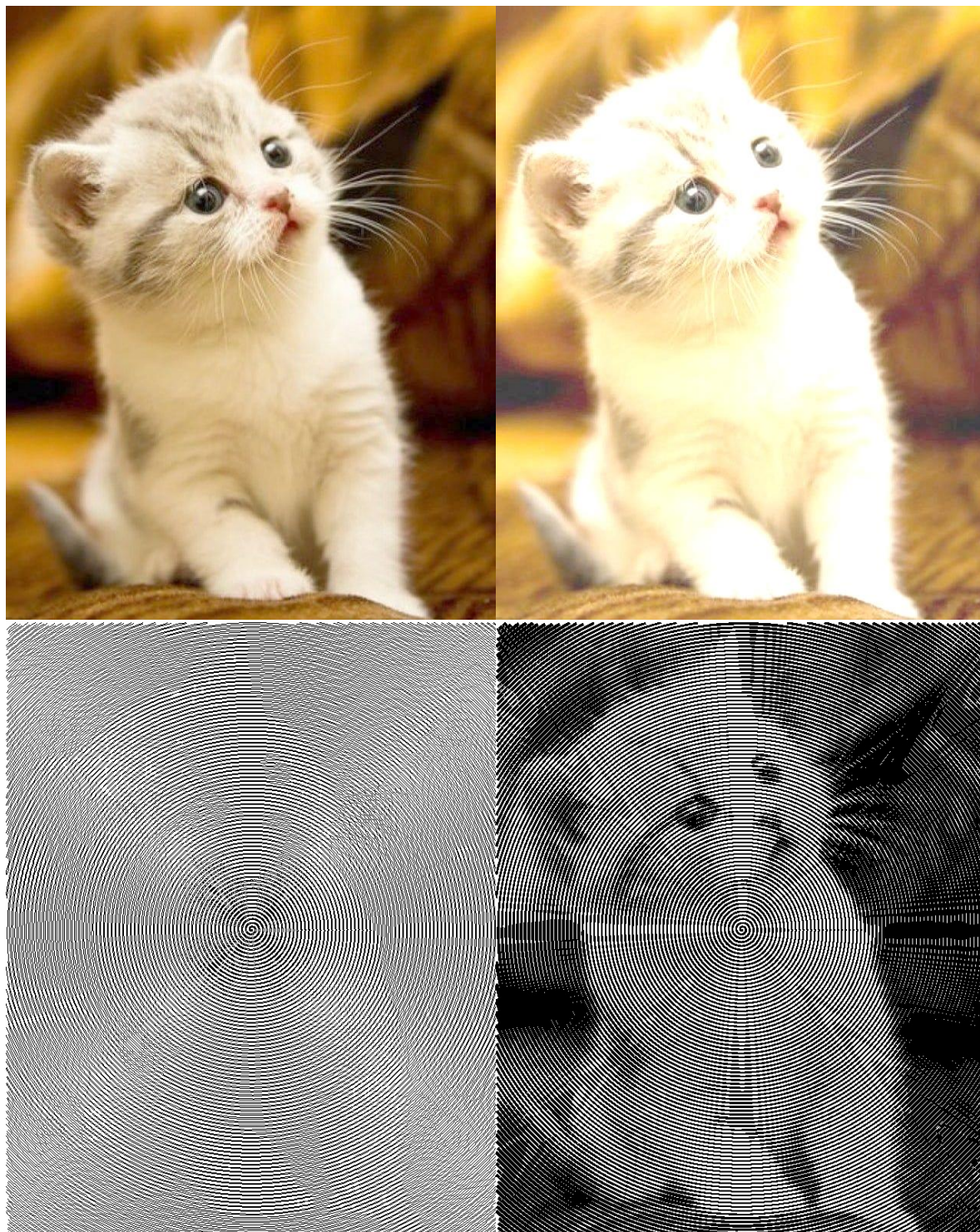


Ширина спирали 7px, контрастирование – нет

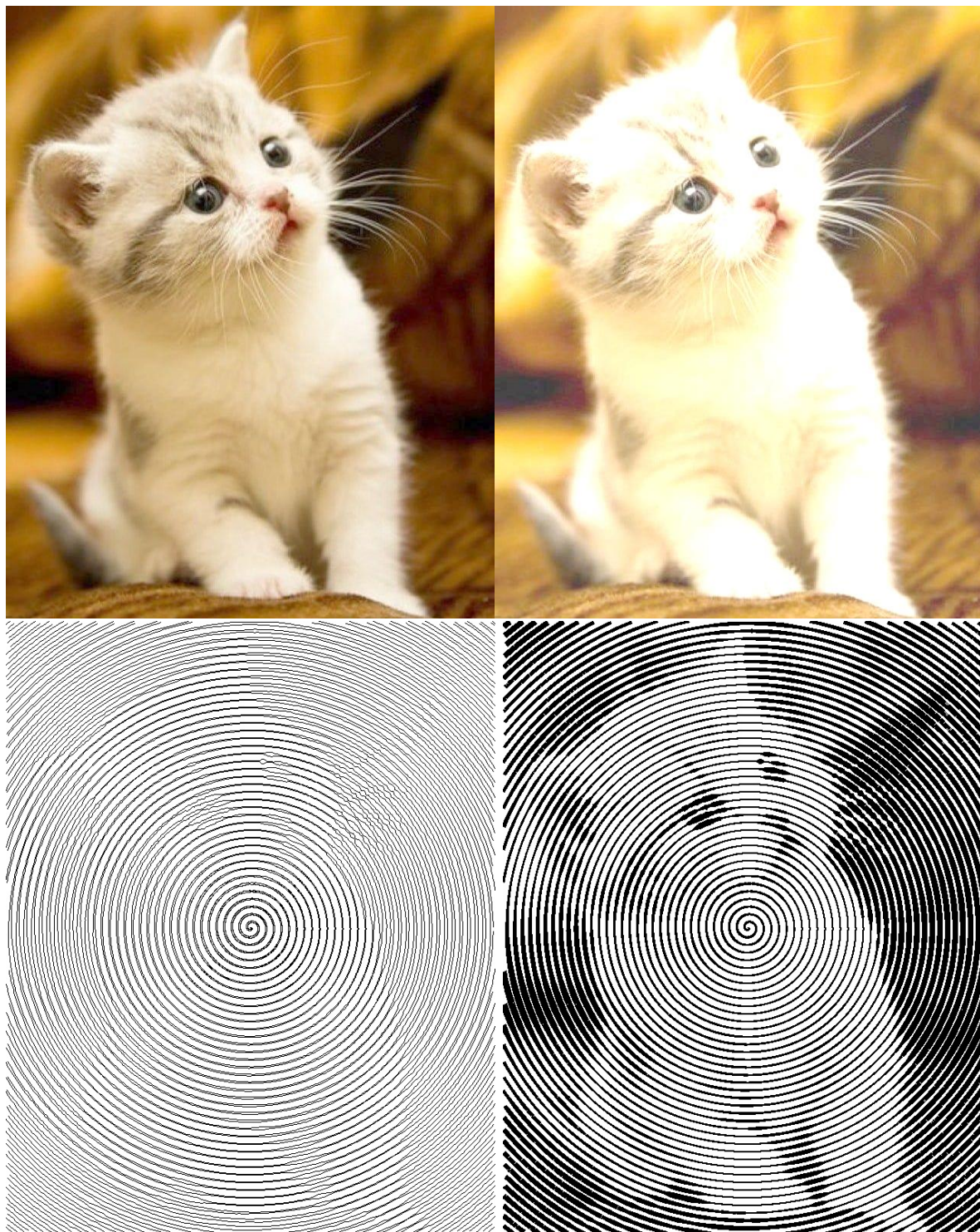


Хоть такие изображения тоже являются достаточно красивыми, они не являются желаемым результатом из-за образовавшихся артефактов. Невооруженным глазом видно, что контрастирование в более яркую сторону помогает избежать различные артефакты и может использоваться для повышения качества конечного изображения.

Ширина спирали 5px, контрастирование – да

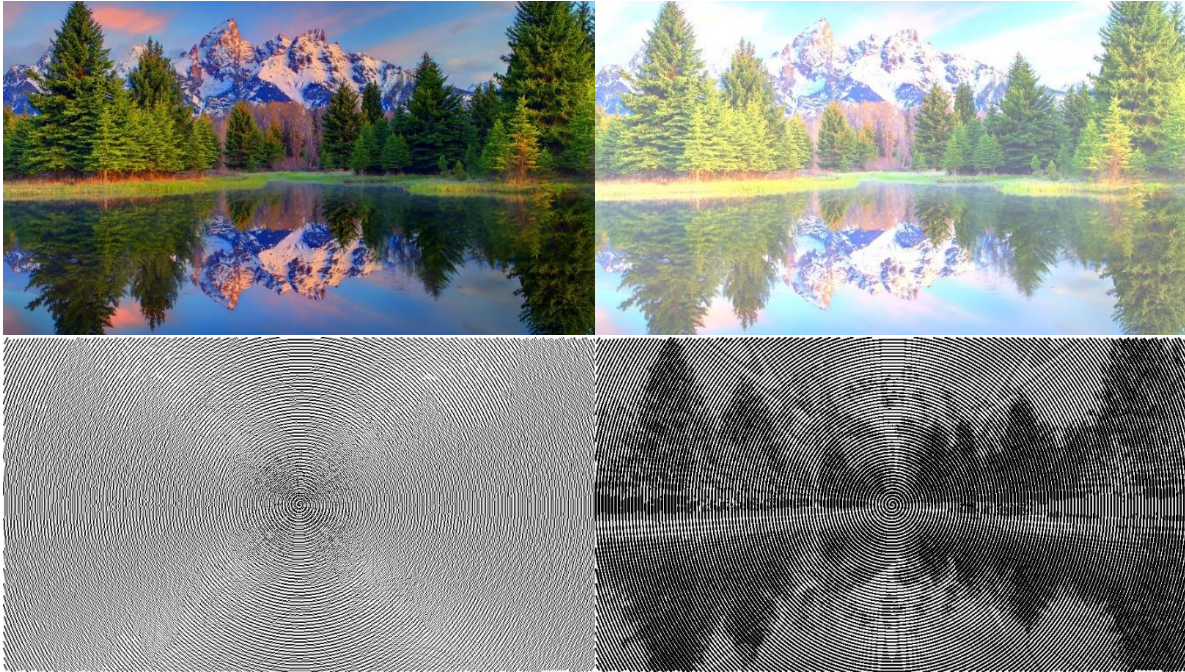


Ширина спирали 10px, контрастирование – да

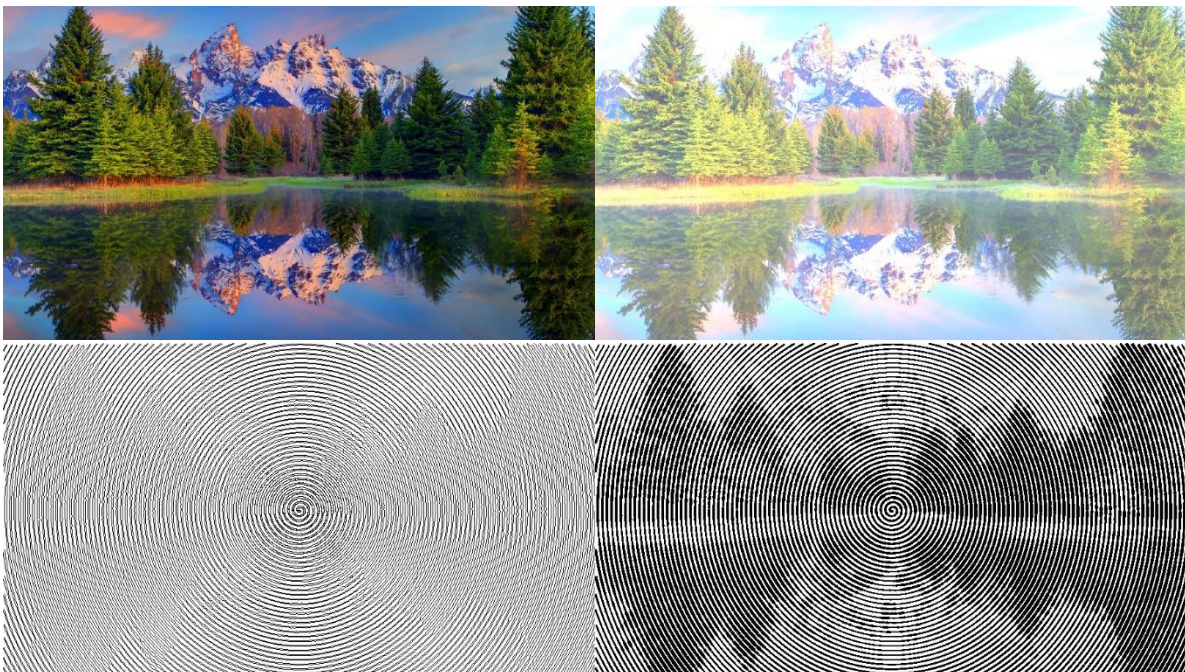


Как видно, самые четкие контуры и хорошо разборчивые силуэты получаются при отсутствии контрастирования либо когда уменьшается расстояние между спиралями. Но обратная сторона заключается в том, что появляются негативные артефакты. Поэтому приходится выбирать между “широкими” спиралями и большой потерей информации об изображении, но без различных пятен либо хорошие контуры но наличие пятен.

Ширина спирали 4px, контрастирование – да, пейзаж с большим количеством маленьких объектов



Ширина спирали 7px, контрастирование – да, пейзаж с большим количеством маленьких объектов



Видно, что в данном случае, когда имеется большое количество маленьких и удаленных объектов – меньших чем шаг спирали. То в таком случае маленький шаг наоборот делает картинку разборчивее.

Руководство по запуску

- 1) Установите репозиторий <https://github.com/matyushkovvv/Image-Processing>
- 2) Установите OpenCV через vcpkg
- 3) Соберите проект
- 4) `./x64/Debug/coursework.exe <path_image> <spiralWidth>`
 - Path_image – путь к изображению
 - SpiralWidth – шаг спирали

Выводы

В ходе выполнения курсовой работы по теме "Создание спиральных раскрасок на основе произвольных изображений" была разработана программа на языке C++ с использованием библиотеки OpenCV, которая успешно преобразует входные изображения в формат спиральной раскраски. В процессе выполнения работы были достигнуты следующие результаты:

1. Реализован алгоритм преобразования изображений:

- Разработан и внедрен алгоритм создания спиральных контуров, адаптированных к контрасту исходного изображения. Алгоритм учитывает яркость пикселей для изменения толщины спиральных линий, что позволяет получить визуально привлекательный и разнообразный результат.

2. Использование OpenCV:

- Эффективно применены функции библиотеки OpenCV для загрузки, обработки и анализа изображений.

3. Интеграция CMake:

- Для удобства разработки и сборки проекта использовался инструмент CMake, который обеспечил кроссплатформенную сборку и управление зависимостями. Среда разработки Visual Studio 2022 предоставила удобный интерфейс и интеграцию с Git для контроля версий.

4. Тестирование и оптимизация:

- Программа была протестирована на различных изображениях, что подтвердило её работоспособность и эффективность. Были внесены оптимизации для улучшения производительности и качества получаемых изображений.

5. Практическое применение:

- Созданные спиральные раскраски могут быть использованы в различных областях, включая художественное творчество. Программа предоставляет пользователю интересный инструмент для создания уникальных визуальных эффектов.

Данная курсовая работа продемонстрировала возможности применения методов и средств обработки изображений для решения творческих задач. Полученные результаты показывают, что разработанная программа может быть расширена и улучшена, например, за счет добавления дополнительных параметров настройки спиралей или интеграции новых алгоритмов анализа изображений.