

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Базовая кафедра геоинформационных систем

УТВЕРЖДАЮ
Заведующий кафедрой

_____ В.И. Харук

подпись

« ____ » _____ 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 - Информационные системы и технологии

Разработка модуля Quantum GIS для корреляционного анализа растровых изображений

Руководитель _____ доц. каф. Б-ГИС, к.т.н. А.С. Савельев
подпись, дата

Выпускник _____ А.С. Кудашев
подпись, дата

Нормоконтроль _____ Е.В. Федотова
подпись, дата

Красноярск 2016

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
Базовая кафедра геоинформационных систем

УТВЕРЖДАЮ

Заведующий кафедрой

_____ В.И. Харук

подпись

« ____ » _____ 2016 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту Кудашеву А.С.

Группа: КИ12-16Б Направление (специальность): 09.03.02

Информационные системы и технологии

Тема выпускной квалификационной работы: «Разработка модуля Quantum GIS для корреляционного анализа растровых изображений»

Утверждена приказом по университету _____

Руководитель ВКР: Савельев А.С., доцент кафедры Б-ГИС, канд. техн. наук

Исходные данные для ВКР: документация и руководство пользователя Python, Quantum GIS

Перечень разделов ВКР: обзор средств разработки; разработка анализа и программного обеспечения; руководство пользователя

Перечень графического материала: слайды презентации

Руководитель ВКР _____

А.С. Савельев

подпись

Задание принял к исполнению _____

А.С. Кудашев

подпись

« ____ » _____ 2016 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка модуля Quantum GIS для корреляционного анализа растровых изображений» содержит 51 страницу текстового документа, 14 использованных источников.

Объект изучения: язык программирования Python совместно с программой QGIS.

Цель работы: разработка модуля для QGIS для корреляционного анализа растровых изображений.

Задачи:

- разработать алгоритмы, для расчета локальных и глобальных коэффициентов корреляции, и с учетом зон для двух растровых изображений;
- реализация этих алгоритмов на языке Python, тестирование разработанных программ;
- тестирование разработанных программ на различных наборах растровых изображений;
- разработка пользовательского интерфейса, для программ расчета пространственной корреляции, при помощи QT Creator.

В результате были разработаны алгоритмы для расчета локальных и глобальных коэффициентов корреляции с учетом зон для двух растровых изображений. Так же алгоритмы были протестированы на различных растровых изображениях и разработан интерфейс программ расчета пространственной корреляции при помощи QT Creator.

СОДЕРЖАНИЕ

Введение.....	4
1 Обзор использованного программного обеспечения данных, языковых средств, библиотек	6
1.1 Язык программирования Python.....	6
1.1.1 Особенности языка Python.....	7
1.1.2 Архитектура Python	9
1.1.3 Среда исполнения Python.....	10
1.1.4 Типы данных Python	11
1.2 Геоинформационная система QGIS	12
1.3 Библиотека GDAL.....	14
1.4 Обзор библиотеки NumPy.....	15
1.5 Обзор библиотеки Math.....	16
1.6 Библиотека QT Creator	17
1.7 Выводы обзора использованного программного обеспечения данных, языковых средств, библиотек	17
2 Описание алгоритмов	19
2.1 Совмещение растров.....	19
2.2 Работа в программе Python	23
2.3 Глобальная, локальная корреляция и классификация зон.....	25
2.4 Выводы для описания алгоритмов	29
3 Описание элементов программирования.....	30
3.1 Использование библиотек для чтения и записи данных.....	30
3.2 Структура программы	31
3.3 Чтение и запись растра из формата GEOTIFF	33
3.4 Расчет глобальной, локальной корреляции и классификации зон	34
3.5 Пользовательский интерфейс	37
3.6 Выводы по описанию элементов программирования	39
4 Тестирование разработанных программ.....	40

4.1 Результаты расчета глобальной корреляции.....	40
4.2 Результаты расчета локальной корреляции	42
4.3 Классификации зон	45
4.4 Выводы по тестированию разработанных программ.....	47
Заключение	49
Список использованных источников	50

ВВЕДЕНИЕ

Для изучения взаимосвязи различных географических явлений, используют методы корреляционного и регрессионного анализа. Корреляционный анализ позволяет измерить тесноту связи, а регрессионный анализ дает возможность выразить взаимосвязь между двумя переменными в виде уравнений. Например, в случае линейной взаимосвязи при помощи методов наименьших квадратов подбираются параметры прямой, которая наилучшим образом опишет выборочные точки.

Для того чтобы изучать в геоинформационных системах взаимосвязь между явлениями, которые показаны в слоях ГИС, также можно использовать корреляционный и регрессионный анализ. Для представления пространственных данных в GIS существуют векторная и растровая модель данных. В данной работе акцент сделан на том, чтобы оценивать связь между растровыми данными, изображениями, данными дистанционного зондирования, дискретными растрами.

В Quantum GIS и ArcGIS есть геостатистические модули, которые позволяют проводить статистический анализ для пространственных данных в слоях. В Quantum GIS существуют богатые возможности для построения трендов и остаточных поверхностей. Но конкретно модулей для расчета корреляции двух изображений в Quantum GIS не обнаружено. Программа Quantum GIS содержит возможности для программирования, для улучшения функциональности, для тонкой настройки ГИС под нужным пользователем, используя различные языки программирования, можно настроить расширение ГИС.

В Quantum GIS - это Python, на котором можно построить разные расширения, а вообще, это стандартный язык для геообработки. Для того, чтобы писать программы - используется Python, чтобы программы работали быстрее - используется язык программирования C. Чтобы создать окна в

программе - используется среда разработки QT Creator, для статистических расчетов используются библиотеки NumPython, SciPy.

Цель работы: разработка модуля Quantum GIS для корреляционного анализа растровых изображений.

Задачи:

- разработать алгоритмы, для расчета локальных и глобальных коэффициентов корреляции, и с учетом зон для двух растровых изображений;
- реализация этих алгоритмов на языке Python, тестирование разработанных программ;
- тестирование разработанных программ на различных наборах растровых изображений;
- разработка пользовательского интерфейса, для программ расчета пространственной корреляции, при помощи QT Creator.

Для оценки взаимосвязи растровой модели, берется один растр, второй растр, совмещаются между собой, далее идет расчет глобальной корреляции, локальной корреляции и классификация зон.

1 Обзор использованного программного обеспечения данных, языковых средств, библиотек

1.1 Язык программирования Python

Для того, чтобы провести расчет пространственной корреляции, используется геоинформационная система Quantum GIS, потому что она бесплатная, распространяемая, удобная в использовании, позволяет решать сложные задачи, позволяет управлять геоданными, отображать, редактировать, анализировать их, также создавать макеты карт. Далее выполняем пространственную корреляцию в программе Python. Python очень прост в изучении, он позволяет решать сложные задачи, у него высокая удобочитаемость кода, высокоуровневый язык, масса библиотек. Для расчетов используются библиотеки NumPython и Scipy.Python, потому что они написаны на языке C, высокопроизводительные, удобные в использовании, позволяют быстро проводить любые расчеты.

Python язык программирования высокого уровня с динамической типизацией, автоматическим управлением памятью и удобными высокоуровневыми структурами данных, такими как словари (хеш-таблицы), списки, кортежи. Большая часть реализаций Python, — интерпретаторы. Поддерживает классы, модули (которые могут быть объединены в пакеты), обработку исключений, а также многопоточные вычисления. Питон обладает простым и выразительным синтаксисом. Язык поддерживает несколько парадигм программирования: структурное, объектно-ориентированное, функциональное и аспектно-ориентированное. Python существует для большинства активно используемых платформ. Он распространяется свободно под очень либеральной лицензией, позволяющей использовать его без ограничений в любых приложениях [1].

1.1.1 Особенности языка Python

Несомненно, достоинство интерпретатора Python, в том, что он реализован практически на всех платформах и операционных системах. Язык программирования Python очень прост в изучении, он позволяет решать сложные задачи, у него высокая удобочитаемость кода, высокоуровневый язык, масса библиотек. Большая часть программ на языке Python, выполняется без изменений на всех основных платформах. Перенос программного кода из Linux в Windows, заключается в простом копировании файлов программ с одной машины на другую. Также Python предоставляет массу возможностей по созданию переносимых графических интерфейсов.

В четкой, но достаточно простой структурируемости кода заключается причина популярности языка. Отсутствие громоздких конструкций, обозначающих новые классы, методы или циклы заменяются наглядным выделением пробелами и табуляцией. В таких условиях намного легче отслеживать прогресс создания программы, её легче отлаживать и дополнять. Структурированный код значительно проще понять как новичку в программировании, так и специалисту, сменившему язык программирования. При этом справедливо будет заметить, что без наличия хотя бы теоретических знаний об объектно-ориентированном программировании знакомство с Python и другими языками будет осложнено.

Но красивого оформления мало для достижения успеха в условиях жёсткой конкуренции между разработчиками разных языков программирования. Следующие факторы, так же являются залогом успеха Python на мировой арене.

Интерпретируемость языка пользователю доступен набор различных интерпретаторов. Это графические интерфейсы языка программирования, упрощающие работу с языком. Например, в стандартном дистрибутиве Python уже имеется интерпретатор, который в «горячем» режиме выполняет каждую

введённую программистом команду. Таким образом можно сильно сократить временные затраты на проект, когда для проверки определённого участка кода не нужно составлять полноценную программу, а новичок сможет сразу увидеть результат действия той или иной функции. Интерпретатор используется и как калькулятор.

Объектно-ориентированный подход. ООП - основная модель, на основе которой реализован Пайтон, но эта модель несколько отличается от традиционной: классы могут быть и являются объектами внутри программы; поддерживается множественное наследование; виртуальный полиморфизм классов; инкапсуляция всех уровней; наличие конструкторов, деструкторов и «уборщиков мусора» в базовой сборке; свободная перегрузка практически всех операторов, даже системных; развитые эмулятивные возможности; «вшитые» методы для работы с наиболее часто выполняемыми операциями; метапрограммирование; статические методы, поля и классы; вложенность.

Функциональность программирования. Возможна работа с функцией как объектом, полноценная рекурсивность, замыкания, обращение к частям функций и возможность создания собственных средств.

Пакетно-модульная система расширений языка. Python, благодаря своей простоте, имеет множество надстроек и дополнений, гибко увеличивающие возможности языка по мере необходимости. Подключение новых модулей выполняется в одну команду в коде или интерпретаторе.

Интроспекция эта функция языка позволяет по запросу получать детальную информацию о любом объекте внутри даже выполняющейся программы.

Эти и многие другие функции уже являются частью языка и там, где для многих «тяжеловесных» сред нужны дополнительные модули, Python справляется своим куда более компактным базовым дистрибутивом. Если же необходимой надстройки нет, её установка или самостоятельное написание не потребует много времени [2].

1.1.2 Архитектура Python

Универсальная архитектура языка Python позволяет создавать модели и исследовать концепции самого языка Python.

Любой язык состоит как минимум из двух частей – словаря и синтаксиса. Язык Python организован точно так же, предоставляя синтаксис для формирования выражений, образующих исполняемые программы, и словарь – набор функциональности в виде стандартной библиотеки и подключаемых модулей.

Как уже упоминалось, синтаксис Python достаточно лаконичный. С одной стороны – это хорошо, так как чем проще синтаксис, тем проще его изучить и тем меньше ошибок можно совершить в процессе его использования. Однако у подобных языков есть недостаток – с их помощью можно передавать самую простую информацию и нельзя выражать сложные конструкции.

К Python это не относится, так как это язык простой, но упрощенный. Дело в том, что Python является языком с более высоким уровнем абстракции, позволяет передать количество информации в меньшем объеме исходного кода.

Также Python является языком общего назначения, поэтому может применяться практически в любой области разработки ПО (standalone, клиент-сервер, Web-приложения) и в любой предметной области. Кроме того, Python легко интегрируется с уже существующими компонентами, что позволяет внедрять Python в уже написанные приложения.

Другая составляющая успеха Python – это его модули расширения, как стандартные, так и специфические. Стандартные модули расширения Python – это отлично спроектированная и неоднократно проверенная функциональность для решения задач, возникающих в каждом проекте по разработке ПО, обработки строк и текстов, взаимодействие с операционной системой, поддержка Web-приложений. Эти модули также написаны на языке Python, поэтому обладают его важнейшим свойством – кросс-платформенностью,

позволяющей безболезненно и быстро переносить проекты с одной операционной системы на другую.

Если необходимой функциональности не оказалось в стандартной библиотеке Python, то можно создать собственный модуль расширения для его последующего неоднократного использования. Здесь стоит отметить, что модули расширения для Python можно создавать не только на самом языке Python, но и с помощью других языков программирования. В этом случае появляется возможность более эффективной реализации ресурсоемких задач, например сложных научных вычислений, однако теряется преимущество кросс-платформенности, если язык модуля расширения не является сам по себе кросс-платформенным, как Python [2].

1.1.3 Среда исполнения Python

Благодаря средствам выполнения Python, можно создавать приложения с помощью языка программирования и выполнять их с помощью оптимизированного интерпретатора Python. Который, в себя включает разнообразные инструменты для разработки веб-приложений, в том числе и моделирования.

Все кросс-платформенные языки программирования построены по одной модели: это действительно переносимый исходный код и среды исполнения, которые не являются переносимыми и специфичными для каждой конкретной платформы. В эту среду исполнения обычно входит интерпретатор, который исполняет исходный код, и различные утилиты, необходимые для сопровождения приложения.

В среду исполнения Python входит только интерпретатор, который одновременно является и компилятором, однако компилирует исходный код Python непосредственно в машинный код целевой платформы [3].

1.1.4 Типы данных Python

Базовые типы и структуры данных в языке Python, обладают большой гибкостью и более широкими возможностями, чем типы данных, доступные в низкоуровневых языках программирования. Python представляет наиболее типичные структуры данных, такие как: список, кортеж, словарь. Встроенные объекты могут расширяться и сжиматься по мере необходимости, могут комбинироваться друг с другом для представления данных со сложной структурой.

Типы данных, используемых в Python, также совпадают с другими языками – целые и вещественные типы данных; дополнительно поддерживается комплексный тип данных – с вещественной и мнимой частью (пример такого числа – $1.5J$ или $2j$, где J представляет собой квадратный корень из -1). Python поддерживает строки, которые могут быть заключены в одинарные, двойные или тройные кавычки, при этом строки, являются `immutable`-объектами, не могут изменять свое значение после создания.

Есть в Python и логический тип данных `bool` с двумя вариантами значения `True` и `False`. Однако в старых версиях Python такого типа данных не было, и, кроме того, любой тип данных мог быть приведен к логическому значению `True` или `False`. Все числа, отличные от нуля, и непустые строки или коллекции с данными трактовались как `True`, а пустые и нулевые значения рассматривались как `False`. Эта возможность сохранилась и в новых версиях Python, однако для повышения читаемости кода рекомендуется использовать для логических переменных тип `bool`. В то же время, если необходимо поддерживать обратную совместимость со старыми реализациями Python, то в качестве логических переменных стоит использовать `1` (`True`) или `0` (`False`).

В Python определены три типа коллекций для хранения наборов данных: кортеж (`tuple`), список (`list`), словарь (`dictionary`).

Кортеж представляет собой неизменяемую упорядоченную последовательность данных. В нем могут содержаться элементы различных типов, например другие кортежи. Кортеж определяется в круглых скобках, а его элементы разделяются запятыми. Специальная встроенная функция tuple() позволяет создавать кортежи из представленной последовательности данных.

Список – это изменяемая упорядоченная последовательность элементов. Элементы списка также разделяются запятыми, но задаются уже в квадратных скобках. Для создания списков предлагается функция list().

Словарь является хеш-таблицей, сохраняющей элемент вместе с его идентификатором-ключом. Последующий доступ к элементам выполняется тоже по ключу, поэтому единица хранения в словаре – это пара объект-ключ и связанный с ним объект-значение. Словарь – это изменяемая, но не упорядоченная коллекция, так что порядок элементов в словаре может меняться со временем. Задается словарь в фигурных скобках, ключ отделяется от значения двоеточием, а сами пары ключ/значение разделяются запятыми. Для создания словарей доступна функция dict [3].

1.2 Геоинформационная система QGIS

Для того, чтобы провести расчет пространственной корреляции, используется геоинформационная система Quantum GIS, потому что данная программа бесплатная, распространяемая, удобная в использовании. Также Quantum GIS позволяет решать сложные задачи и управляет геоданными. В программе нужно совместить два растровых изображения, через функцию совмещение растров.

Функция совмещение растров делается, чтобы получились два растра, которые имеют одинаковую ширину, высоту, размер, размер ячейки, одинаковую начальную точку левого нижнего угла. Чтобы каждая клеточка

растра в пространстве совпадала с клеточкой с такими же координатами, что и на другом растре.

Программа QGIS – свободная кроссплатформенная геоинформационная система, позволяющая управлять геоданными, отображать, редактировать и анализировать их, а также создавать макеты карт. Благодаря интеграции с GRASS, эта ГИС обладает мощным аналитическим функционалом (рисунок 1, блок - схема 1).

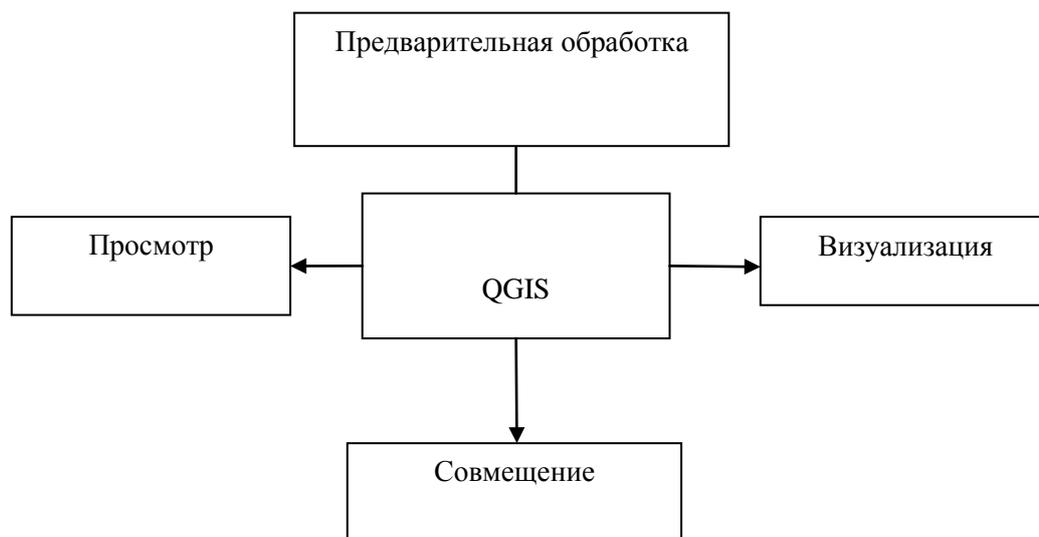


Рисунок 1 – Блок-схема 1. Функции QGIS

В QGIS можно просматривать и накладывать друг на друга векторные и растровые данные в различных форматах и проекциях без преобразования во внутренний или общий формат. Поддерживаются следующие основные форматы:

- пространственные таблицы с использованием PostGIS, векторные форматы, включая shape-файлы ESRI, MapInfo;
- форматы растров и графики, поддерживаемые библиотекой GDAL такие, как GeoTIFF, Erdas IMG, ArcInfo ASCII Grid, JPEG, PNG;

- базы данных SpatiaLite ;
- растровый и векторный форматы GRASS (область, набор данных).

Геоинформационная система QGIS может быть адаптирован к особым потребностям с помощью расширяемой архитектуры модулей. QGIS предоставляет библиотеки, которые могут использоваться для создания модулей. Можно создавать отдельные приложения, используя язык программирования Python.

С помощью удобного графического интерфейса можно создавать карты и исследовать пространственные данные. Графический интерфейс включает в себя множество полезных инструментов, например:

- перепроецирование «на лету»;
- компоновщик карт;
- панель обзора;
- пространственные закладки;
- определение/выборка объектов;
- редактирование/просмотр/поиск атрибутов;
- подписывание объектов;
- изменение символики векторных и растровых слоев;
- сохранение и загрузка проектов [4].

1.3 Библиотека GDAL

Утилиты командной строки, входящие в состав библиотеки GDAL, широко используются для выполнения разнообразных задач. GDAL - это свободная библиотека для работы с растровыми данными.

Для того, чтобы выполнить перепроецировку, совмещение растров, обрезку растров, произвести геотрансформацию, получить информацию о количестве каналов растра, извлечь данные и метаданные - устанавливается

библиотека GDAL. Также модуль GDAL выполняет чтение из разных форматов например: GeoTiff, jpeg. Для работы с пространственными данными, корреляционным анализом и дистанционным зондированием, главным форматом является GeoTiff. Потому что в формате GeoTiff присутствует геопривязка и трансформация проекции. Также GeoTiff может открывать изображение и накладывать на него другое.

Библиотека GDAL - очень полезный, бесплатно распространяемый набор инструментов и библиотек, с открытым исходным кодом:

- `from osgeo import gdal;`
- `from osgeo import ogr;`
- `import numpy as np.`

Использование этих инструментов - это один из немногих способов осуществить географическую привязку растрового изображения используя исключительно бесплатное (и легальное) ПО.

Первоначально GDAL разрабатывался Фрэнком Вармердамом вплоть до выпуска версии 1.3.2, после которого поддержкой проекта стал заниматься комитет GDAL/OGR Project Management Committee под контролем OSGeo

Если используется GDAL версии 1.5 и выше, рекомендуется использовать «основные» модули. А для случаев, когда необходимо использовать код написанный ранее, можно сделать проверку. В большинстве случаев достаточно подключить только модуль GDAL [5].

1.4 Обзор библиотеки NumPy

Для расчета коэффициента глобальной корреляции, локальной корреляции и классификации зон, используется библиотека NumPython. Библиотека NumPython превращает Python в свободный и более мощный эквивалент системы. Библиотека выполняет реализацию матричных и

векторных операций, численных алгоритмов. Чтобы использовать модуль NumPython его нужно импортировать командой `import numpy as np`.

Библиотека NumPy обеспечивает работу с известными математическими функциями `sin`, `cos`, `exp`. Но в NumPy эти функции называются универсальными (`ufunc`).

Библиотека NumPy — это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами.

Создание массивов, атрибутов, базовых операций, поэлементном применении функций, индексах, срезах, итерировании. Рассматриваются различные манипуляции с преобразованием формы массива, объединение массивов из нескольких и наоборот, разбиение одного на несколько более мелких.

Основным объектом NumPy является однородный многомерный массив. Это таблица элементов (обычно чисел), всех одного типа, индексированных последовательностями натуральных чисел. Под «многомерностью» массива мы понимаем то, что у него может быть несколько измерений или осей. Поскольку слово «измерение» является неоднозначным, вместо него мы чаще будем использовать слова «ось» (`axis`) и «оси» (`axes`). Число осей называется рангом (`rank`). Например, координаты точки в трехмерном пространстве $(1, 2, 1)$ это массив ранга 1 у него есть только одна ось. Длина этой оси 3. Другой пример, массив $(1, 0, 0)$, $(0, 1, 2)$ представляет массив ранга 2 (то есть это двухмерный массив) [6].

1.5 Обзор библиотеки Math

В глобальной корреляции, локальной корреляции и классификации зон, функции `Corrcoef`, `ravel`, используются из модуля `Math`. Функция `Corrcoef`, возвращает коэффициенты корреляции, а функция `ravel`, из матрицы делает

развертку вектора, который можно разложить в два вектора ряда. Модуль `math` содержит более сложные математические функции. Чтобы использовать модуль, его нужно импортировать командой `import math`.

Модуль `math` – один из наиважнейших в Python. Этот модуль предоставляет обширный функционал для работы с числами [7].

1.6 Библиотека QT Creator

Для программы коэффициента глобальной корреляции, локальной корреляции и классификации зон, через программу QT Creator, был нарисован интерфейс программы - это окна, кнопки и другие нужные компоненты.

Основная задача Qt Creator — упростить разработку приложения с помощью фреймворка `Qt` на разных платформах

Программа QT Creator кроссплатформенная свободная IDE для разработки на C, C++ , python. Разработана Trolltech (Digia) для работы с Фреймворком QT. Включает в себя графический интерфейс отладчика и визуальные средства разработки интерфейса как с использованием QtWidgets, так и QML.

Поэтому среди возможностей, присущих любой среде разработки, есть и специфичные, такие как отладка приложений на QML и отображение в отладчике данных из контейнеров Qt, встроенный дизайнер интерфейсов как на QML, так и на QtWidgets [8]

1.7 Выводы обзора использованного программного обеспечения данных, языковых средств, библиотек

Программа Quantum GIS используется для:

- совмещения растров;
- загрузки и обработки изображений;

- выбора метода;
- обрезки растров;
- выбора системы координат;
- выбора размера ячейки;
- выбора офсет сетки;
- выбора обрезки по обхвату растра и установки разрешения.

Программа Quantum GIS выбрана, потому что она бесплатная, распространяемая, удобная в использовании, она позволяет решать сложные задачи, позволяет управлять геоданными, отображать, редактировать, анализировать их, а также создавать макеты карт.

Программа Python используется для реализации:

- глобальной корреляция;
- локальной корреляция;
- классификации зон.

Python выбран, потому что, он реализован практически на всех платформах и операционных системах, очень прост в изучении, позволяет решать сложные задачи, у него высокая удобочитаемость кода, высокоуровневый язык, масса библиотек.

Программа QT Creator используется для:

- создания окон;
- создания кнопок;
- создания интерфейса.

QT Creator выбран, потому что, он реализован практически на всех платформах и операционных системах, очень прост в изучении.

Модуль GDAL используется для:

- перепроецирования;
- совмещения растров;
- обрезки растров;

- геотрансформации;
- получении информации о количестве каналов раstra;
- извлечения данных.

Библиотека GDAL выбрана, потому что, это свободная библиотека для работы с растровыми данными, она использует географическую привязку растрового изображения и читает форматы GeoTIFF.

Модуль NumPy используется для подсчета:

- глобальной корреляция;
- локальной корреляция;
- классификации зон.

Модуль NumPy выбран, потому что, он выполняет реализацию матричных и векторных операций, численных алгоритмов. NumPy обеспечивает работу с известными математическими функциями, а также этот модуль работает с массивами и матрицами.

В совокупности, все выше перечисленные программы, библиотеки и данные, помогут создать модуль для Quantum GIS и выполнить подсчет глобальной, локальной корреляции и классификации зон.

2 Описание алгоритмов

2.1 Совмещение растров

Для визуализации данных, интерфейса, работы с изображениями, анализа данных, проектов - используется программа Quantum GIS. Так как данная программа бесплатная, распространяемая, удобная в использовании, управляет геоданными, отображает, редактирует, анализирует данные, а также создает макеты карт. Самая главная функция, которая есть в Quantum GIS – это совмещение растров.

Для дальнейшей работы нужно совместить растровые изображения. Чтобы начать расчет корреляций NumPy corrcoef необходимо взять растры. Эти растры будут разной ширины, высоты, размера, размера ячейки, каждая клеточка растра в пространстве не будет совпадать с клеточкой с такими же координатами, что и на другом растре. Для того, чтобы совместить растры, используется программа Quantum GIS, функция совмещение растров. Это позволит совместить их и сделать одинаковой ширины, высоты, совместить сетки в пространстве, сделать одного разрешения. Совмещение растров нужно для подсчета глобальной, локальной корреляции и классификации зон.

Для начала выполнения работы берется за разные годы, но с одинаковой местностью, растры, из Landsat 8. Допустим, Landsat 8 2014 года и Landsat 8 2015 года, в формате GEOTIFF. Из Landsat выбираем любой из каналов. Например, возьмем первый и второй канал в пределах одного изображения. Далее через программу Quantum GIS нужно провести операцию совмещение растров из этих двух каналов. В программе Quantum GIS, в верхней строке, нажать на окошко и выбрать иконку под названием растр, далее в этом окне ниже выбрать операцию совместить растры (рисунок 2).

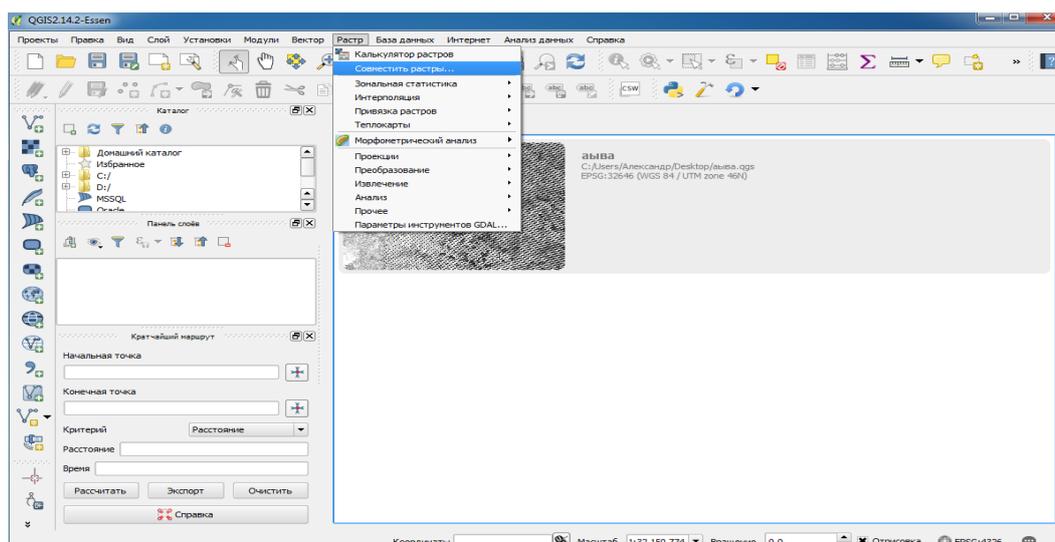


Рисунок 2 – Работа с растрами

В открывшемся окне добавляется первый и второй растр с разных каналов, затем нужно выбрать, куда сохранить готовые, обработанные два растра. Ниже выбирается функция, resampling method, где находятся 7 методов, которыми можно воспользоваться (рисунок 3):

- ближайший сосед;
- билинейная (2 на 2 кернел);
- кубическая(4 на 4)
- кубический В-сплайн(4 на 4 кернел);
- ланчос(6 на 6 кернел);
- среднее;
- мода.

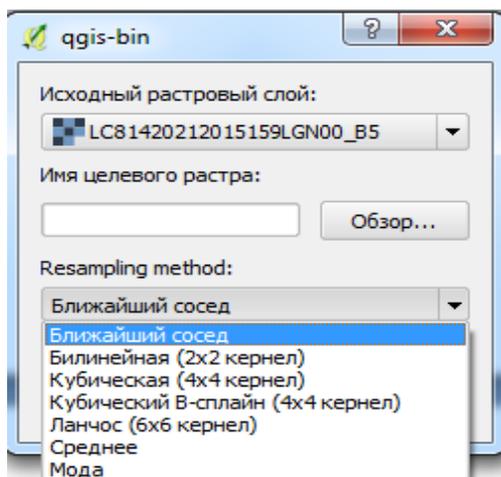


Рисунок 3 – Выбор метода

Ближайший сосед - это метод, который вычисляет индексы ближайшего соседства на основе среднего расстояния, от каждого объекта до ближайшего к нему соседнего объекта [9].

Метод билинейной интерполяции может относиться к фильтрации или билинейной интерполяции. Данный метод выполняется путем интерполяции между четырьмя пикселями, ближайшим к точке, которая

наилучшим образом представляет этот пиксель (как правило, в середине или в верхней левой части пикселя) [10].

Метод аппроксимации кубической полиномы, используется для определения уровней серого в изображении, через взвешенное среднее из шестнадцати ближайших пикселей к входным координатам. Затем полученное значение присваивается к выходным координатам. Этот метод, как правило, считается лучше, чем билинейная интерполяция, и он не имеет бессвязного появления ближайшего соседа интерполяции [11].

Ланчос - это метод, который является методом интерполяции, используется для вычисления новых значений и для выборочных данных. Метод часто используется в многомерной интерполяции, например, для масштабирования изображения (для изменения размеров цифровых изображений). Также может использоваться для любых других данных.

Среднее - это метод для работы с растровыми географическими данными. Для растрового слоя или любой зоны слоя, могут быть вычислены статистические характеристики, включающее среднее значение и медиану. Наиболее часто встречающиеся значения - это дисперсия и множество других статистических характеристик. Для нескольких слоев может быть проведено их статистическое сравнение, например, при помощи регрессионного или дисперсионного анализа [12].

Из всех данных методов был выбран один, это метод аппроксимации кубической полиномы, так как он самый удобный и быстрый.

После того как был выбран метод, нужно задать растрам систему координат, в данной работе была выбрана система координат WGS-84. Затем ниже в окне задать размер ячейки, офсет сетки и, если необходимо, можно обрезать по обхвату растр и установить общее разрешение. Если необходимо, по обхвату нужно обрезать растры, для того, чтобы быстрее загружались изображения в Quantum GIS и Python. После того, как все параметры заданы,

нажимаем ОК и программа проделывает все выше описанные операции с растрами (рисунок 4).

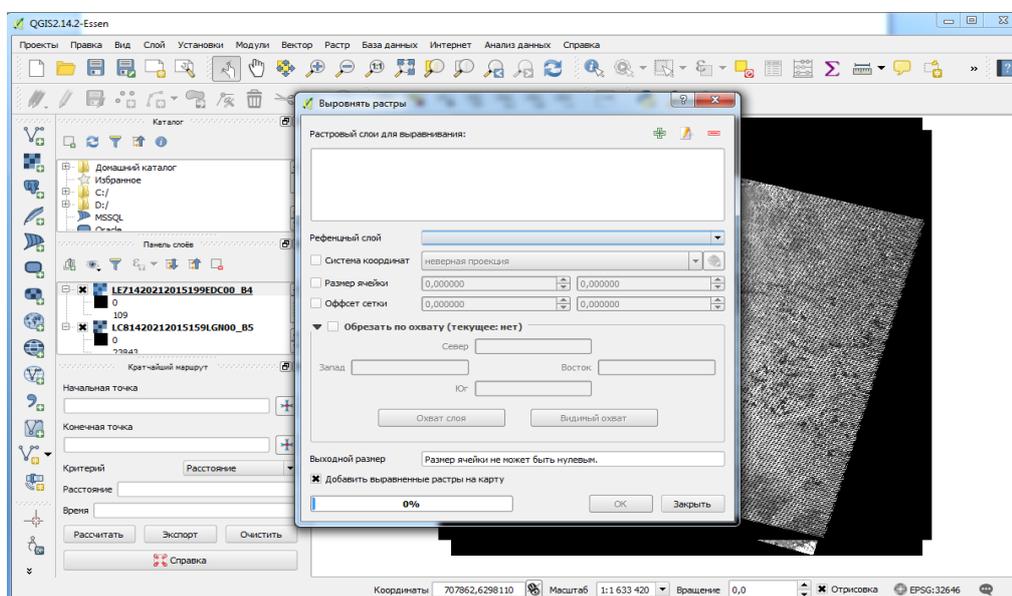


Рисунок 4 – Совмещение растров

В итоге получились два растра, которые имеют одинаковую ширину, высоту, размер, размер ячейки, одинаковую начальную точку левого нижнего угла. Каждая клеточка растра в пространстве совпадает с клеточкой с такими же координатами, что и на другом растре. Это необходимо для подсчета пространственной корреляции, чтобы расчеты были точны.

2.2 Работа в программе Python

Для выполнения коэффициентов корреляции используется интерпретатор Python. Потому, что он очень прост в изучении, позволяет решать сложные задачи, у него высокая удобочитаемость кода, высокоуровневый язык с динамической типизацией, он автоматически управляет памятью и структурными данными, содержит массу библиотек.

После того, как обработаны два растровых изображения, начинается работа в программе Python. Для глобальной, локальной корреляции и

классификации зон, необходимо установить дополнительные библиотеки, такие как Gdal, NumPy, Math. Библиотека GDAL поможет произвести перепроецировку, совмещение, обрезку растров, выполнить геотрансформацию, чтение из GEOTIFF. Библиотека NumPy превращает Python в свободный и более мощный эквивалент системы, также выполняет реализацию матричных и векторных операций, численных алгоритмов. Библиотека Math работает с множеством математических функций, этот модуль предоставляет обширный функционал для работы с числами. (рисунок 5 – блок-схема 2).

Программа Python и выше перечисленные модули, помогут выполнить расчет глобальной, локальной корреляции и классификации зон.

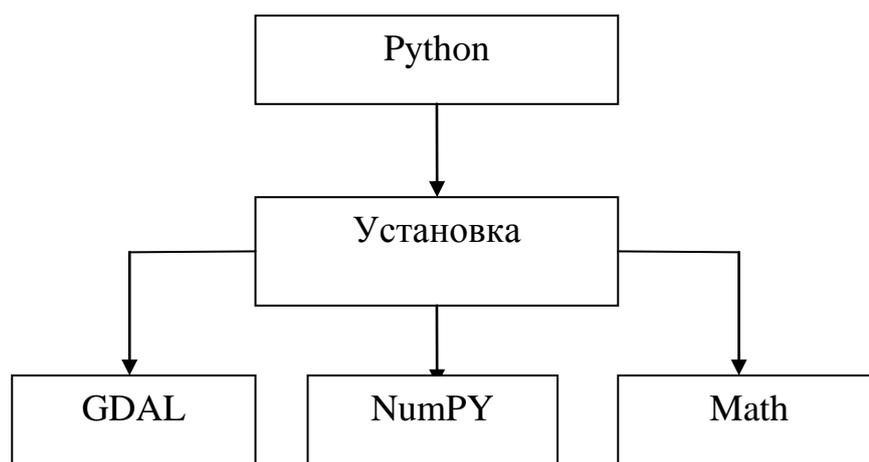


Рисунок 5 – Блок-схема 2. Установка в Python библиотек Gdal, NumPy, Math.

После установки всех компонентов можно приступить к написанию программы. Загружаем в программу первый растр, второй растр и считаем сначала коэффициент локальной корреляции, глобальной корреляции и классификацию зон. Итогом на выходе получится готовый коэффициент корреляции, в виде третьего растра.

2.3 Глобальная, локальная корреляция и классификация зон

Было выбрано три функции корреляции:

- глобальная корреляция;
- локальная корреляция;
- классификация Zone (рисунок 10 – блок-схема 2).

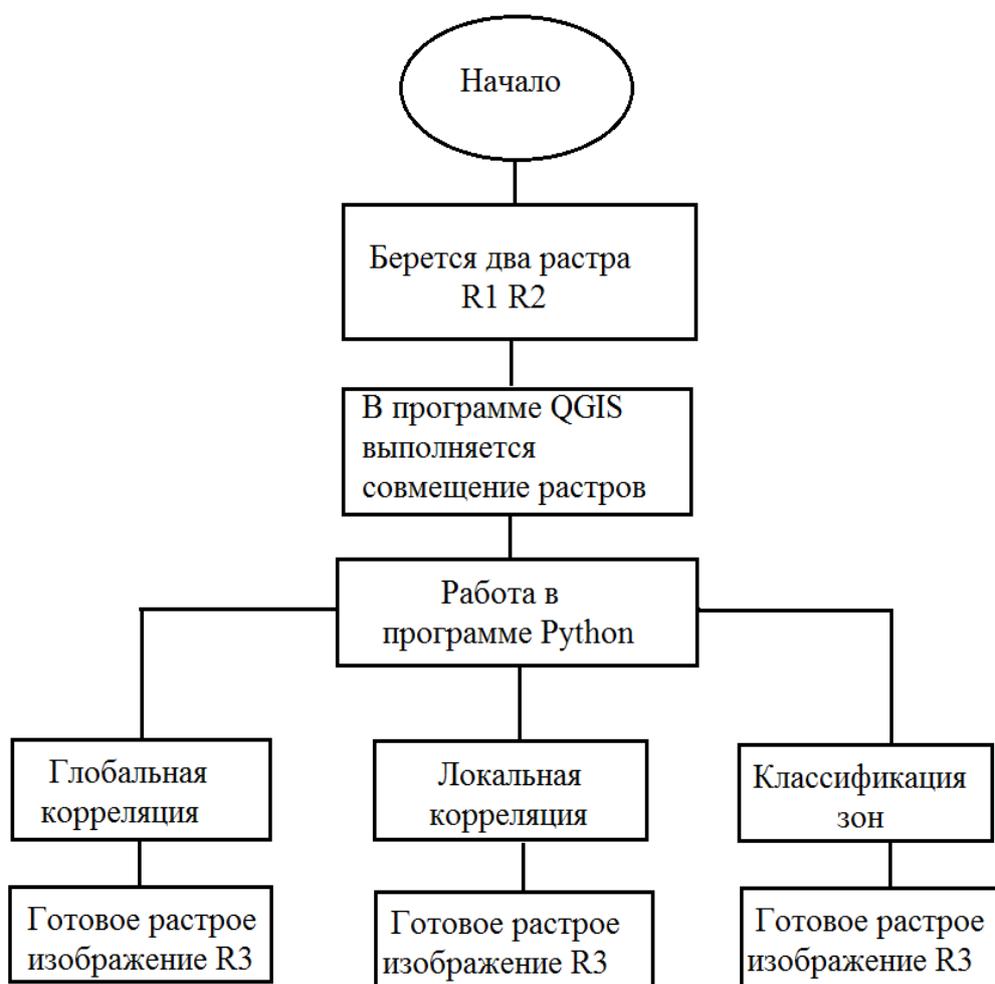


Рисунок 6 – Блок-схема 3. Нахождение локальной, глобальной корреляции
классификация зон

Первая корреляция, которая была рассмотрена – это глобальная корреляция: корреляция считается по всему растру. Растр раскладывается в два длинных ряда и считается коэффициент корреляции. Эту функцию выполняет библиотека Python NumPy.corrcoef. После этого, растры загружаются в массивы и можно обращаться к массиву, где будут все значения чисел в этом растре. С помощью функции ravel, двумерный массив, можно развернуть в два длинных вектора строки, получится большое количество членов этого ряда. После развертки, помещаются данные в формулу Пирсона и считается коэффициент корреляции. (рисунок 7).

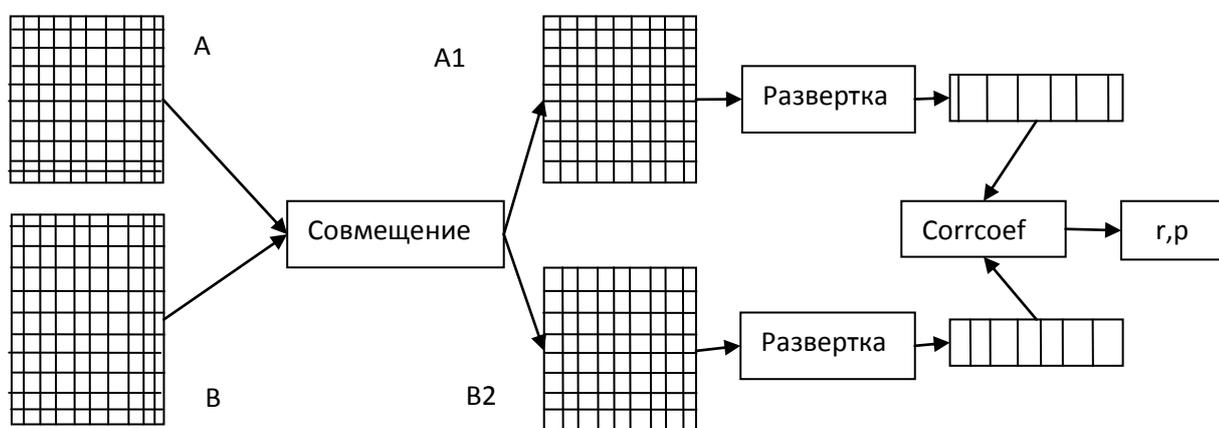


Рисунок 7 – Глобальная корреляция

Вторая корреляция, которая была рассмотрена это - локальная корреляция: проверяется коэффициент корреляции не полностью, а в определенных местах. Берется первый растр в виде матрицы, также второе изображение в виде матрицы, проверяется матрица одного размера или нет,

далее берется скользящее окно пять на пять и сдвигается по каждому пикселю. Передается массив NumPy первый и второй, проверяется его размерность, количество строк. Создается пустое изображение из нулей, и выполняется обход раstra по строкам и столбцам, выделяется подматрица. Библиотека `ravel` из матрицы делает развертку вектора, которая проходит по изображению, также как в глобальной корреляции, разворачивается в две длинные строки вектора и с каждым сдвигом, по каждой клетке, считается корреляция (рисунок 8).

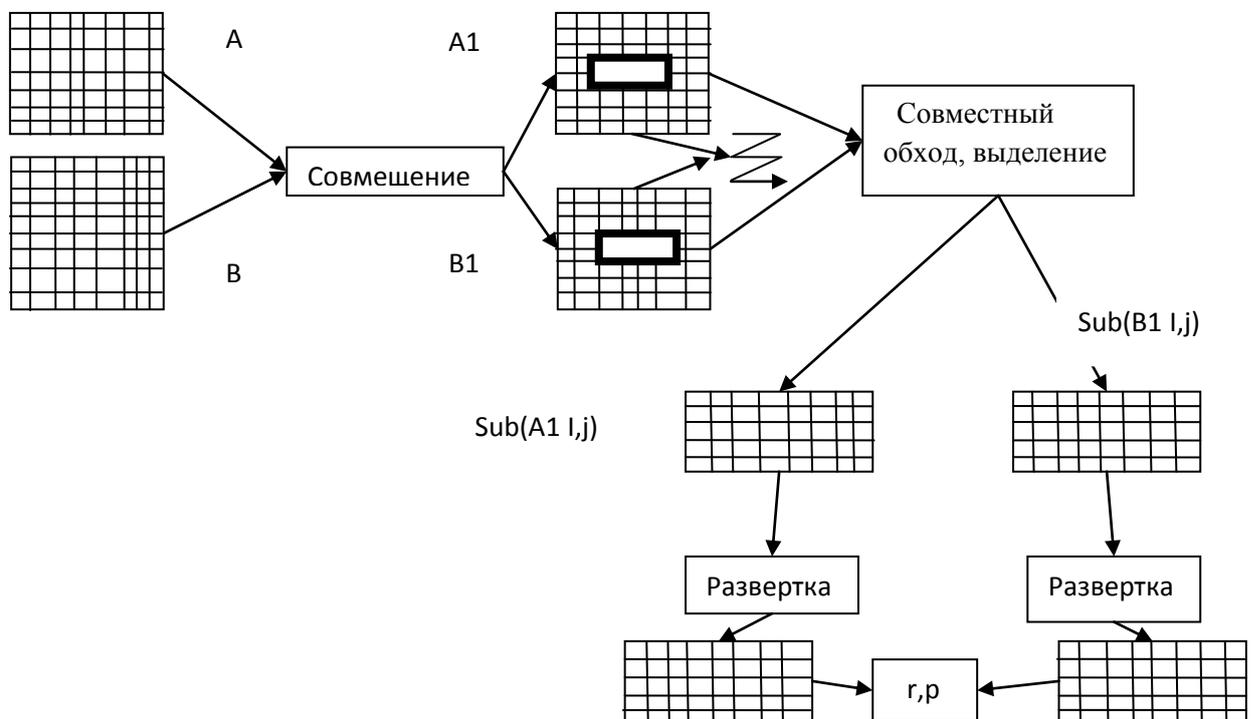


Рисунок 8 – Локальная корреляция

Третьим была рассмотрена классификация *Zone*: вычисление по классам, берется первый и второй растр, также выбирается третий растр в котором есть классификация, в третьем растре выделяется зона и в этой зоне считается коэффициент корреляции, например классификация растительности и для нее выделяется на этом растре нужные пиксели. (рисунок 11).

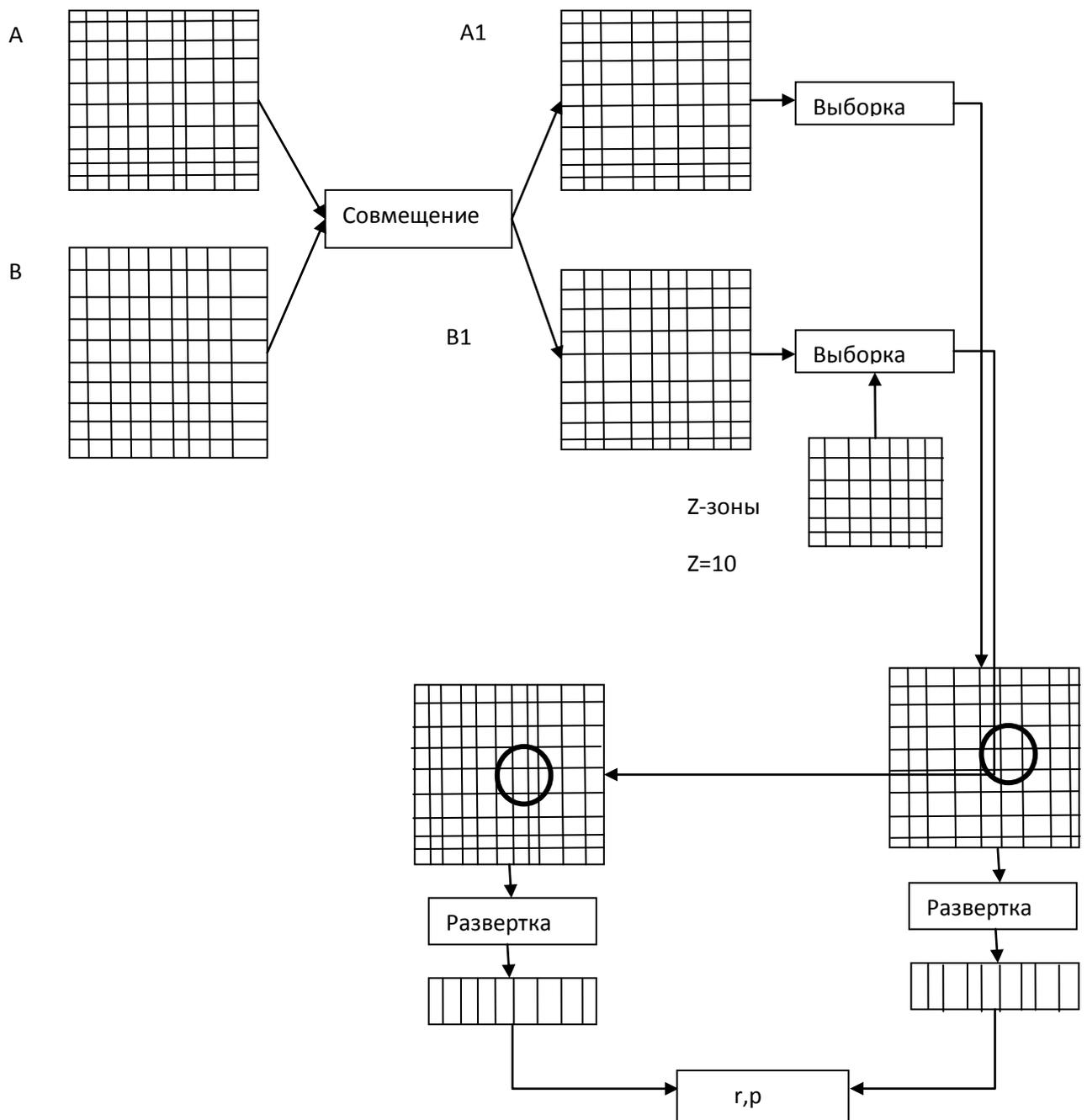


Рисунок 9 – Классификация зон

Было рассмотрено и описано три вида корреляции: глобальная корреляция, локальная корреляция и классификация зон.

2.4 Выводы для описания алгоритмов

В разделе описания алгоритмов рассказывается, как в программе Quantum GIS было выполнено совмещение растров, был выбран метод аппроксимации кубической полиномы, была задана растром система координат WGS-84. Также как делается размер ячейки, офсет сетки и установка разрешения для двух растров.

После обработки двух растров в Quantum GIS далее шла работа в программе Python. Чтобы рассчитать коэффициент глобальной, локальной корреляции и классификации зон, необходимо установить дополнительные библиотеки, такие как GDAL, NumPy, Math:

- GDAL производит перепроецировку, совмещение, обрезку растров, выполняет геотрансформацию, чтение из GeoTif;
- NumPy превращает Python в свободный и более мощный эквивалент системы, также выполняет реализацию матричных и векторных операций, численных алгоритмов;
- Math работает с множеством математических функций, этот модуль предоставляет обширный функционал для работы с числами.

Также были описаны алгоритмы выполнения коэффициентов глобальной, локальной корреляции и классификации зон. С помощью программ Quantum GIS, Python и установленных библиотек к интерпретатору Python, можно посчитать коэффициенты пространственной корреляции.

3 Описание элементов программирования

3.1 Использование библиотек для чтения и записи данных

В программе Python для начала работы загружаются готовые библиотеки: `gdal`, `math`, `numpy`, `numpy as np`, `from scipy.stats import pearsonr`, `from aom_zones import`, `from aom_correl import inZoneCorrelationMap`. Эти библиотеки помогут для работы с растровыми данными, для работы с различными числами, многомерными массивами, матрицами. Также посчитают локальную корреляцию, глобальную корреляцию и помогут разбить растр по зонам и посчитать его классификацию.

Для начала работы нужно импортировать три важных библиотеки, которые помогут произвести расчет глобальной, локальной корреляции и классификации зон:

- `import gdal;`
- `import math;`
- `import numpy.`

Библиотека GDAL - это свободная библиотека для работы с растровыми данными. Утилиты командной строки, входящие в состав библиотеки, широко используются для выполнения разнообразных задач. Таких как перепроецировка, совмещение, обрезка растров, выполняет геотрансформацию, может получать информацию о количестве каналов растра, извлекать данные и метаданные. Читает форматы GEOTiff.

Библиотека Math одна из наиважнейших в Python. Этот модуль предоставляет обширный функционал для работы с числами и с множеством математических функций. Такие как `ravel`, `corrcoef`.

Библиотека NumPy - это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой

библиотекой высокоуровневых математических функций для операций с этими массивами.

Также импортируется одна из важнейших библиотек, которая считает глобальную корреляцию, локальную корреляцию и классификацию зон:

- `import numpy as np;`
- `from scipy.stats import pearsonr.`

Далее идет импорт библиотеки, которая помогает разбить растр на определенные, нужные участки зон и потом посчитать в тех местах, коэффициент корреляции:

- `from aom_zones import;`
- `from aom_correl import inZoneCorrelationMap.`

3.2 Структура программы

В Python все программы организованы по модульному принципу, используя методы объектно-ориентированного программирования. Структура программы начинается с объявления ключевого модуля `def-define` (определение), каждый модуль - это файл. В модуль входит функция и объект, в данной работе используется только функции. Разработанный модуль состоит из функций (см. таблица 1)

Таблица 1 – Модуль `define`

Название функции	Передаваемые параметры	Комментарии
<code>def raster2array</code>	<code>filename</code>	Загружает растровое изображение в массив NumPy
<code>def nearest</code>	<code>loaded_image, w, h, x, y,</code>	Получает пиксель

Продолжение таблицы 1

Название функции	Передаваемые параметры	Комментарии
def diff	(loaded_image_1, loaded_image_2, out_filename, zone = None	Находится разница между двумя растрами и выводится в файл в градациях серого
def save_file	name, size, arr	Сохраняется получение глобальной, локальной корреляции и классификации зон
def process	filename_1, filename_2, output_file_name, correlation_type, zone_file=None, zone_num=None	В окне программы загрузка первого растра, второго растра, выходного растра. Выбор локальной, глобальной корреляции или классификации зон
def __init	self, start, end	Инициализация начала и конца
def __init	self, min, max	Выбор минимума и максимума.
def contains	self, point	Содержит какой-либо пункт
def localSpatialPearson	r1, r2, winSize=7	Загрузка формулы Пирсона для подсчета корреляции
mpZones2Stdout	imgArray, zoneRadius, transform=None	Выделение зон на растре

Выше перечислены, все объявления функции def (define). Директив #define определяет идентификатор и последовательность символов, которые будет замещать данный идентификатор, при его обнаружении в тексте программы. Обратим внимание, что в данном операторе отсутствует точка с запятой. Между идентификатором и последовательностью символов может быть любое число пробелов.

3.3 Чтение и запись растра из формата GEOTIFF

Формат GEOTIFF – это открытый формат представления растровых данных в формате TIFF, совместно с метаданными о географической привязке, картографической проекции, географических координат. Формат GEOTIFF нужен в данной работе потому, что работа идет с растровыми данными. GEOTIFF - это формат дистанционного зондирования, он нужен, чтобы читать данные из файла и записывать в файл, чтобы можно было открывать промежуточные результаты в Quantum GIS и просматривать их. Разработанный модуль состоит из функций:

Таблица 2 – Работа с GEOTIFF

Название функции	Передаваемые параметры	Комментарии
Raster Band Return band	gdal.Open(rasterfn) raster.GetRasterBand(1) raster.GetRasterBand(1)	Загрузка формата GEOTIFF, используется библиотека GDAL
def	raster2array(rasterfn)	загружается растровое изображение в массив numpy. Rasterfn
def	array2TIFF(newRasterfn,rasterSize,geoT ransformList,rasterProj,array) newRasterfn - имя файла. rasterSize - размер растра: tuple или list (ширина, высота) geoTransformList - привязка растра (параметры преобразования)	Для сохранения в массив numpy в GEOTIFF
dst_ds	SetGeoTransform(geoTransformList)	Выполняется геотрансформация

Продолжение таблицы 2

Название функции	Передаваемые параметры	Комментарии
outband	SetNoDataValue(-99) WriteArray(array) FlushCache()	Установка значений
dst_ds outband	SetProjection(rasterProj) dst_ds.GetRasterBand	Выбор проекции

Таким способом производится чтение и запись данных в формате GEOTIFF.

3.4 Расчет глобальной, локальной корреляции и классификации зон

Для начала работы нужно загрузить в программу два растра и выполнить глобальную корреляцию. Корреляция считается по всему растру. Растр раскладывается в два длинных ряда и считается коэффициент корреляции. Эту функцию выполняет библиотека Python NumPy.corrcoef:

- raster = gdal.Open(filename);
- band = raster.GetRasterBand(1);
- return band.ReadAsArray().

Растр раскладывается в два длинных ряда и считается коэффициент корреляции. Эту функцию выполняет библиотека Python NumPy.corrcoef. После этого загрузить растры в массивы, так можно обращаться к массиву, там будут все значения чисел в этом растре.

После получить ближайший пиксель к заданному пикселю:

- def nearest(loaded_image, w, h, x, y, square_size=7.

Получение корреляции между окрестностями двух пикселей на разных изображениях:

- `def localSpatialPearson (r1, r2, winSize=7);`
- `def correlate (loaded_image_1, loaded_image_2, w, h, x, y);`
- `param loaded_image:` загруженное изображение;
- `param x:` x-координата;
- `param y:` y-координата;
- `param square_size:` сторона квадрата "окна" вокруг заданной точки

(по умолчанию 7×7);

- `return:` массив пикселей из `square_size` в квадрате или меньше (если точки близко к краю).

Далее локальная корреляция. Загружается первый растр в виде матрицы, также второе изображение в виде матрицы, проверяется матрица одного размера или нет, далее берется скользящее окно пять на пять и сдвигается по каждому пикселю:

- `def localSpatialPearson (r1, r2, winSize=7).`

Загружается два растра:

- `r1` - первое изображение, представленное в виде матрицы (массив `numpy`);
- `r2` - второе изображение, представленное в виде матрицы (массив `numpy`).

Устанавливается размер окна.

- `winSize` - размер окна (по умолчанию=7).

Проверяется, чтобы матрицы имели одинаковый размер:

- `if r1.shape [0] != r2.shape [0] or r1.shape [1] != r2.shape [1];`
- `print("Input rasters have different size, adjust them!");`
- `sys.exit (1).`

Затем записывается в исходных матрицах, количество строк, радиус окна:

- `cols = int (r1.shape[1])`: количество столбцов в исходных матрицах;
- `rows = int (r1.shape[0])` количество строк;
- `int ((winSize-1)/2)` радиус окна.

Создается пустое изображение (матрица), в которую будет записываться результат. Выполняется выборка подматриц и превращение их в вектор-строки:

- `win1 = r1[row-d:row+d+1, col-d:col+d+1].ravel()`: первое окно, с функцией `ravel`;

- `win2 = r2[row-d:row+d+1, col-d:col+d+1].ravel()`: второе окно, с функцией `ravel`.

Вычисление коэффициента корреляции Пирсона:

- `r = np.corrcoef(win1, win2)[0,1]`.

Если получили NaN (нулевая дисперсия - деление на 0):

- `if np.isnan(r): r=-99`: заменяется NaN на NoData в GEOTIFF.

Записывается результат в выходное изображение:

- `out[outRow, outCol] = r`;
- `outCol+=1`;
- `outRow+=1`;
- `return out`.

Затем выполняется классификация по зонам, выделяется зона и в этой зоне считается коэффициент корреляции, например классификация растительности, и для нее выделяется на этом растре нужные пиксели.

Для начала выделяем зоны на третьем растре:

- `def dumpZones2Stdout(imgArray,zoneRadius,transform=None)`;
- `cols = int(imgArray.shape[1])`: количество столбцов;
- `rows = int(imgArray.shape[0])`: количество строк;
- `if visited[row,col]==0`: если эта ячейка еще не входит в зону;
- `if imgArray[row,col]<=0: continue`.

Создается зона, выполняется наращивание областей:

- `newZone=rasterZone(imgArray,visited,row,col,zoneRadius);`
- `newZone.dumpme(transform).`

После наращивания зон, выбирается необходимая зона, в которой нужно произвести корреляцию. Корреляция будет считаться только в этом выбранном участке. Также растр раскладывается в два длинных ряда и считает коэффициент корреляции. Эту функцию выполняет библиотека Python NumPy.corrccoef. После развертки, помещаем данные в формулу Пирсона и считается коэффициент корреляции.

3.5 Пользовательский интерфейс

Когда программа коэффициентов корреляции по глобальной, локальной корреляции и классификации зон подсчитана, далее при помощи Python и подключаемых библиотек, функций и модулей нужно разработать интерфейс для программы, чтобы возможно было загрузить программу расчета корреляций в Quantum GIS. Через программу QT Creator, будет нарисован интерфейс, окна, кнопки и оставшиеся компоненты.

Основная задача Qt Creator — упростить разработку приложения с помощью фреймворка QT на разных платформах. QT Creator кроссплатформенная свободная IDE для разработки на C, C++ , Python.

С сайта http://landsat.usgs.gov/band_designations_landsat_satellites.php, для выполнения корреляции, было скачано два растровых изображения. Далее в программе Quantum GIS, через операцию совмещение раstra, были совмещены два растровых изображения. В итоге получились два раstra, которые имеют одинаковую ширину, высоту, размер, размер ячейки, одинаковую начальную точку левого нижнего угла. Каждая клеточка раstra в пространстве совпадает с клеточкой с такими же координатами, что и на другом растре. После через программу Python и его библиотеки, модули, функции подсчитан коэффициент корреляции по глобальной, локальной корреляции и классификации зон.

Затем через QT Creator был создан интерфейс программы (рисунок 10).

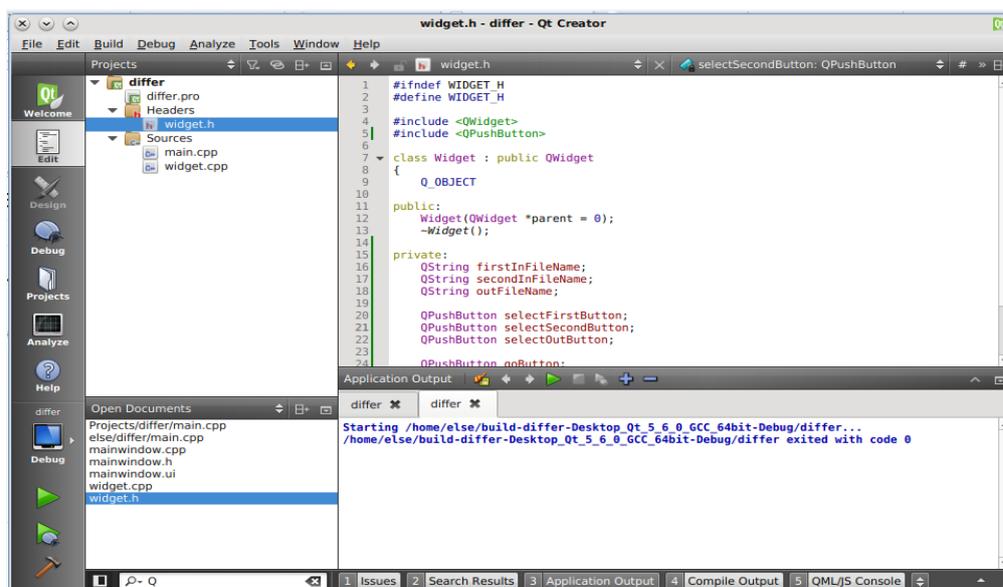


Рисунок 10 – Код QT Creator

В Qt Creator, через программный код были заданы значения и разработан пользовательский интерфейс. Показано визуальное проектирование интерфейса, виджеты обработчика визуального проектирования пользовательского интерфейса. Созданы окна, кнопки (рисунок 11).

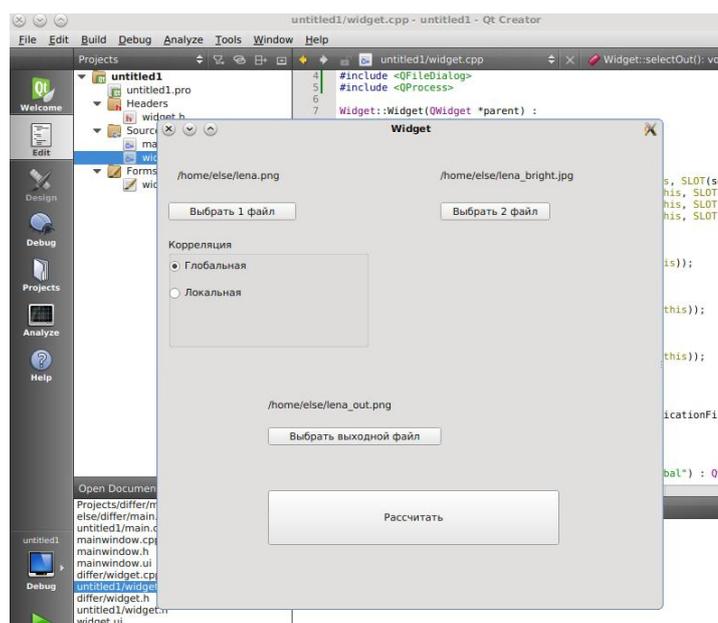


Рисунок 11 – Окно готового интерфейса

Через программу QT Creator был разработан интерфейс, окна, кнопки. Далее в программу нужно загрузить один растр, потом второй растр, потом выбрать необходимую классификацию и нажать кнопку рассчитать. После всех заданных параметров, программа подсчитает выбранную корреляцию и выдаст готовый третий растр.

3.6 Выводы по описанию элементов программирования

В разделе описания элементов программирования было рассмотрено:

- использование библиотек для чтения и записи данных;
- чтение и запись растра из GEOTIFF;
- расчет глобальной, локальной корреляции и классификации зон.

С помощью языка программирования Python, его библиотек, методов и функций удалось сделать расчет пространственной корреляции.

Для начала работы с языком программирования Python, были загружены необходимые для работы библиотеки, такие как: gdal, math, numpy, numpy as np, `from scipy.stats import pearsonr`, `from aom_zones import`, `from aom_correl import inZoneCorrelationMap`. Данные библиотеки предназначены, для работы с различными числами, многомерными массивами, матрицами, растровыми данными.

Далее через библиотеку GDAL было загружено расширение, для прочтения формата GEOTIFF. Формат GEOTIFF нужен в данной работе потому, что работа идет с растровыми данными, чтобы в любой момент можно было открывать промежуточные результаты в Quantum GIS и просматривать их.

Затем, когда все библиотеки, модули и функции были загружены в программу Python, можно приступить к расчету глобальной, локальной корреляции и классификации зон.

4 Тестирование разработанных программ

4.1 Результаты расчета глобальной корреляции

Для начала работы были взяты растры, через программу Quantum GIS были совмещены эти растры, задана проекция, выбран размер, также задана ширина, высота, размер ячейки и каждая клеточка растра в пространстве совпадает с клеточкой на другом растре. После работы с Quantum GIS был нарисован интерфейс (окна, кнопки), через программу QT Creator. Далее, после всех этих операций в интерпретаторе Python, при помощи его библиотек NumPy, Math, GDAL, написана программа для расчета локальной, глобальной корреляции и классификации зон. В результате получились три корреляции. Были взяты растры из Landsat 8, местность - Красноярское водохранилище (рисунок12).

Landsat-8 OLI and TIRS Bands (μm)		
30 m Coastal/Aerosol	0.435 - 0.451	Band 1
30 m Blue	0.452 - 0.512	Band 2
30 m Green	0.533 - 0.590	Band 3
30 m Red	0.636 - 0.673	Band 4
30 m NIR	0.851 - 0.879	Band 5
30 m SWIR-1	1.566 - 1.651	Band 6
100 m TIR-1	10.60 – 11.19	Band 10
100 m TIR-2	11.50 – 12.51	Band 11
30 m SWIR-2	2.107 - 2.294	Band 7
15 m Pan	0.503 - 0.676	Band 8
30 m Cirrus	1.363 - 1.384	Band 9

Рисунок 12 – Каналы Landsat 8

Данная корреляция считается по всему растру, растр раскладывается в два длинных вектора ряда и считается коэффициент корреляции. После этого, растры загружаются в массивы с помощью функции `ravel`, двумерный массив, можно развернуть в два длинных вектора строки, получится большое количество членов этого ряда. После развертки, помещаются данные в формулу Пирсона и считается коэффициент корреляции. Эту функцию выполняет библиотека Python `NumPy.corrcoef`. В итоге каждый канал сравнивается друг с другом и получаются коэффициенты корреляции Пирсона между изображениями каналов Landsat 8 (рисунок 13).

	1	2	3	4	5	6	7
1	1	0,9896	0,8696	0,9126	-0,0785	0,6556	0,7970
2	0,9896	1	0,9074	0,9539	-0,0374	0,7012	0,8278
3	0,8696	0,9074	1	0,9281	0,3210	0,8549	0,8257
4	0,9126	0,9539	0,9281	1	0,0542	0,7866	0,8852
5	-0,0785	-0,0374	0,3210	0,0542	1	0,5135	0,1458
6	0,6556	0,7012	0,8549	0,7866	0,5135	1	0,8961
7	0,7970	0,8278	0,8257	0,8852	0,1458	0,8961	1

Рисунок 13 – Коэффициенты корреляции Пирсона между изображениями каналов Landsat 8

На рисунке 13 показано, как друг с другом связаны каналы. Первый канал со вторым, с третьим, с четвертым и седьмым имеют очень высокую связь, с шестым чуть меньше, а пятый канал сильно отличается от всех остальных. Для того, чтобы изображения были информативны, нужно чтобы каналы отличались друг от друга, на них лучше всего будет выделяться объект.

4.2 Результаты расчета локальной корреляции

После того как посчитана глобальная корреляция и видно, какие каналы не похожи друг на друга, можно посчитать локальную корреляцию. Проверяется коэффициент корреляции не полностью, а в определенных местах. Берется первый растр в виде матрицы, также второе изображение в виде матрицы, проверяется матрица одного размера или нет, далее берется скользящее окно семь на семь и сдвигается по каждому пикселю. Передается массив NumPy первый и второй, проверяется его размерность, количество строк. Создается пустое изображение из нулей, и выполняется обход растра по строкам и столбцам, выделяется подматрица. Библиотека `ravel` из матрицы делает развертку вектора, которая проходит по изображению, также как в глобальной корреляции, разворачивается в две длинные строки вектора и с каждым сдвигом, по каждой клетке, считается корреляция.

Считается коэффициент корреляции между первым и шестым каналом, так как они отличаются друг от друга (рисунок 14).

Приведенная гистограмма красная и совмещена в положительную область – это означает, что имеется сильная взаимосвязь. Слева на гистограмме указано количество пикселей, а снизу значение. На изображении видно, что пикселей имеющих положительную взаимосвязь гораздо больше, чем пикселей имеющих низкую взаимосвязь.

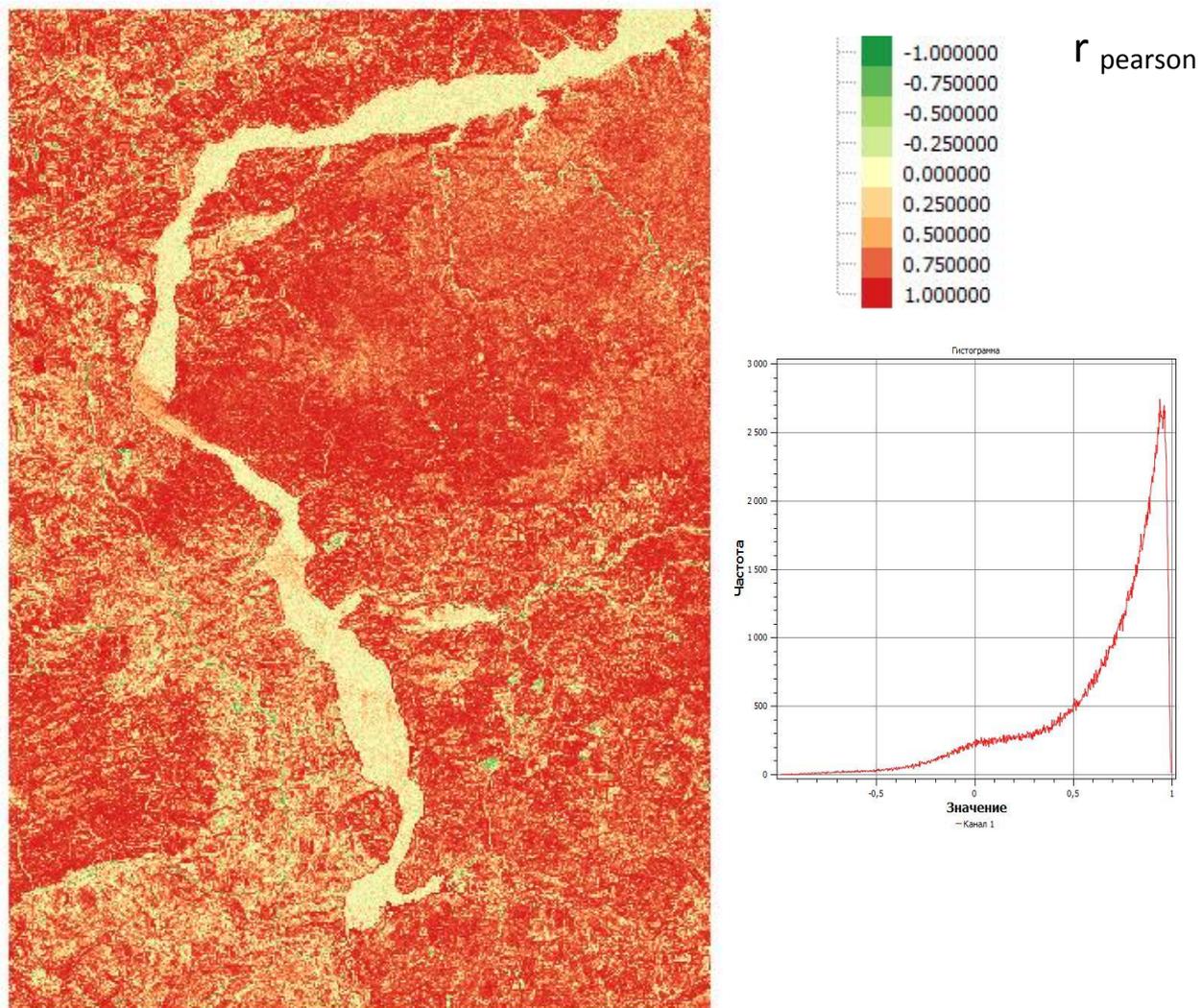


Рисунок 14 – Результат локальной корреляции между первым и шестым каналом

Далее рассчитывается локальная корреляция по каналу два и пять (рисунок 15).

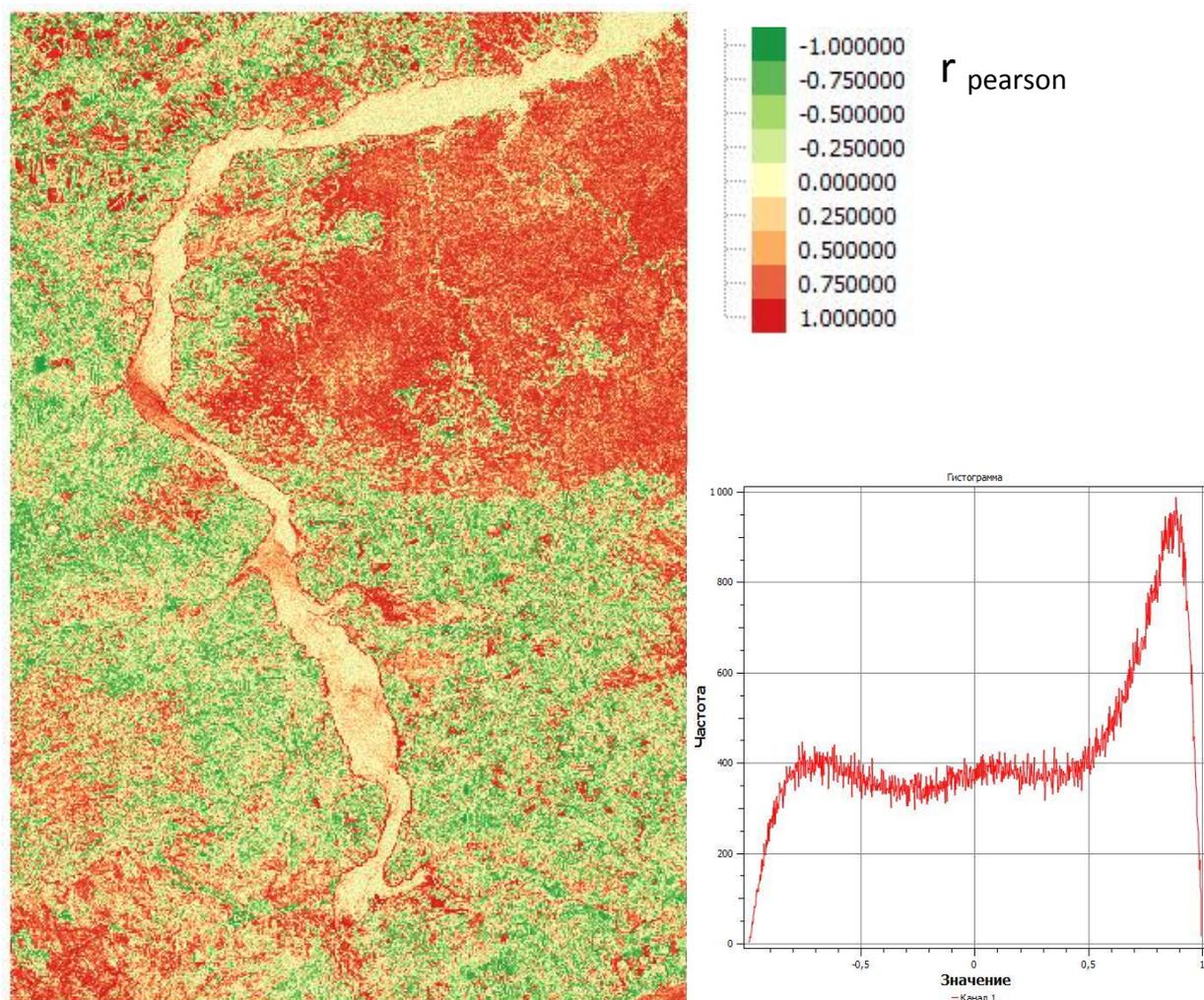


Рисунок 15 – Результат локальной корреляции между вторым и пятым каналом

Гистограмма второго и пятого канала, отличается от гистограммы первого и шестого канала. На данной корреляции другое распределение локальных коэффициентов корреляции, потому что у пятого канала глобальный коэффициент ниже, чем у шестого. Здесь есть области выделенные красным, они означают очень высокий коэффициент корреляции, также есть области, где низкие коэффициенты корреляции, которые суммируясь, дают ноль. На изображении видны красные, желтые и зеленые пятна. Красными пятнами выделены леса, зелеными - сельскохозяйственные земли, а желтыми – пахотные

земли. Разные ландшафты, разные типы поверхности по-разному отображаются на взаимосвязи этих двух каналов.

4.3 Классификации зон

С сайта Global Land Cover Facility <http://glcf.umd.edu/data/lc> выбирается глобальная карта и проводится визуальная классификация. Это делается для того, чтобы посмотреть и понять какой цвет, что обозначает на готовой, рассчитанной карте локальной корреляции, первого и шестого канала, также второго и седьмого канала. Все это поможет понять где лес, вода, пахотные поля, степи, города (рисунок 16).

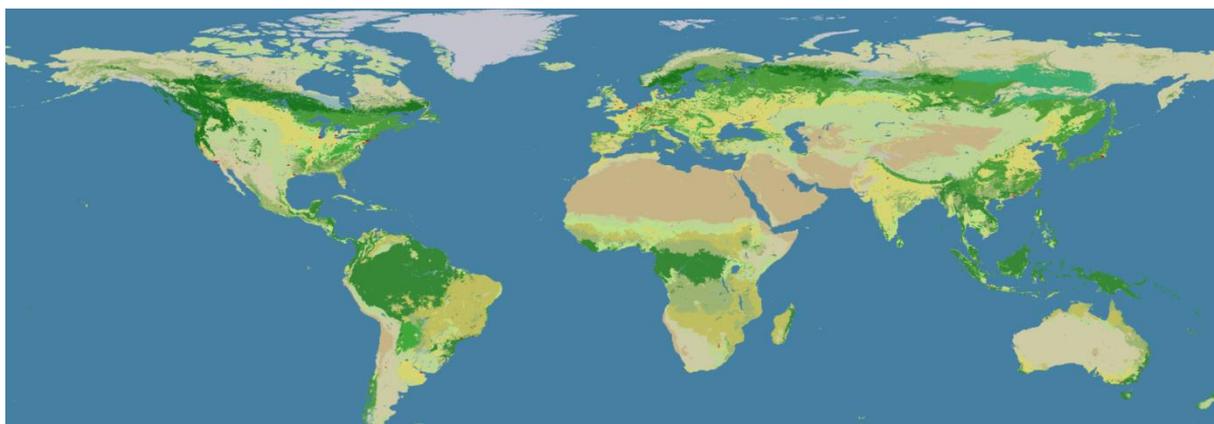


Рисунок 16 – Глобальная карта Land Cover MODIS

С глобальной карты берется и вырезается кусок карты MODIS Land Cover, с местностью Красноярского водохранилища и разрешением пять минут. Для того, чтобы визуальное увидеть, какой цвет что обозначает на карте локальной корреляции. После того, как был вырезан кусок карты, можно более подробно рассмотреть цвета (рисунок 17)



Рисунок 17 – Карта по съемки MODIS Красноярского водохранилища

Обозначение цветов на карте:

- 12 - желтый цвет – пахотные земли;
- 0 - синий цвет – вода;
- 5 - зеленый цвет – смешанные леса;
- 13 - красный цвет – степи.

Смотря на карту локальной корреляции и карту по съемке MODIS Красноярского водохранилища, можно визуально увидеть, что обозначает каждый цвет. Для воды корреляция нулевая, для степной и лесной растительности высокая корреляция, а для пахотных земель обратная высокая взаимосвязь. Теперь понятно, где что находится на картах локальной корреляции. Красными пятнами выделены города, а зелеными -

сельскохозяйственные земли, желтыми - пахотные земли. Разные ландшафты, разные типы поверхности, по-разному отображаются на взаимосвязи этих двух каналов.

4.4 Выводы по тестированию разработанных программ

Рассчитано два вида корреляции: глобальная и локальная. Было взято семь каналов из Landsat 8, с местностью - Красноярского водохранилища. В глобальной корреляции, с помощью коэффициент корреляции Пирсона, каждый канал сравнивается друг с другом. На рисунке 13 показано, как друг с другом связаны каналы. Первый канал со вторым, с третьим, с четвертым и седьмым имеют очень высокую связь, с шестым чуть меньше, а пятый канал сильно отличается от всех остальных. Для того, чтобы изображения были информативны, нужно чтобы каналы отличались друг от друга, тогда на них лучше всего будет выделяться объект.

Далее, после того как посчитана глобальная корреляция и видно какие каналы не похожи друг на друга, можно посчитать локальную корреляцию. Был взят первый и шестой канал, также второй и пятый, потому что они отличаются друг от друга. Но между первым и шестым, вторым и пятым имеется разница. Гистограмма второго и пятого канала отличается от гистограммы первого и шестого канала. Здесь другое распределение локальных коэффициентов корреляции, потому что у пятого канала глобальный коэффициент ниже, чем у шестого. Здесь есть области выделенные красным, они означают очень высокий коэффициент корреляции, также есть области, где низкие коэффициенты корреляции, которые суммируясь, дают ноль. На изображениях видны красные, зеленые, желтые пятна. С сайта Global Land Cover Facility <http://glcf.umd.edu/data/lc> берется карта и проводится визуальная классификация. Для того, чтобы посмотреть и понять какой цвет, что обозначает на готовой, рассчитанной карте локальной корреляции, первого и

шестого канала, также второго и седьмого канала. Это поможет понять, где лес, вода, пахотные поля, степи.

Красными пятнами выделены города, а зелеными - сельскохозяйственные земли, желтые - пахотные поля. Разные ландшафты, разные типы поверхности по-разному отображаются на взаимосвязи этих двух каналов.

ЗАКЛЮЧЕНИЕ

Цель и задачи, поставленные в работе, выполнены. Разработаны алгоритмы для расчета локальных и глобальных коэффициентов корреляции, с учетом зон для двух растровых изображений, принципы их работы и реализации. Также разработан пользовательский интерфейс, для программ расчета пространственной корреляции, при помощи QT Creator. Проведено исследование возможностей языка Python и его библиотек NumPy, GDAL, Math. Также проведена работа с программой Quantum GIS это – совмещение растров, выбор методов, перепроецировка, изменения размера, размера ячейки, обрезка растров.

Программа Quantum GIS содержит возможность для программирования, для улучшения функциональности, для тонкой настройки ГИС под нужным пользователем. Используя различные языки программирования, можно настроить расширение ГИС. В Quantum GIS - это интерпретатор Python, на котором можно построить разные расширения, это стандартный язык для геообработки. Для того чтобы писать программы - используется Python, чтобы программы работали быстрее - используется язык программирования C. Чтобы нарисовать окна в программе - используется программа QT Creator, для расчетов программы - используется NumPython.

Язык Python является одним из самых популярных и быстрых скриптовых языков программирования. Также Python бесплатный, распространяемый, удобный в использовании, позволяет решать сложные задачи, позволяет управлять геоданными, отображать, редактировать, анализировать их, также создавать макеты карт.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Python [Электронный ресурс] // Python . – Режим доступа: <https://http://dimonvideo.ru/articles/1368/>
2. Разработка Python [Электронный ресурс] // Разработка, Python на Хабре. – Режим доступа: <https://habrahabr.ru/post/150302/>
3. Python [Электронный ресурс] // учебник Python 3.1. – Режим доступа: <https://ru.wikibooks.org>
4. QGIS [Электронный ресурс] // Quantum GIS – Режим доступа: <https://gis.web.tsti.ru>
5. GDAL [Электронный ресурс] // GIS-LAB – Режим доступа: <http://gis-lab.info/>
6. Python 3 для начинающих [Электронный ресурс] // NumPy, часть 1: начало работы. – Режим доступа: <http://pythonworld.ru/numpy/1.html>
7. Python 3 для начинающих [Электронный ресурс] // Модуль Math. – Режим доступа: <http://pythonworld.ru/moduli/modul-math.html>
8. QT Creator Python [Электронный ресурс] // Все о кроссплатформенном программировании – Режим доступа: <http://doc.crossplatform.ru/>
9. Анализ ближайшего соседства [Электронный ресурс] // QGIS Tutorials and Tips – Режим доступа: <http://www.qgistutorials.com/>
10. Билинейная интерполяция [Электронный ресурс] // GIS-LAB – Режим доступа: <http://gis-lab.info/>
11. Метод аппроксимации кубической полиномы [Электронный ресурс] // Образовательный блог – Режим доступа: <http://all4study.ru/>
12. Расчет среднего значения [Электронный ресурс] // GIS-LAB – Режим доступа: <http://gis-lab.info/>
13. Глобальная карта [Электронный ресурс] // Global Cover Facility – Режим доступа: <http://glcf.umd.edu/data/lc>

14. Снимки Landsat 8 [Электронный ресурс] // USGS – Режим доступа:
<https://www.usgs.gov/>