

Содержание	
Введение	4
1. Техническое задание	5
1.1. Назначение разработки и область применения	5
1.2. Технические требования	5
2. Анализ технического задания	6
2.1. Выбор операционной системы	6
2.2. Выбор языка программирования	7
2.3. Выбор среды разработки	9
2.4. Обзор существующих систем склейки изображений	10
2.5. Выбор подхода к решению задачи склейки изображений	12
3. Разработка структуры системы склейки изображений	14
3.1. Разработка общей структуры системы	14
3.2. Разработка алгоритма вычисления особых точек изображений	16
3.3. Разработка алгоритма принятия решения	17
4. Разработка программных средств	18
4.1. Разработка интерфейса пользователя системы	18
4.2. Программная реализация модулей системы	20
5. Тестирование системы	25
5.1. Описание набора данных	25
5.2. Описание методики тестирования	28
5.3. Результаты вычислительного эксперимента	31
Заключение	34
Список литературы	35

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Крахалева Т.И.			Система создания бесшовных изображений		Лит.	Лист
Провер.		Гай В.Е.						Листов
Реценз.								
Н. Контр.							3	35
Утверд.							НГТУ им. Р.Е.Алексеева	

Введение

В век высоких технологий все чаще используются системы для создания бесшовных изображений. Эти системы часто используются для ориентирования, то есть анализа движения земной коры, изменений рельефа, исследования перемен погоды, застроек, проведение спасательных операций практически невозможно без них. Сейчас сложно вообразить жизнь без доступного использования карт местности, улиц городов, GPS. Актуальными являются и портативные средства (смартфоны, автомобильные бортовые компьютеры) выстраивающие кратчайший маршрут до пункта назначения или определяющий ваше местоположение на карте [1].

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

1. Техническое задание

1.1. Назначение разработки и область применения

Разрабатываемая система предназначена для создания из отдельной пары фотографий единого бесшовного изображения.

Области применения разрабатываемой системы:

1) Дистанционное зондирование земли с беспилотного летательный аппарата и самолетов

2) Создание подробных карт, панорам улиц и достопримечательностей

3) Средства географического ориентирования (gps и ГЛОНАСС)

Данная система предназначена для использования на портативных и стационарных компьютерах.

1.2. Технические требования

Опишем требования, предъявляемые разрабатываемой системой к ЭВМ:

1) операционная система Microsoft Windows XP и выше

2) требования к аппаратному обеспечению определяются операционной системой

3) мышь, дисплей, клавиатура

Рассмотрим, каким функционалом должна обладать разрабатываемая система создания бесшовного изображения:

1) Система должна предоставить возможность пользователю выбрать изображения для склейки и загрузить их для обработки.

2) Реализовать поиск связующих точек между изображениями.

3) Вывести на экран «сшитое» изображение.

2. Анализ технического задания

2.1. Выбор операционной системы

На этапе начала разработки важным является выбор операционной системы, так как каждая из них имеет свои особенности, влияющие на разработку. Рассмотрим наиболее распространенные операционные системы: Mac OS, Linux, Windows.

1) Windows – семейство операционных систем корпорации Microsoft, ориентированных на применение графического интерфейса при управлении. Система обладает интерфейсом дружелюбным для пользователя, благодаря чему получила широкое распространение.

Microsoft представила операционную среду под названием Windows, 20 ноября 1985 года, в качестве графической оболочки операционной системы для MS-DOS, в ответ на растущий интерес к графическим пользовательским интерфейсам. Microsoft Windows стал доминировать в мире персональных компьютеров с более чем 90% рынка, обогнав Mac OS [1].

Существует несколько версий ОС Windows, в зависимости от задач на нее возлагаемых: для предприятий (больше возможностей обеспечения безопасности, управление сетевыми ресурсами), для персональных компьютеров, для планшетов, смартфонов, встраиваемых устройств. Последней версией ОС Windows для персональных компьютеров, планшетов, смартфонов и встраиваемых устройств является ОС Windows 10. Самой последней версией для серверных компьютеров является Windows Server 2016 [1].

2) Mac OS – операционная система, разработанная Apple Inc. Предназначена для использования на серии персональных компьютеров Macintosh с 1984 года. Является второй по распространенности операционной системой. Главным отличием от Windows считается то что Mac OS основана на системе Unix [3].

3) Linux – название семейства Unix – подобных операционных систем на основе ядра linux и включающих в себя библиотеки и системные программы проекта GNU. Особенностью является открытость кода. Также Linux создается и распространяется как свободное и открытое ПО, посредством готовых дистрибутивов, настроенных для решения определенных задач [4].

Для создания системы создания бесшовных изображений была выбрана система Windows, так как она данная операционная система обладает наибольшим выбором программного обеспечения для разработки.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

2.2. Выбор языка программирования

Выбор языка программирования так же является важным этапом разработки приложения. Для разработки нами была выбрана операционная система Windows для которой можно выделить такие языки программирования как: C++, Java, Python, R и т.д.

1) C++ - является языком программирования общего назначения. Он имеет процедурные, объектно-ориентированные и общие программные функции, а также включает средства высокоуровневых и низкоуровневых языков. C++ широко используется: для создания драйверов, операционных систем, различных прикладных программ. Синтаксис является наследником языка C и полностью совместим с ним [5].

2) Java – объектно-ориентированный язык программирования, разработанный в 1991 году компанией Sun Microsystems. Изначально имел название Oak и использовался для бытовой электроники. Изначально для этой цели использовался язык C++, но стало очевидно, что нужен платформо-независимый язык, который позволит создать программы, которые можно было бы использовать на различных процессорах под различные ос. Впоследствии использовался для написания приложений, апплетов и серверного ПО, и был переименован в Java.

Программы, написанные на Java транслируются в байт-код и выполняются на виртуальной машине – JVM, обрабатывающей байт-код и передающей инструкции оборудованию. В отличие от текста байт-код обрабатывается намного быстрее, что является достоинством [6].

3) Python – высокоуровневый язык программирования общего назначения. Поддерживает часть парадигм программирования таких как объектно-ориентированное, структурное, функциональное программирование. Код организован в функции и классы, объединяющиеся в модули. Также является скриптовым языком и исходный код не компилируется, а выполняется с помощью интерпретатора [7].

4) R – язык программирования, использующийся для статистической обработки данных и работы с графикой. В R реализованы многие статистические методы (линейные и нелинейные модели, классификация, проверка статистических гипотез). Есть возможность реализации функций на C, C++, для более сложных вычислительных задач [8].

Преимуществом R является наличие множества расширений и пакетов. Базовые пакеты присутствуют сразу после установки R, при наличии интернета дополнительные пакеты можно установить прямо из R командой `install.packages()`. На 2016 год насчитывается около 8000 пакетов.

Для разработки программной системы был выбран язык R, так как в нем реализованы необходимые нам библиотеки для работы с графическими файлами, такие как:

`library('jpeg')` – этот пакет обеспечивает легкий и простой способ чтения, записи и отображения растровых изображений, сохраненные в формате JPEG. Она может читать и сохранять как файлы, так и исходные векторы.

library('magick') – Magick пакет является амбициозной попыткой модернизировать и упростить обработку изображений высокого качества в R. Он основывается на STL ImageMagick, который является наиболее обширной библиотекой обработки изображений, с открытым исходным кодом, доступным сегодня.

Библиотека ImageMagick имеет подавляющий объем функциональности. Поддерживает множество распространенных форматов (PNG, JPEG, TIFF, PDF и т.д.) и манипуляций (вращение, масштабирование, обрезка, обрезка, зеркальное отражение, размытие, и т.д.). Все операции векторизации STL ImageMagick работают либо на одном кадре, либо серии кадров: для работы со слоями, коллажами, или анимацией [9].

Для создания графического интерфейса был выбран язык Java, ввиду простоты использования и наличия множества библиотек, обеспечивающих взаимодействие с R.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

2.3. Выбор среды разработки

Существует множество сред разработки на языке Java:

1) NetBeans – среда разработки с открытым исходным кодом, ориентированная на интернет, мобильные и настольные приложения.

2) IntelliJ IDEA – имеет поддержку фреймворков Spring (Spring MVC framework, Spring Security, Spring Boot, Spring Integration и т. д.), Node.js, Angular React, Grails, возможность использования дополнительных языков (javascript, typescript, coffeescript) и взаимодействует почти со семи популярными серверами.

3) Eclipse – поддерживает несколько языков, но используется как приверженец Java.

4) Android Studio – помимо всех возможностей, которые дает исходная IDE IntelliJ IDEA, Android Studio включает в себя множество надстроек от Google, как чисто визуальных (макеты, форматы, GPU профайлер), так и функциональных (JUnit 4 и Firebase Test Lab для тестирования и отладки, система сборки Gradle, Instant Run).

Самой распространенной интегрированной средой разработки считается Eclipse, который и был выбран нами, в связи с наличием огромного количества информации, удобством использования и бесплатностью [10].

Изначально работа с R ведется из командной строки. В консольном интерфейсе пользователю предоставляется история команд, список возможных продолжений команд. Есть возможность сохранять информацию и объекты между сессиями. Графическая оболочка расширяет базовые возможности работы на языке R благодаря наличию встроенного редактора кода с подсветкой, редактора массива данных, отладчика.

Для разработки блока обработки изображений была выбрана интегрированная среда разработки RStudio, сочетающая в себе интуитивный пользовательский интерфейс с мощной консолью R, бесплатность. Также считающаяся одной из самых популярных сред разработки.

2.4. Обзор существующих систем склейки изображений

Задача создания бесшовных изображений не является сравнительно новой, существует несколько наработок по данной теме, различные подходы и алгоритмы [1]. Чаще всего в специализированных системах нахождение связующих точек состоит из:

1) Разбиение изображений на области с примерно одинаковыми фотометрическими характеристиками.

2) Полученные области переводятся в приближенные контурные фигуры

3) Для каждой пары перекрывающихся изображений вычисляются коэффициенты корреляции определенным образом выбранного подмножества пар фигур.

4) С помощью оптимизированного переборного алгоритма, использующего рассчитанный коэффициент корреляции, а также дополнительные геометрические особенности расположения фигур, производится окончательное сопоставление фигур пары снимков и вычисление приблизительного соответствия точек на паре изображений.

5) Результаты этапа 4 поступают на вход иерархическому площадному коррелятору, с помощью которого полностью автоматически вычисляется необходимое число точек. После этого применяются различные критерии отбора, тем самым исключаются потенциально ошибочные точки.

6) Расположение фотографий согласно полученным расчетам.

HUGIN Software.

Программа позволяет производить множество сложных преобразований: создание мозаики из перекрывающихся изображений и особенно панорам, нахождение контрольных точек, HDR [1].

Пользователь обычно имеет дело только с графической надстройкой над группой программ, которые и осуществляют процесс сшивки из нескольких кадров единой панорамы. Основные из этих программ:

1) Panorama Tools и входящая в них библиотека Libpano 12. Основная библиотека, которая используется для преобразования проекций и коррекции искажений.

2) pona — осуществляет геометрические и фотометрические преобразования исходных снимков, создает и записывает в файл с расширением «*.pto» сценарий необходимых действий.

3) srfind — приложение Hugin для поиска и расстановки контрольных точек.

4) Enblend сводит изображения в бесшовные панорамы.

5) Enfuse объединяет сделанные с вилкой экспозиции фотографии. При одинаковой экспозиции снимков на входе Enfuse объединяет их зоны резкости.

Результат работы программы представлена на рисунке 2.1.



Рис 2.1. Результат работы программы

Ceres Solver.

Открытая библиотека компании Google написанная на языке C++ с реализацией функций для решения нелинейных уравнений методом наименьших квадратов. Проект отличается высокой масштабируемостью и позволяет обеспечить оптимальное использование ресурсов, при выполнении как на мобильных телефонах, так и на крупных серверах [11].

Google данный код использует в системе Google Maps для формирования трёхмерного представления объектов по плоским спутниковым снимкам и для создания реалистичных моделей Street View. Библиотека может быть полезной для реконструкции 3D – объектов по нескольким фотографиям, снятым под разным углом (например, снимки из машины, самолёта или спутника)

PHOTOMOD

Первая разработанная в России цифровая фотограмметрическая система. Применяется для фотограмметрической обработки данных дистанционного зондирования Земли. Это единственная поддерживаемая в актуальном состоянии ЦФС в России, не уступающая зарубежным аналогам [1].

2.5. Выбор подхода к решению задачи склейки изображений

Для решения задачи склейки изображения была выбрана теория активного восприятия изображения [12].

Эта теория подразумевает реализацию двух этапов системы распознавания образов: разработка алгоритма предварительной обработки данных и формирование системы признаков см. рис. 2.2.

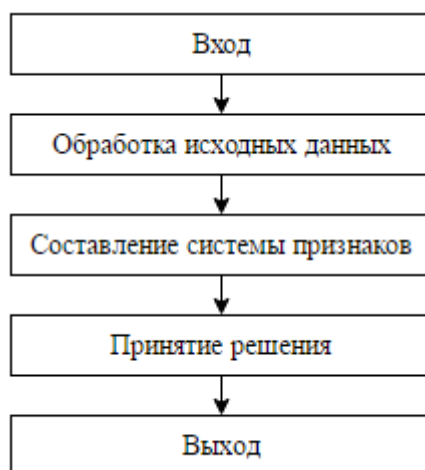


Рис 2.2. Алгоритм обработки данных и формирование системы признаков

Предварительная обработка данных, в данной теории понимается, как операция интегрирования, а составление системы признаков понимается, как операция дифференцирования.

В первом случае результатом операции интегрирования будет являться множество «визуальных масс», данное преобразование определено, как Q – преобразование.

Во втором случае результатом операции дифференцирования будет являться вектор «спектральных коэффициентов» $\mu = (\mu_0, \mu_1, \mu_2, \dots, \mu_{15})$. Операция дифференцирования реализуется по средствам 16 фильтров (рис.2.3), где темный элемент «-1», светлый элемент – «+1».

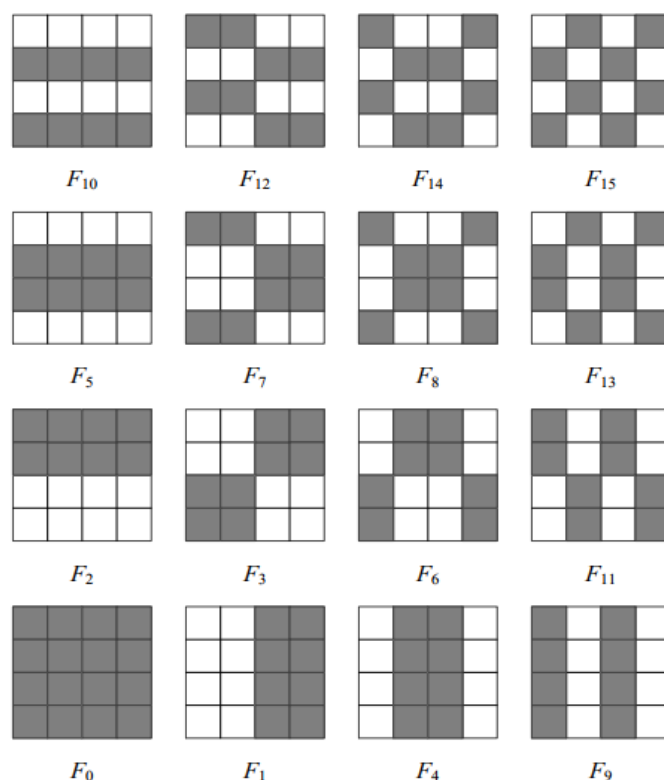


Рис. 2.3. 16 фильтров

Наблюдается схожесть фильтров теории активного восприятия с фильтрами Уолша, однако они различаются алгоритмом применения к изображению. Фильтры теории активного восприятия изображения используются только после выполнения преобразования интегрирования (Q-преобразования).

Пара преобразований, интегральное и дифференциальное, вместе составляют U – преобразование см. рис. 2.4.

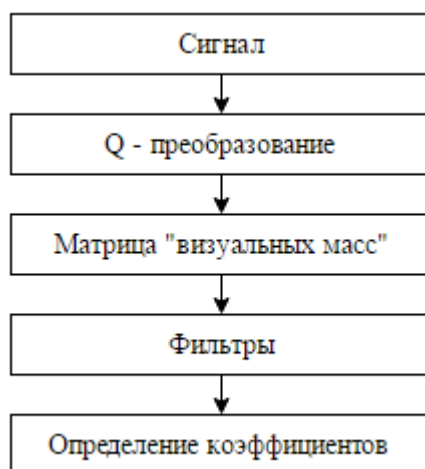


Рис. 2.4. U – преобразование

3. Разработка структуры системы склейки изображений

3.1. Разработка общей структуры системы

Программную систему склейки изображений можно представить в виде следующих модулей:

1) Программа создания итоговых изображений представлена на рис. 3.1.

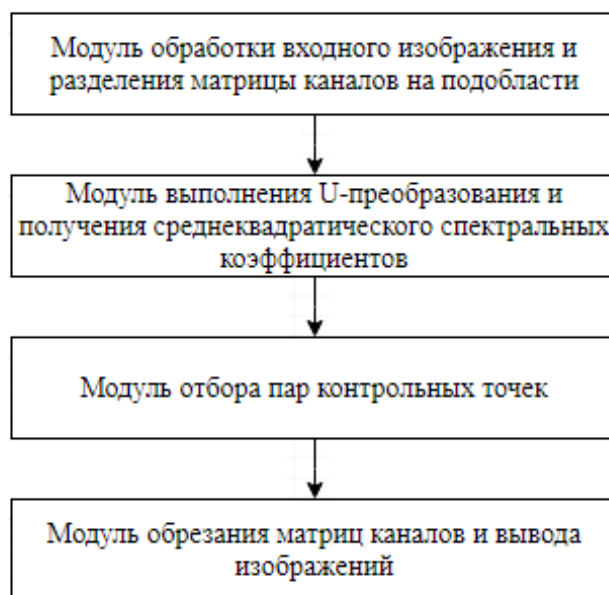


Рис. 3.1. Алгоритм основной программы

Рассмотрим данные модули.

1.1) Модуль обработки входного изображения и разделения матрицы каналов на подобласти: программа получает на вход изображение формата jpeg и преобразовывает его в матрицу коэффициентов каналов изображения. Разделяем начальную матрицу на подобласти, так чтобы получилось 1600 подобластей.

1.2) Модуль выполнения U – преобразования и получения среднеквадратического спектральных коэффициентов: с помощью U – преобразования получаем матрицу спектральных коэффициентов, с помощью которой находим среднеквадратическое значение от 15 спектральных коэффициентов для каждой подобласти и объединяем все значения в одну матрицу среднеквадратических значений.

1.3) Модуль отбора пар контрольных точек: находим пары, которые имеют самые большие значения. Разница парных значений должна быть не больше 0.1. Выбирается та пара, которая находится ближе к краям изображений. Сравниваем среднеквадратические значения, которые находится на 5 вверх и на 5 вниз от выбранных парных значений. То есть 5 значений выше, и 5 значений ниже от выбранных парных значений левого и правого изображений, не должны различаться более, чем на 0.1.

1.4) Модуль обрезания матриц каналов и вывода изображений: вычисляем в каком столбце матрицы каналов находятся парные точки на левом и правом изображениях. Удаляем все столбцы, включая столбец с парной точкой из

матрицы каналов левого изображения. Удаляем все столбцы, до столбца с парной точкой из матрицы каналов правого изображения.

2) Клиентская программа см. рис. 3.2.

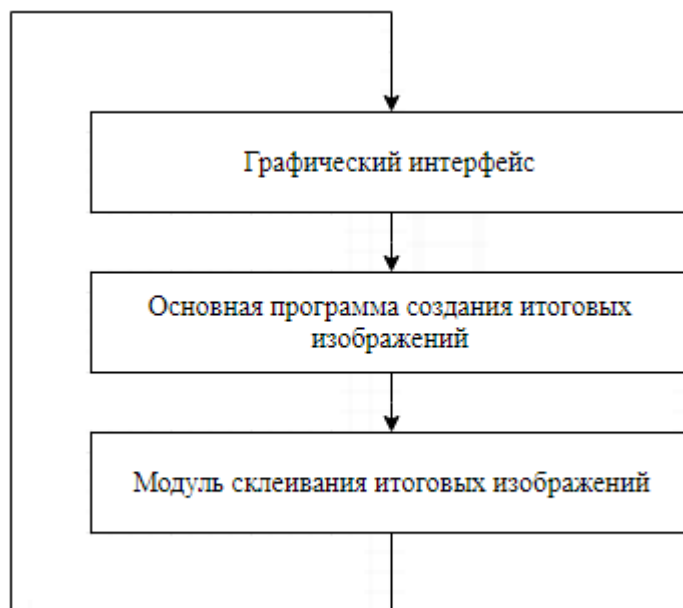


Рис 3.2. Алгоритм работы клиентской программы.

Пользователь средствами клиентской программы может загрузить 2 изображения, для склейки, а также просмотреть результат выполнения программы непосредственно в графической оболочке. Результатом работы является выведенное на экран склеенное изображение.

Так как необходимо отобразить результат работы в графической оболочке, итоговое изображение сохраняется на диск, в директории с программой, для дальнейшей работы с ним в клиентской части.

3.2. Разработка алгоритма вычисления особых точек изображений

Рассмотрим блок-схему работы системы вычисления особых точек для склейки изображений см. рис. 3.3.



Рис 3.3. Алгоритм вычисления особых точек

Программным путем из загруженного изображения получаем начальную матрицу каналов. Далее переводим начальную матрицу в градации серого путем сложения R, G, B каналов и нормализуем ее. Начальную матрицу разделяем на подобласти, так чтобы получилось 1600 подобластей и приводим каждую к размеру 4×4 . Применяем 15 фильтров, получаем матрицу спектральных коэффициентов. Находим среднеквадратическое значение от 15 спектральных коэффициентов для каждой подобласти. Объединяем в матрицу среднеквадратических значений.

3.3. Разработка алгоритма принятия решения

Для склеивания изображений нам нужно найти пары ключевых точек на изображениях, по которым будет происходить склейка. Рассмотрим алгоритм принятия решений при поиске ключевых точек см. рис. 3.4.



Рис 3.4. Алгоритм принятия решений

Необходимо найти пару среднеквадратических значений, удовлетворяющих следующим условиям по следующему алгоритму:

- 1) Находим пары, имеющие наибольшие значения
- 2) Разница парных значений должна быть не больше 0,1
- 3) Выбирается та пара, которая находится ближе к краям изображений. То есть на левом изображении ближе к правому краю, на правом изображении ближе к левому краю.

4) Сравниваем среднеквадратические значения, которые находятся на 5 вверх и на 5 вниз от выбранных парных значений. То есть 5 значений выше, и 5 значений ниже от выбранных парных значений левого и правого изображений, не должны различаться более, чем на 0.1.

4. Разработка программных средств

4.1. Разработка интерфейса пользователя системы

Разработка интерфейса пользователя проводилась на языке Java. Использовалась библиотека Java Swing [13]. Рассмотрим примененные классы и методы данной библиотеки:

1) Класс JFrame – представляет пустое окно с рамкой, строкой заголовка и кнопками «Свернуть», «Во весь экран», «Заккрыть». Есть возможность менять его размеры и перемещать по экрану.

2) Класс JButton – кнопка. Имеются различные методы для ее конфигурации. Использовался метод «setSize» для изменения размеров кнопки и метод «setLocation» для установки позиции внутри окна «JFrame».

3) JLabel – обычный текст. Выводится в заданном месте окна и используется для вывода вспомогательной текстовой информации: подписи к другим элементам, инструкции и предупреждения для пользователя внутри «JFrame». Используются методы аналогичные «JButton».

4) Класс JFileChooser – компонент библиотеки Java Swing используемый для создания диалога выбора файлов, навигации по файловой системе см. рис. 5.2. Рассмотрим программную реализацию см. рис. 4.1.

```
chooseButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        JFileChooser fileopen = new JFileChooser();
        int ret = fileopen.showDialog(null, "Открыть файл");
        if (ret == JFileChooser.APPROVE_OPTION) {
            File file = fileopen.getSelectedFile();
            getFilef = file.getAbsolutePath();
            label.setText(getFilef);
            try(FileWriter writer = new FileWriter("ff.txt", false))
            {
                writer.write(getFilef);
            }
            catch(IOException ex){
                System.out.println(ex.getMessage());
            }
        }
    }
});
```

Рис.4.1. Программная реализация «JFileChooser».

В первой строке создается новый экземпляр класса «JFileChooser». С помощью второй строки отображается диалог пользователю. В диалоге пользователь выбирает необходимый файл. С помощью функции «getAbsolutePath» путь к файлу присваивается строковой переменной «getFile», для последующей записи в промежуточный текстовый документ «ff.txt», который будет использован основной программой поиска.

5) Метод addActionListener – используется для обработки нажатий на кнопки. Пример обработчика нажатий для кнопки выхода из программы представлен на рис. 4.2.

```
exitButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});
```

Рис. 4.2. Обработчик нажатий.

6) Класс JOptionPane – предназначен для вывода в графический интерфейс различной информации и выбора простых данных.

Был использован метод «showMessageDialog» принадлежащий этому классу, для вывода результата работы программы в виде изображения.

Программная реализация представлена на рис. 4.3.

```
JOptionPane.showMessageDialog(null, icon, "Склеенное изображение", JOptionPane.PLAIN_MESSAGE);
```

Рис. 4.3. Вывод результата работы.

4.2. Программная реализация модулей системы

4.2.1. Модуль обработки входного изображения и разделения матрицы каналов на подобласти

С помощью функции «readJPEG» библиотеки «JPEG» языка R создаем матрицу каналов для первого и второго изображений.

Переводим изображения в градации серого путем сложения RGB – каналов с помощью функции:

```
if (length(dim(img)) == 3)
{
  img <- img[, , 1] + img[, , 2] + img[, , 3];
  img <- img / 3;
}
```

Нормируем матрицы каналов:

```
img <- img - min(img)
img <- img / max(img)
```

Для разделения матриц каналов получаем массив, содержащий границы каждой подобласти, после чего с помощью данного массива выполняем разделение матриц каналов и заполнения массива подобластей.

Заполнение массива подобластей:

```
for (j in seq(1,80,2))
{
  for (i in seq(1,80,2))
  {
    matr1 <- img[swF[i]:swF[i + 1], shF[j]:shF[j + 1]]
    listOfMassF[[j1F]] <- list(matr1)
    j1F = j1F + 1
    k = k + 1
  }
  k = k + 1
}
```

4.2.2. Модуль выполнения U – преобразования и получения среднеквадратического спектральных коэффициентов

Выполняем Q – преобразования с каждой полученной матрицей подобластей. В результате преобразований получаем массив матриц «визуальных масс» «listMassF».

```
for (j in seq(1,8,2))
{
  for (k in seq(1,8,2))
  {
    dt <- listProm[swF1[j]:swF1[j], shF1[k]:shF1[k]]
    mass[k1F,j1F] = sum(dt)
    listMassF[[i]] <- list(mass)
    j1F = j1F + 1
  }
  j1F = 1
  k1F = k1F + 1
}
```

С помощью операции интегрирования получаем матрицы спектральных коэффициентов «spectrFirstFin» для каждой подобласти для обоих изображений.

```
for(j in 1:1600)
{
  listPredFirst<-matrix(unlist(listMassF[j]), nrow=4,ncol=4)
  for (i in 1:16)
  {
    spectrFirst[i]<-sum(flt[[i]] * listPredFirst)
  }
  spectrFirstFin[[j]]<-
list(matrix(spectrFirst,nrow=4,ncol=4))
}
```

Следующим этапом является получение матрицы среднеквадратических значений «listSDMatr» для каждого изображения.

```
for(i in 1:1600)
{
  for(j in 1:15)
  {
    listVal1 <- unlist(spectrFirstFin[[i]])
  }
  listSD[[i]] <- sd(listVal1)
}
listSDMatr <- matrix(unlist(listSD),nrow = 40, ncol = 40)
```

4.2.3. Модуль отбора пар контрольных точек

В данном модуле отбираются парные точки изображений. Выбираются максимальные среднеквадратические значения, отклонения в значении которых менее 0.1, а также сравниваем среднеквадратические значения, которые находятся на 5 вверх и на 5 вниз от выбранных парных значений. То есть 5 значений выше, и 5 значений ниже от выбранных парных значений левого и правого изображений, не должны различаться более, чем на 0.1.

После получения матрицы среднеквадратических значений «listSDMatr» необходимо найти максимальные значения. Максимальные значения первого изображения записываются в массив «maxFirst», позиции этих значений записываются в массив «maxFirstPos». Соответственно, максимальные значения второго изображения записываются в массив «maxSecond», а их позиции в «maxSecondPos».

```
for(i in 1:1600){  
  if(listSDMatr[[i]] > maxFirst[[1]]){  
    maxFirst[[1]] <- listSDMatr[[i]]  
    maxFirstPos[[1]] <- i  
  }  
}
```

После нахождения максимальных значений необходимо их сравнить. Удовлетворяющие значения не должны различаться более чем на 0.1, на то же число не должны различаться 5 значений выше и 5 значений ниже первого и второго изображений, относительно сравниваемых значений. В результате выполнения условий получаем несколько пар значений, положения которых записываются в массив «PosPairs».

4.2.4. Модуль обрезания матриц каналов и вывода изображений

Задаем параметры новых изображений:

Инициализация переменных для сохранения номера столбца, по которому будет происходить обрезание изображения

```
minVal1 <- posPairs[[1]]  
minVal2 <- posPairs[[2]]  
minLast <- posPairs[[2]]
```

Для обрезания выбираем ту пару, в которой значение столбца второго изображения самое маленькое

```
for(i in seq(1, 40, 2)){  
  if(posPairs[[i + 1]] < minLast & posPairs[[i]] > 0)  
  {  
    minSumPos <- i  
    minVal1 <- posPairs[[i]]  
    minVal2 <- posPairs[[i + 1]]  
    minLast <- posPairs[[i + 1]]  
  }  
}
```

Вычисление размеров новых изображений

```
newSizeF <- h1 * (minVal1 %% 40 + 1)  
newSizeS <- (h2 * (minVal2 %% 40 + 1))  
img <- image_read(location)  
img2 <- image_read(location2)
```

Запись размеров новых изображений

```
imgSzF <- paste(newSizeF, "x", hF, sep="")  
imgSzS <- paste(hS, "x", wS, "+", newSizeS, sep="")  
#Обрезание изображений до новых параметров  
newMatrixF <- image_crop(img, imgSzF)  
newMatrixS <- image_crop(img2, imgSzS)
```

Сохранение итоговых изображений, готовых для склейки:

```
image_write(newMatrixF, path = "new1.jpg", format = "jpg")  
image_write(newMatrixS, path = "new2.jpg", format = "jpg")  
}
```

4.2.5. Модуль склейки изображений

Создаем два экземпляра класса «File» для дальнейшего открытия 1 и 2 изображений и их склейки

```
File one = new File("new1.jpg");  
File two = new File("new2.jpg");
```

Создаем экземпляры класса «BufferedImage» и присваиваем им переменные типа «File» для склейки изображений

```
BufferedImage img1 = ImageIO.read(one);  
BufferedImage img2 = ImageIO.read(two);
```

Создаем новое изображение, ширина которого равна сумме двух входных изображений.

```
BufferedImage img = new BufferedImage(img1.getWidth() +  
img2.getWidth(), Math.max(img1.getHeight(), img2.getHeight()),  
BufferedImage.TYPE_INT_ARGB);
```

Рисуем изображение.

```
Graphics g = img.createGraphics();  
g.drawImage(img1, 0, 0, null);  
g.drawImage(img2, img1.getWidth(), 0, null);  
g.dispose();
```

Сохраняем новое изображение на диск

```
ImageIO.write(img, "png", new File("newimg.png"));
```

5. Тестирование системы

5.1. Описание набора данных

Для создания качественных панорамных снимков лучше использовать фотографии сделанные со штатива, с одинаковой выдержкой и освещенностью, чтобы избежать смазывания итогового изображения. Разработанная система позволяет склеить только два изображения одинакового размера, без поворотов.

Ниже приведены примеры изображений:

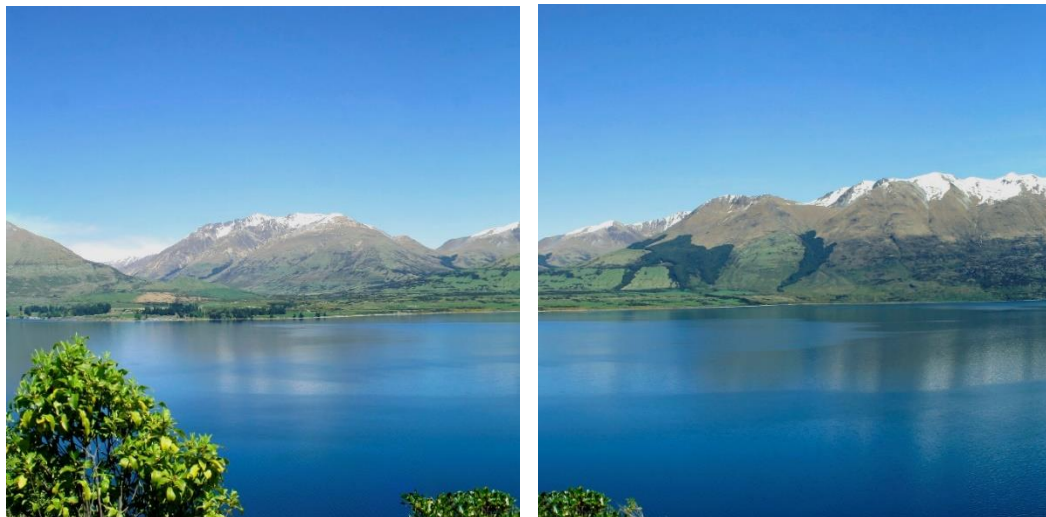


Рис. 5.1. Тестовая пара изображений 1



Рис. 5.2. Тестовая пара изображений 2



Рис. 5.3. Тестовая пара изображений 3

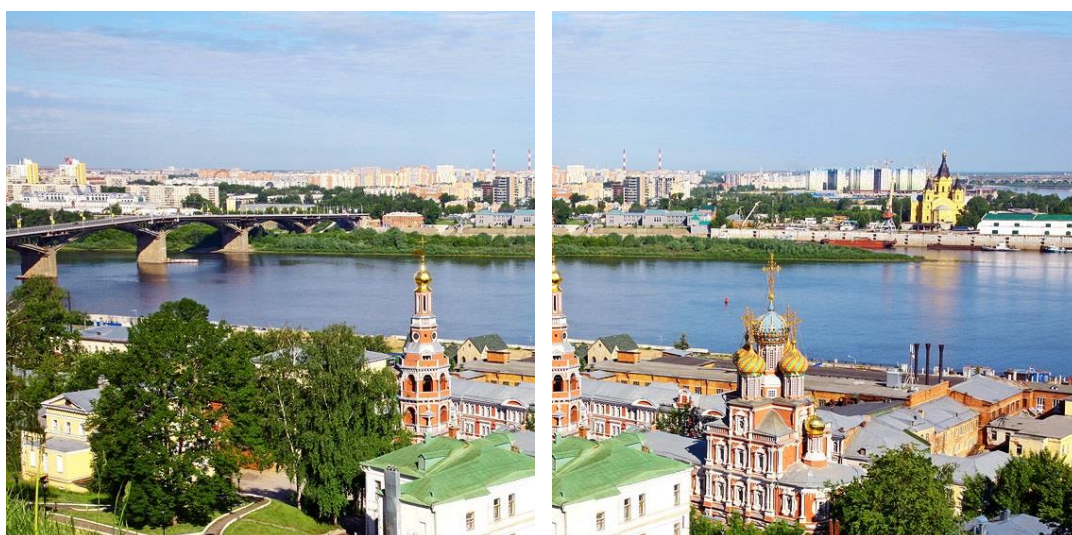


Рис. 5.4. Тестовая пара изображений 4

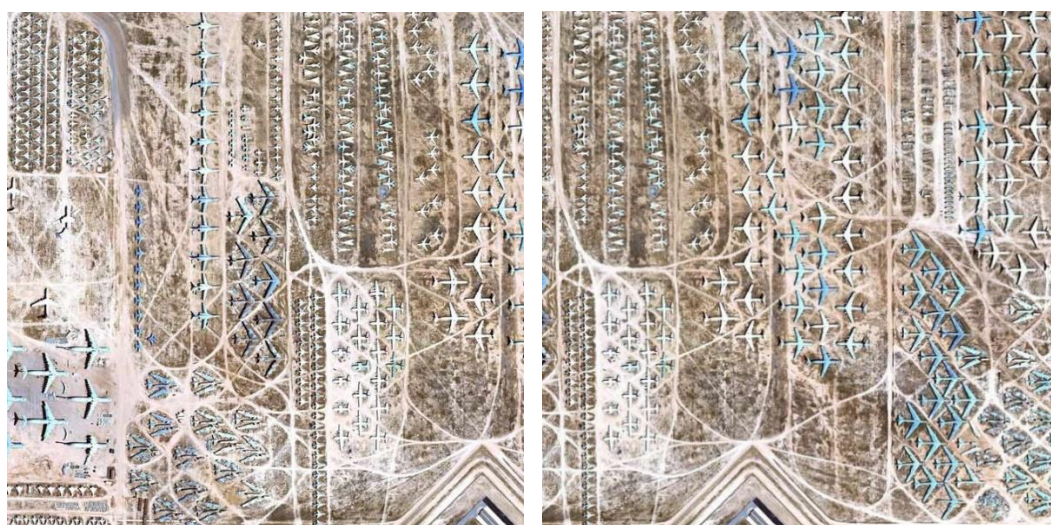


Рис. 5.5. Тестовая пара изображений 5



Рис. 5.6. Тестовая пара изображений 6

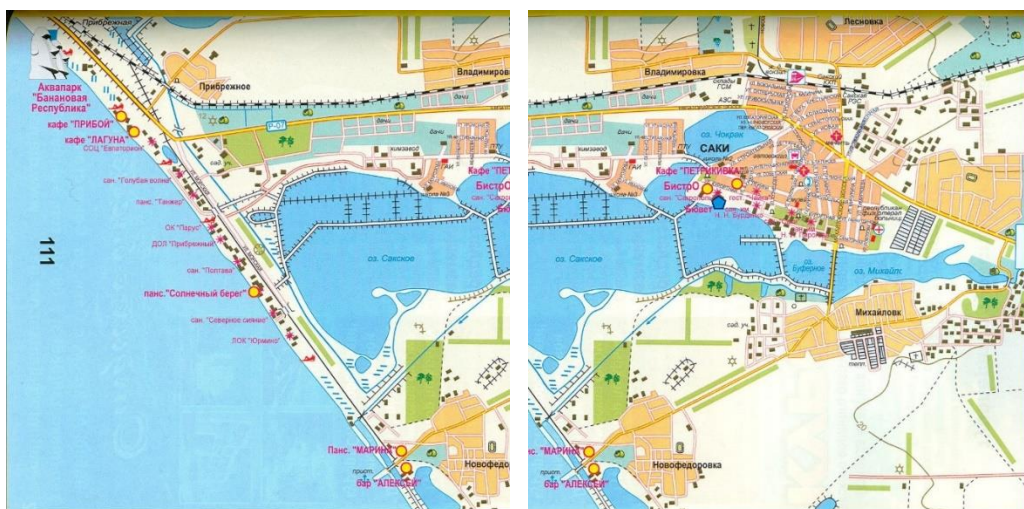


Рис. 5.7. Тестовая пара изображений 7

5.2. Описание методики тестирования

Проверим работоспособность программы. Для этого запустим пользовательский интерфейс загрузим изображения тестовой пары 1, посмотрим результат склейки.

Пользовательский интерфейс представлен на рисунке 5.8.

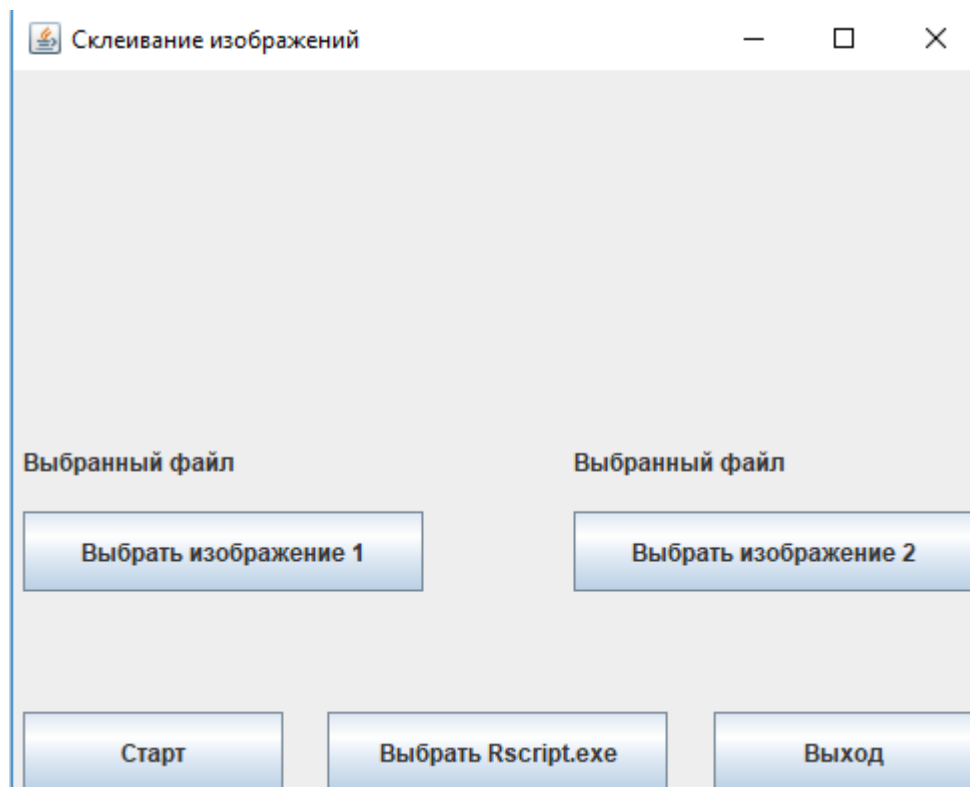


Рис. 5.8. Пользовательский интерфейс

Порядок работы системы склейки изображений:

- 1) Выбрать первое изображение из файла см. рис.5.9;
- 2) Выбрать второе изображение из файла;
- 3) Выбрать Rscript.exe см. рис.5.10;
- 4) Нажать «Старт».

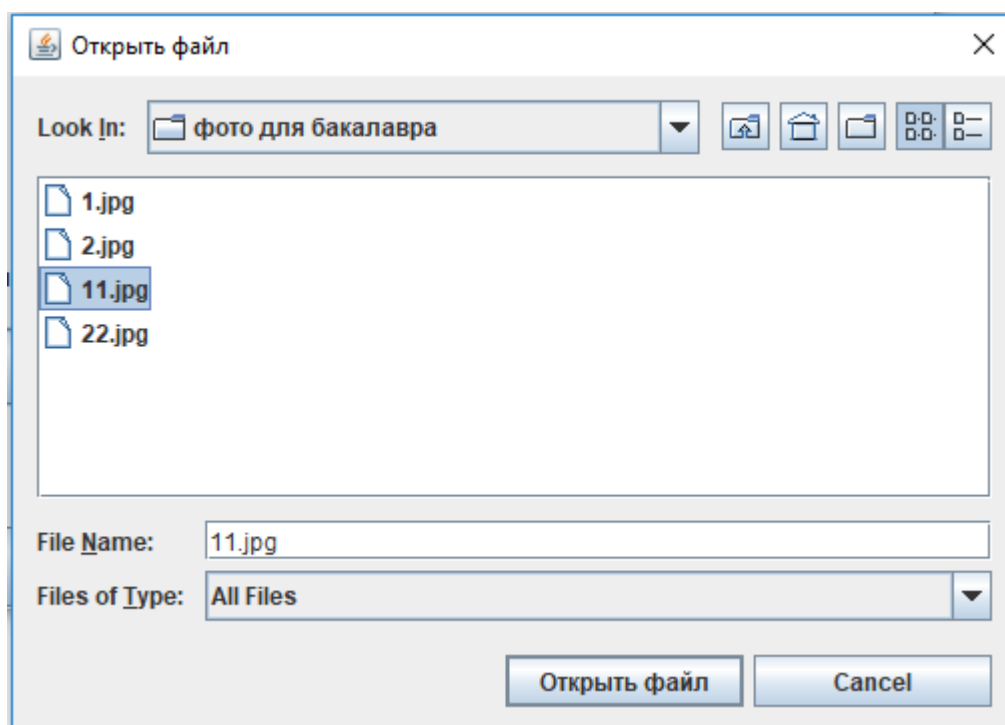


Рис. 5.9. Выбор изображений для склейки

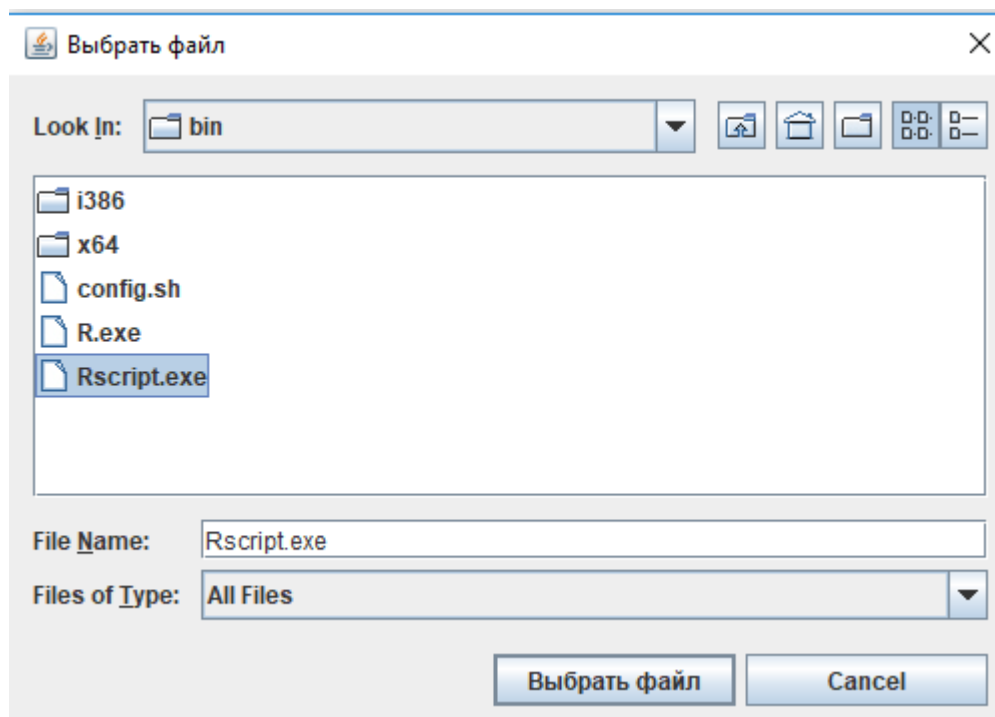


Рис. 5.10. Выбор файла Rscript.exe

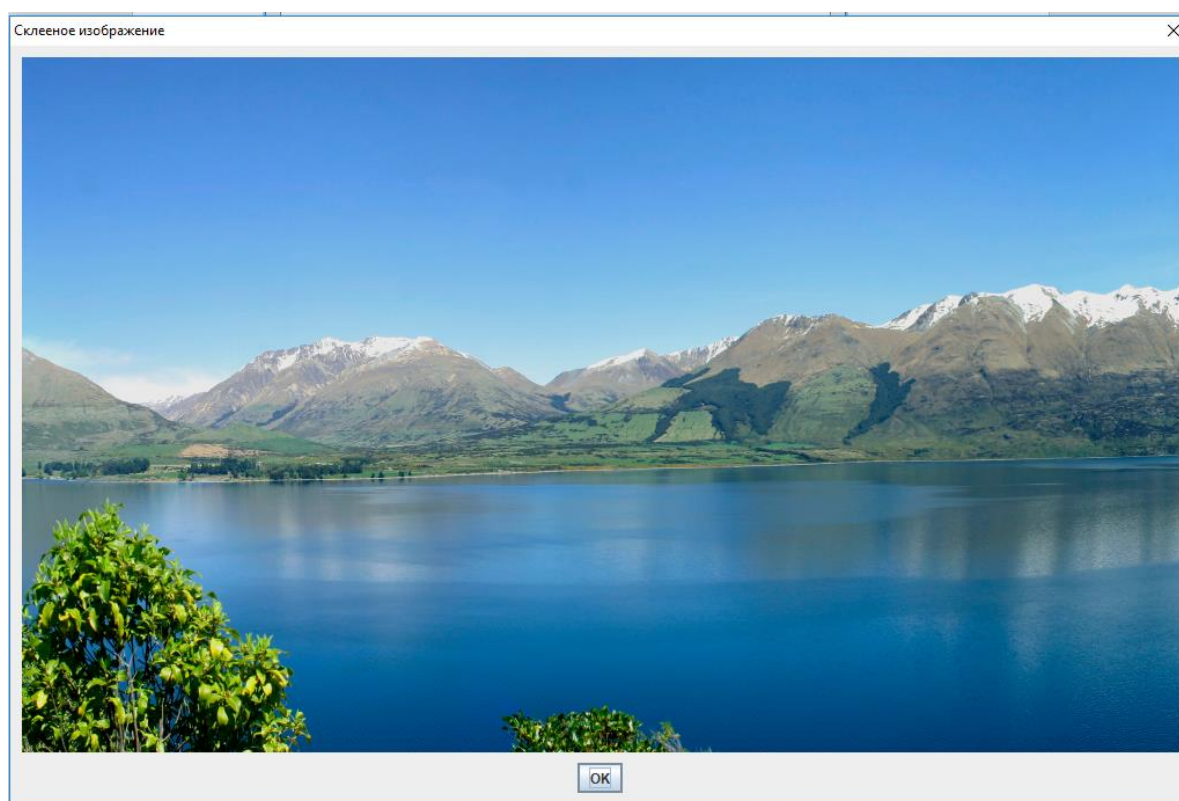


Рис. 5.11. Результат выполнения программы.

В результате мы наблюдаем бесшовно склеенное изображение. Далее нажимаем «ОК». Есть возможность выбрать для склейки другие изображения или нажать «Выход»

При повторном тестировании системы выбирать заново файл Rscript.exe не требуется.

5.3. Результаты вычислительного эксперимента

Рассмотрим результат работы программы с тестовыми парами изображений.

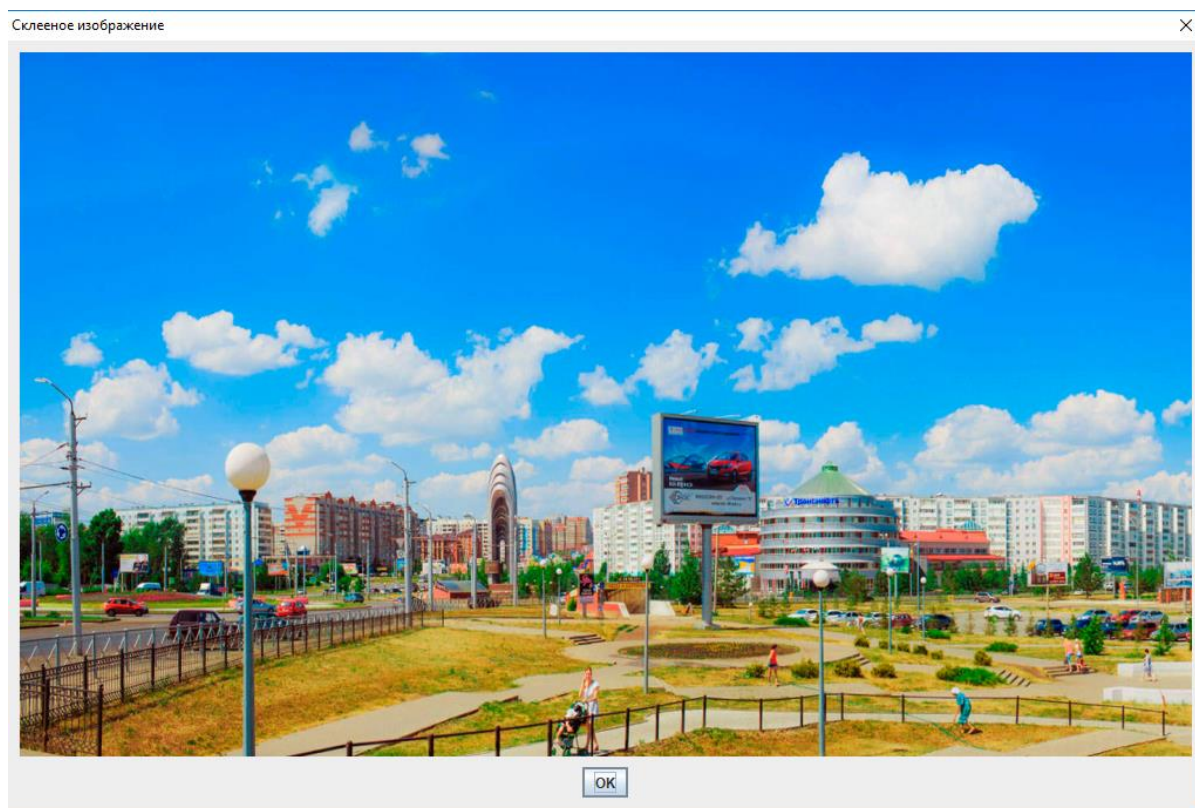


Рис. 5.12. Результат склейки тестовой пары 2.

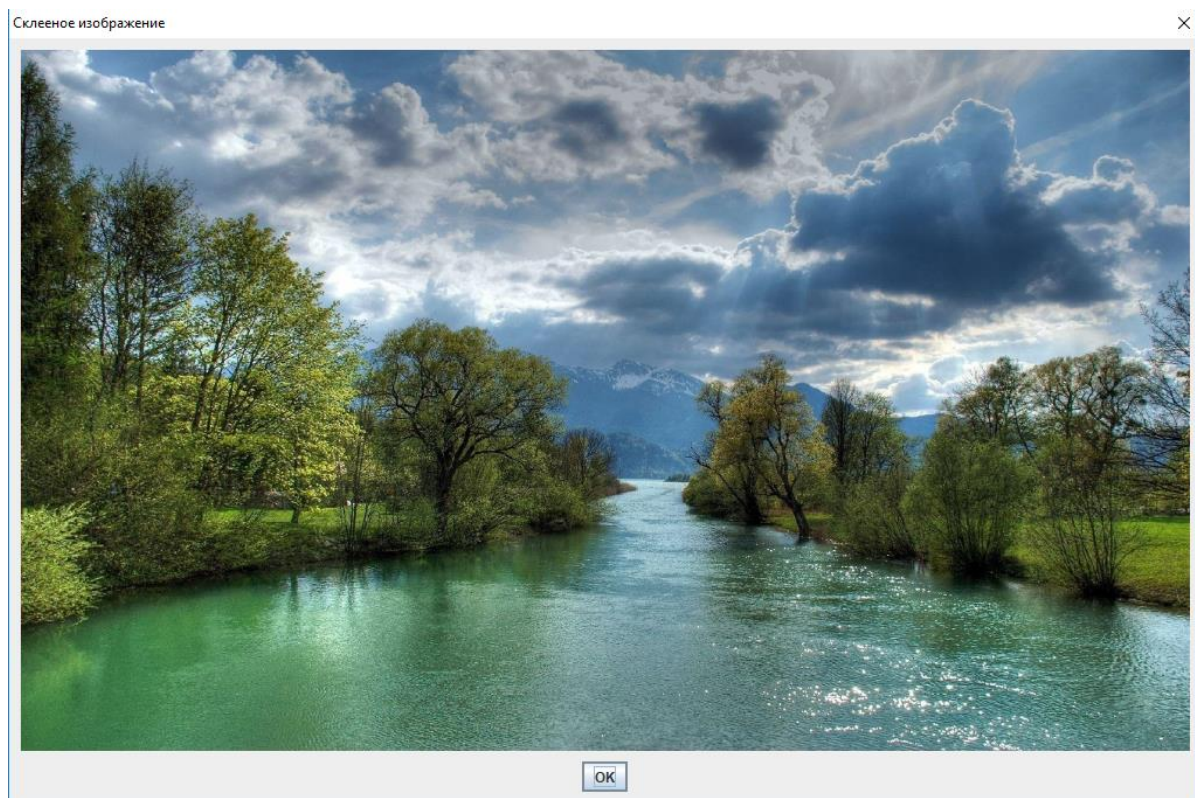


Рис. 5.13. Результат склейки тестовой пары 3.



Рис. 5.14. Результат склейки тестовой пары 4.



Рис. 5.15. Результат склейки тестовой пары 5.



Рис. 5.16. Результат склейки тестовой пары 6.

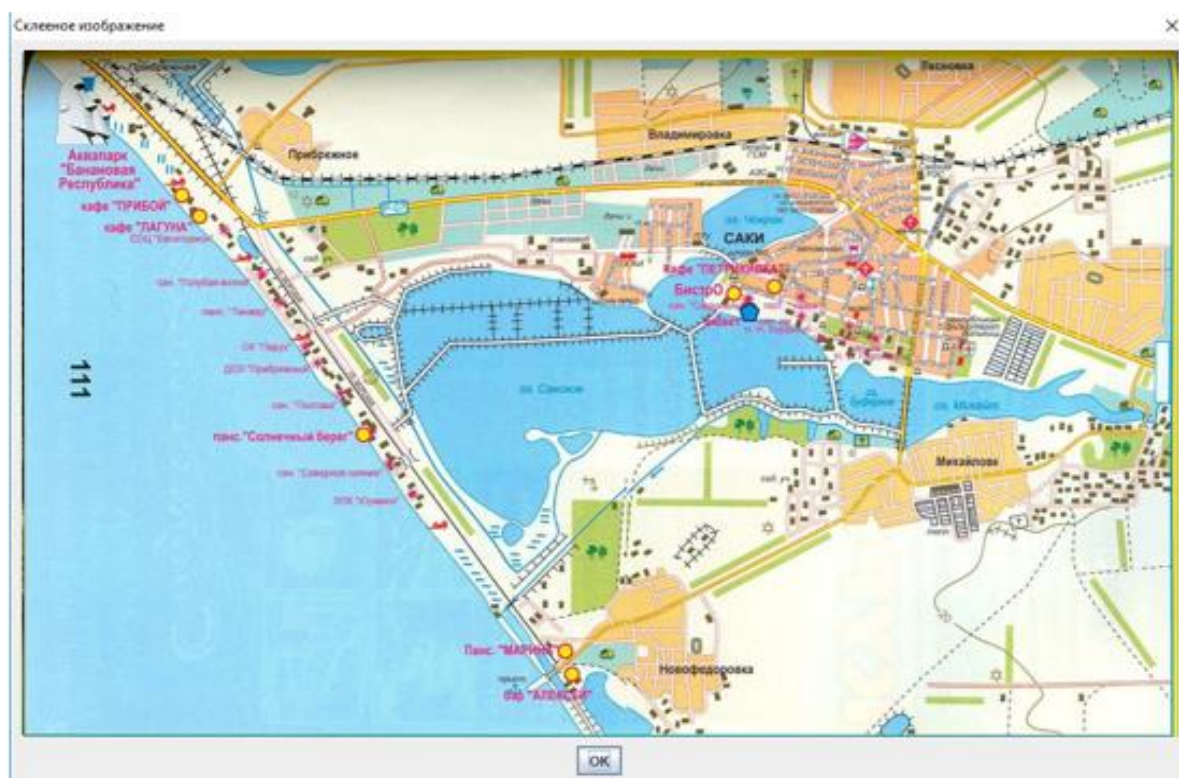


Рис. 5.17. – Результат склейки тестовой пары 7.

В результате искажение было замечено только на тестовой паре 6. Это произошло по причине высокой сложности рисунка и похожих значений среднеквадратических значений подобластей. В остальных случаях склейка прошла не заметно.

Изм.	Лист	№ докум.	Подпись	Дата

ВКР-НГТУ-09.03.01-(13-B-2)-009-17

Лист

33

Заключение

В результате выполнения выпускной квалификационной работы была спроектирована и программно реализована система бесшовной склейки изображений. Общий алгоритм работы программы был основан на теории активного восприятия.

Разработанная система предназначена для склеивания изображений одинакового размера, без поворотов друг относительно друга. Тестирование программы подтвердило ее работоспособность и возможность дальнейшего использования для решения данной задачи.

В дальнейшем планируется улучшить данную систему за счет увеличения количества склеиваемых изображений и добавления склейки изображений с поворотом.

					ВКР-НГТУ-09.03.01-(13-В-2)-009-17	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34

Список литературы

1. Создание бесшовного изображения – http://wiki.gislab.info/w/Создание_бесшовного_изображения
2. Windows – <https://ru.wikipedia.org/wiki/Windows>
3. MacOS – <https://ru.wikipedia.org/wiki/Macintosh>
4. Linux - <https://ru.wikipedia.org/wiki/Linux>
5. C++ – <https://ru.wikipedia.org/wiki/C%2B%2B>
6. JAVA – <https://ru.wikipedia.org/wiki/Java>
7. Python – <https://ru.wikipedia.org/wiki/Python>
8. R – [https://ru.wikipedia.org/wiki/R_\(язык_программирования\)](https://ru.wikipedia.org/wiki/R_(язык_программирования))
9. Библиотека ImageMagick – <https://cran.r-project.org/web/packages/magick/vignettes/intro.html>
10. Лучшие IDE для Java – https://geekbrains.ru/posts/java_ides
11. Ceres Solver – <https://habrahabr.ru/post/143205/>
12. Утробин, Владимир Александрович «Компьютерная обработка изображений. Основы специальной теории восприятия» - Нижний Новгород : Нижегородский гос. технический ун-т им. Р. Е. Алексеева, 2015
13. Java Swing – <https://javaswing.wordpress.com/>