ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc137484066)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 4](#_Toc137484067)

[1.1. Основные определения 4](#_Toc137484068)

[1.2. Представление цифрового изображения 5](#_Toc137484069)

[1.3. Анализ методов сегментации цифрового изображения 8](#_Toc137484070)

[1.3.1. Метод k-средних 8](#_Toc137484071)

[1.3.1.1. Улучшение метода k-средних с помощью роевого алгоритма 10](#_Toc137484072)

[1.3.1.2. Улучшение с помощью эвристики Exponential PSO. 14](#_Toc137484073)

[1.3.1.3. Поиск кластера со зданиями. 14](#_Toc137484074)

[1.3.2. Метод выделения контуров 15](#_Toc137484075)

[1.3.2.1. Перекрестный градиентный оператор Робертса 16](#_Toc137484076)

[1.3.2.2. Градиентный оператор Превитта. 16](#_Toc137484077)

[1.3.2.3. Лапласиан 17](#_Toc137484078)

[1.3.2.4. Получение контурного препарата 18](#_Toc137484079)

[1.4. Выводы по главе 1 19](#_Toc137484080)

[2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММЫ СЕГМЕНТИРОВАНИЯ СПУТНИКОВЫХ ИЗОБРАЖЕНИЙ 21](#_Toc137484081)

[2.1. Разработка программы 21](#_Toc137484082)

[2.2. Тестирование. 26](#_Toc137484083)

[2.3. Выводы по главе 2 29](#_Toc137484084)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30](#_Toc137484085)

ВВЕДЕНИЕ

Современные спутниковые снимки представляют собой богатый источник информации для различных исследований и задач геопространственного анализа. В частности, одной из таких задач является выделение зданий на спутниковых снимках. Эта задача является важной для многих применений, включая городское планирование, мониторинг изменений городской среды и создание карты городской застройки.

Цель данной работы - разработать алгоритм выделения зданий на спутниковых снимках с использованием метаэвристических алгоритмов и компьютерного зрения. Чтобы достичь этой цели, в работе будут рассмотрены различные методы обработки изображений, а также алгоритмы сегментации и классификации объектов на изображении. Результаты работы могут быть использованы для создания высокоточных карт городов и улучшения прогнозирования изменений в городской застройке.

Актуальность данной бакалаврской работы заключается в том, что выделение зданий на спутниковых снимках является важным элементом в развитии геоинформационных технологий, так как с помощью них можно получить точную карту города с множеством информации о расположении зданий, дорог, парков и других объектов, что значительно облегчает принятие управленческих решений.

Более того, выделение зданий снимках также имеет большое значение в сфере безопасности. Например, это может быть использовано для анализа изменений в городском ландшафте, определения областей с повышенным риском возникновения пожаров или других ЧС, а также для мониторинга строительных работ и нарушений зонирования города.

В целом, развитие алгоритмов выделения зданий на спутниковых снимках не только увеличивает точность и быстроту анализа информации, но и затрагивает большое количество областей жизни, что делает данную работу важной и актуальной.

Цель исследования: разработать программу на высокоуровневом языке программирования для сегментации спутниковых изображений для анализа плотности застройки.

Задача исследования:

1. Изучить теоретическую информацию по данной теме.
2. Проанализировать существующие методы сегментации спутниковых изображений.
3. Выбрать наилучший алгоритм с точки зрения быстродействия и точности выделения объектов.
4. Разработать соответствующую программу на языке Python.
5. Получить карту плотности застройки в конкретном регионе (по одному снимку)
6. Масштабировать полученный алгоритм на серию снимков, охватывающих всю городскую территорию.
7. Проанализировать полученный результат и сделать выводы о соответствии плотности застройки нормам.

Методы исследования: метод теоретического исследования (анализ и изучение источников) и практический анализ (эксперименты и моделирование).

Теоретическая значимость: проведен анализ существующих методов сегментации спутниковых изображений, а также рассмотрены способы их оптимизации. Аналитическая информация может служить полезной информацией для начального и последующего изучения данной темы.

Практическая значимость: экспериментальные исследования и разработанная программа. Возможность использовать разработанную программу для внедрения в систему с целью создания высокоточных карт городов и улучшения прогнозирования изменений в городской застройке.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Основные определения

Цифровое изображение – двумерное изображение, представленное в цифровом виде. В зависимости от способа описания, изображение может быть растровым или векторным.

Растровое изображение – изображение, представляемое как прямоугольный двумерный массив чисел, при этом каждое число соответствует одному элементу изображению или пикселю.

Векторное изображение – изображение, полученное путем математического описания элементарных геометрических объектов, обычно называемых примитивами.

Пиксель – наименьший логический элемент двумерного цифрового изображения в растровой графике, представляющий собой неделимый объект прямоугольный формы, характеризуемый определенным цветом.

Цветовая модель – математическая модель описания представления цветов в виде кортежей чисел (трех или четырех значений), называемых цветовыми компонентами.

RGB-модель – аддитивная цветовая модель, описывающая способ кодирования цвета с помощью трех основных цветов: красного, зеленого и синего.

Сегментация – процесс разделения цифрового изображения на несколько сегментов, целью которого является упрощение и/или изменение представление изображения для простоты и удобства анализа.

Шум – дефект, проявляемый в виде черных и белых пикселей на полутоновом изображении.

1.2. Представление цифрового изображения

В выпускной квалификационной работе будут обрабатываться растровые изображения RGB-модели. Данные изображения описываются разрешением (высотой и шириной снимка), а также значением пикселей, входящих в него. Каждый пиксель представляет собой набор их трех значений, цветовых компонент, каждая из которых принимает значения от 0 до 255. При этом меньшее значение компоненты соотносится с темным оттенком; большее – со светлым. Так, за счет комбинаций компонент, пиксель можно окрасить в один из 16.7 миллионов цветов и расположить в трехмерном пространстве (рисунок 1.1).



Рисунок 1.1 – RGB-цветовая модель, представленная в виде куба с нанесенными на нем точками и значениями компонент.

Некоторые алгоритмы сегментации изображений используют оттенки серого - цветовой режим изображений, которые отображаются в оттенках серого цвета, размещённые в виде таблицы в качестве эталонов яркости белого цвета (рисунок 1.2).



Рисунок 1.2 – 4096 уровней серого.

Преобразовать цветное изображение в оттенки серого можно несколькими способами, получив яркость Y, находящуюся как:

1. . Метод усреднения. Данный метод дает более черное изображение, нежели другие алгоритмы. Проблема возникает из-за того, что каждый цвет имеет свою длину волны и собственный вклад в формирование изображения. Потому, если брать среднее из трех цветов, общий баланс белого нарушается.

2. . Модель и . Метод светимости. Для данного уравнения коэффициенты были подобраны с учетом восприятия изображения человеческим глазом (чувствительность к зеленому и синему цвету)

Сравнение двух методов можно наблюдать на рисунках 1.3-1.4.



Рисунок 1.3 – Исходное изображение.



Рисунок 1.4 – Преобразованное изображение а) – методом усреднения;  
б) – методом светимости.

Серая шкала — это контрольное изображение равномерного ряда оптических плотностей нейтрально-серых полей, изготовленных на прозрачной или непрозрачной основе и предназначенное для оценки и измерений качества тонопередачи. Такая шкала передаёт 256 оттенков (градаций) серого цвета, или яркости (значение 0 представляет чёрный цвет, а значение 255 — белый).

Серая шкала отражает интенсивность света в каждом пикселе видимой части электромагнитного спектра. Так как серая шкала располагается на диагонали в цветовом кубе модели RGB, то каждая составляющая получает одинаковые значения, равные значениям оттенков серого. Таким образом, на этом этапе каждый пиксель изображения приводится к виду .

1.3. Анализ методов сегментации цифрового изображения

1.3.1. Метод k-средних

Так как в общем виде задача кластеризации выражена в необходимости выделения групп точек в каком-либо многомерном пространстве для решения проблемы выделения объектов на изображении будем представлять каждый пиксель изображения как точку с тремя координатами:

где , , – красная, зеленая и синяя составляющие пикселя изображения, представленного в цветовой модели RGB.

Цель сегментации изображения – разбить все пиксели на заданном число кластеров. Будем полагать, что расстояния между точками внутри кластера мало по сравнению с расстояниями до внешних точек. Для формализации данного выражения введем набор векторов , где , и назовем каждый такой набор центром кластера . Так, необходимо найти такую принадлежность пикселей к кластерам, чтобы сумма квадратов расстояний от каждого пикселя до ближайшего центра кластера была минимальной.

Для представления алгоритма K-средних опишем принадлежность точек к кластеру с помощью набора бинарных переменных , где , показывающих, что пиксель принадлежит к кластеру , если и для . Это представление также известно как «1-из-K».

Теперь можно определить целевую функцию, которую мы будем минимизировать:

Так как каждый пиксель и центр кластера определяется набором из трех координат, формула выше приобретает вид:

Остается найти значения для и , чтобы функция была минимальна. Обратимся к итеративным процедурам, где на каждой итерации будем выполнять два последовательных шага:

1. Определяем принадлежность пикселей к кластерам (изменяем значения при фиксированных значениях ). Формально это можно выразить с помощью формулы:

2. Определяем центры кластеров (изменяем значения при фиксированных значениях ). Так как целевая функция – квадратичная функция, она может быть минимизирована с помощью приравнивая производной по к нулю, получая выражение:

откуда легко выражается:

Пример сегментации точек с двумя параметрами (положение по оси и ) показан на рисунке 1.5.

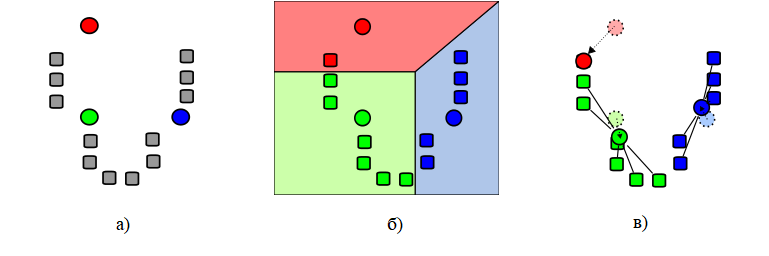


Рисунок 1.5 – Пример сегментации методом k-средних; а) – исходные точки и случайно выбранные центры кластеров; б) – точки, отнесенные к начальным центрам; в) – пересчитанные центры кластеров.

Алгоритм завершается, когда на какой-то итерации не происходит изменения внутриклассного расстояния. Это происходит за конечное число итераций, так как количество возможных разбиений конечного множества конечно, а на каждом шаге суммарное квадратичное отклонение уменьшается, поэтому зацикливание невозможно.

1.3.1.1. Улучшение метода k-средних с помощью роевого алгоритма

Очевидно, слабое место метода k-средних – случайная инициализация центров кластеров, а также возможность попадания в локальный минимум. Для решения этих проблем используется инициализация, основанная на оптимизации роевым алгоритмом (PSO-Based Initialization).

Для реализации PSO-алгоритма населим наше -мерное пространство роем из частиц. Координата -ой частицы () является вектором , который и определит набор факторов оптимизации. Очевидно, что в данном случае главной целью роевого алгоритма является поиск таких центров кластеров, что послужат хорошей отправной точкой для метода k-средних. Тогда будем проводить поиск в пространстве всех возможных центров, потому логично положить – центром -ого кластера.

Как было описано в пункте 1.3.1, , в свою очередь, представляется в виде вектора трех цветовых компонент: красного, зеленого и синего - .

Будем полагать, что любое спутниковое изображение можно сегментировать на три кластера: лесные насаждения, водные и теневые участки, а также сооружения – тогда каждую частицу роя можно представить в виде двумерного массива (таблица 1.1)

Таблица 1.1. Представление частицы в рое для поиска трех кластеров

|  |  |  |  |
| --- | --- | --- | --- |
|  | Центр 1-ого кластера | Центр 2-ого кластера | Центр 3-ого кластера |
| Красная компонента |  |  |  |
| Зеленая компонента |  |  |  |
| Синяя компонента |  |  |  |

Каждая -я частица обладает характеристиками: скорость и местоположение , меняющееся со временем . Очевидно, на каждом шагу позиция частицы в пространстве будет изменяться на величину скорости:

Инициализировать рой необходимо случайным образом так, чтобы частицы располагались по всей области поиска. Каждая -я частица также имеет свой собственный вектор скорости , который в каждый момент времени соответствует конкретной итерации PSO-алгоритма. Так, в случае выделения трех кластеров, вектор каждой частицы также представляется в виде двумерного массива размером 3 на 3.

Для каждой позиции -мерного пространства поиска, в котором побывала -я частица, вычисляется значение целевой функции, которую роевый алгоритм стремится минимизировать.

При этом каждая -я частица запоминает свое лучшее значение целевой функции, а также координаты позиции, соответствующие данному значению.

Естественно, частица не должна двигаться в одном направлении, потому необходимо ввести коррекцию скорости. Так, для каждой -й координаты вектора скорости () -й частицы выполняется корректировка в соответствии с формулой:

где – -я компонента скорости -й частицы в момент времени ; – -я координата позиции частицы ; – коэффициенты локального и глобального ускорения соответственно; – случайные величины на отрезке ; – лучшая позиция -й частицы по координате для всех итераций (когнитивная составляющая); – лучшая позиция всех частиц для всех итераций (глобальная составляющая).

Для определения лучшей частицы необходимо ввести конкретную формулу для вычисления целевой функции для каждого набора кластеров, которая будет одновременно минимизировать среднее евклидово расстояние от частиц до связанных с ними кластеров, а также максимизировать евклидово расстояние между парами кластерных центров:

где для - изображения; – матрица связей пикселей и принадлежностью к кластеру -й частицы, каждый элемент которой принимает целочисленное значение от до – номеру кластера; и – коэффициенты учета параметров; – максимальное среднее евклидово расстояние от частиц до связанных с ними кластеров, вычисляемое по формуле:

где расстояние вычисляется как:

– минимальное евклидово расстояние между парами кластерных центров, вычисляемое по формуле:

Таким образом, целевая функция за счет данных функций предполагает выделение компактных и хорошо разделенных кластеров, а весовые коэффициенты и позволяют работать с алгоритмом для разного рода входных данных.

Так, когнитивная составляющая роя будет рассчитываться как:

Социальная же составляющая роя будет рассчитываться как:

где – общее число частиц роя.

1.3.1.2. Улучшение с помощью эвристики Exponential PSO.

В ходе поиска центров кластеров частицы будут стремиться лететь непосредственно к позиции, благодаря чему перспективные решения поставленной задачи обнаруживаются за довольно короткое время. Тем не менее, именно данное социальное взаимодействие может приводить к попаданию в локальный, а не глобальный минимум. Как только находится лучшее значение для какой-то частицы, все остальные устремляются к данной позиции. Улучшение в эвристике Exponential PSO состоит в том, что в формулу изменения скорости добавляется изменяющийся со временем коэффициент инерции :

Он изменяется с каждой итерацией от до с помощью формулы:

Благодаря введению данного коэффициента движение частиц становится более плавным и сжимающимся к локальному минимуму ближе к концу алгоритма.

1.3.1.3. Поиск кластера со зданиями.

Выходными данными для метода k-средних будет набор из трех кластеров. Последней проблемой для выделения зданий является соотнесение центров кластеров и объектов на снимке. Так, эмпирически было выделено, что преимущественно для снимков можно выделить три закономерности:

* Фрагменты изображения со зданиями имеют преимущественно светлые тона и относятся к кластеру с наибольшей суммой координат центра.
* Фрагменты, содержащие воду и тени имеют темные тона и относятся к кластеру с наименьшей суммой координат центра.
* Фрагменты, содержащие лесные или травяные насаждения, очевидно, имеют ярко выраженный зеленый цвет, а значит, относятся к кластеру с наибольшей суммой разниц между зеленой и красной/зеленой составляющими.

Таким образом, вышеописанные факты можно формализовать в виде формул. Кластер со зданиями будем находить по формуле:

Кластер с водой и тенями находится по формуле:

Кластер с лесами можно вычислить методом исключения или по формуле:

Таким образом можно однозначно определить кластер и объект, который он сегментирует.

1.3.2. Метод выделения контуров

Выделение границ (выделение краёв) — термин в теории обработки изображения и компьютерного зрения, основывающийся на алгоритмах, которые выделяют точки цифрового изображения, в которых резко изменяется яркость или есть другие виды неоднородностей.

Для того, чтобы найти границы объектов, необходимо найти перепады яркостей, которые могут представлять собой как границы объектов, так и градиентные изменения яркости, вызванные плавными изменениями освещенности, коэффициентов отражения или углового положения оболочек самих объектов. При сегментации представляют интерес перепады яркости, обусловленные границам объектов, поскольку целью сегментации, очевидно, является выделение объектов.

Перепады этого типа (рисунки 6 и 7)в идеале представляют собой резкие скачки яркости, однако на практике чаще приходится иметь дело с размытыми световыми границами вследствие апертурных искажений, вносимых системой. На практике применяют два метода обнаружения перепадов яркости, образуемых световыми границами: градиентный метод и метод с использованием лапласиана.

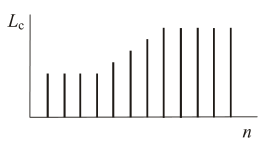
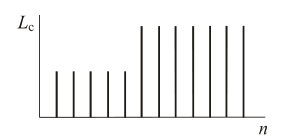


Рисунок 6. Перепад яркости в случае Рисунок 7. Перепад яркости идеально резкой границы для случая размытой границы

По определению градиент аналогового изображения в точке представляет собой вектор , ориентированный в направлении максимального изменения яркости, модуль которого равен

В случае дискретных изображений частные производные и определяются одним из следующих способов.

1.3.2.1. Перекрестный градиентный оператор Робертса

Согласно этому способу, дискретное изображение сканируется окном размером пиксела, показанное на рисунке 8.

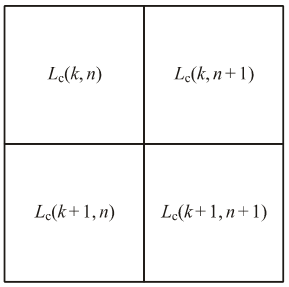


Рисунок 8. Сканирующее окно при использовании  
оператора Робертса.

Для каждого положения окна вычисляются значения и по формулам

а затем по формуле для пиксела, расположенного в -ой строке и в -м столбце, вычисляется модуль градиента.

1.3.2.2. Градиентный оператор Превитта.

В отличие от оператора Робертса, при использовании оператора Превитта дискретное изображение сканируется окном размером пиксела, которое показано на рисунке 9.

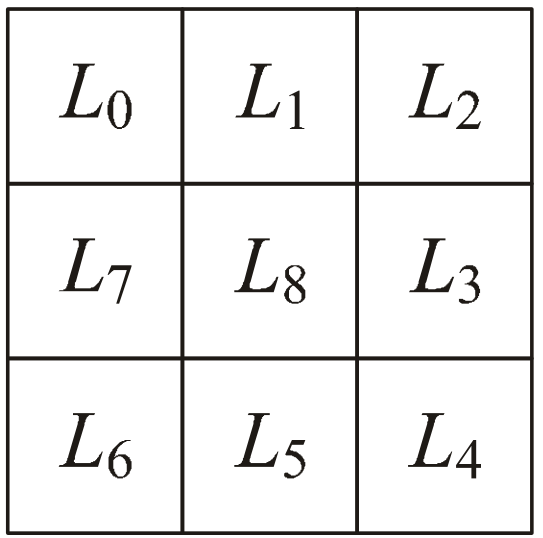


Рисунок 9. Сканирующее окно размером пиксела.

При этом для каждого положения окна значения и вычисляются по формулам

где по-прежнему – значения яркостей пикселов, оказавшихся в пределах окна, показанного на рисунке 9. Далее вычисляется модуль градиента для пиксела изображения, находящегося в центре сканирующего окна.

1.3.2.3. Лапласиан

Лапласиан дискретного изображения для пиксела, расположенного в -й строке и в n-м столбце определяется приближенно одним из следующих двух способов:

В первом случае результат оказывается инвариантным к повороту на углы, кратные 90°, во втором случае инвариантным к повороту на углы 45°.

Недостатком метода обнаружения перепадов яркости с использованием лапласиана по сравнению с градиентным методом является его большая чувствительность к шуму, который всегда имеется на изображениях.

1.3.2.4. Получение контурного препарата

После того, как на изображении были найдены яркостные перепады, результаты сравниваются с порогом. Если результат обнаружения превышает порог, то считается, что этот пиксел изображения принадлежит контуру и ему присваивается значение яркости контура , в противном случае ему присваивается значение яркости фона в формируемом контурном изображении.

В результате такой обработки изображения получается его контурный препарат. Выделенные контуры на контурном препарате обычно разорваны во многих местах, кроме того, на нем имеются точки и штрихи, которые воспринимаются зрительной системой как помеховый фон. На рисунке 10 показаны: исходное изображение (а) и его контурный препарат (б), на котором хорошо видны отмеченные особенности.

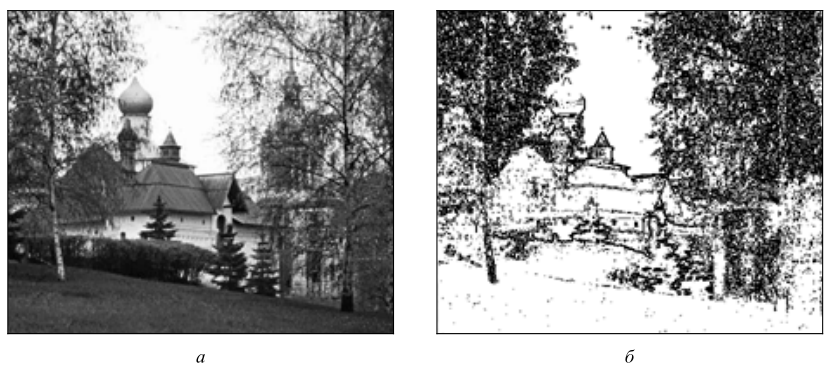


Рисунок 10. Изображение (а) и его контурный препарат.

Выбирая величину порога, можно несколько уменьшить эти неприятные артефакты, однако совсем от них избавиться не удается. Чем выше порог, тем меньше на контурном изображении будет отдельных точек и штрихов, не являющихся элементами выделяемых контуров, однако при этом в выделенных контурах увеличивается количество и протяженность разрывов. Уменьшение величины порога приводит к обратной картине. В некоторой степени количество возникающих на контурном изображении элементов, не принадлежащих контурам, может быть уменьшено путем применения медианной или логической фильтрации, но добиться их полного устранения не удается.

Приведем оптимальные пороговые значения, выведенные экспериментально, для каждого метода:

Таблица 1 – Пороговые значения для методов выделения контуров

|  |  |  |  |
| --- | --- | --- | --- |
| Метод | Робертса | Превитта | Лапласиан |
| Пороговые значения | 50 | 150 | 100 |

1.4. Выводы по главе 1

Проанализировав два наиболее часто используемых алгоритма сегментации изображений, можно сделать вывод о рациональности их использования в решении задачи выделения зданий на спутниковых изображениях.

1. Метод k-средних и выделения контуров имеют сравнительно высокую эффективность при простоте реализации.

2. Для метода k-средних количество кластеров является параметром алгоритма, что является существенным недостатком, однако при выделении зданий можно предполагать их количество (тени, здания и зеленые участки).

3. Оба алгоритма чувствительны к шумам. Для метода k-средних выбросы, далекие от центров настоящих кластеров, все равно учитываются при вычислении их центров.

4. Если границы домов нечеткие или сливаются с другими объектами необходимо точечно выбирать параметры каждого из алгоритмов.

Таким образом, можно утверждать, что оба алгоритма могут хорошо проявить себя при выделении зданий на снимке, а также то, что их целесообразно использовать одновременно.

2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММЫ СЕГМЕНТИРОВАНИЯ СПУТНИКОВЫХ ИЗОБРАЖЕНИЙ

2.1. Разработка программы

Приступим к проектированию программы с исходных данных. Программе должны подаваться изображения формата PNG с RGB-моделью цветового представления. Для получения таких снимков воспользуемся открытым источником Google Earth. Выберем для тестового снимка масштаб и место такие, чтобы в кадр попало как можно больше уникальных строений и локаций. Пример такого снимка можно увидеть на рисунке 11.



Рисунок 11. Спутниковый снимок с присутствующими на нем водоемами и лесными массивами.

Первым шагом для обработки снимка выступит считывание информации о пикcелях и размере снимка. Для этого воспользуемся библиотекой PIL для языка Python и сформируем двумерный массив (а также список, который понадобится для отображения изображения в matplotlib):

pixels\_list = list(image.getdata())

pixels\_array = []

image\_rs = []

image\_gs = []

image\_bs = []

for i in range(image.size[1]):

image\_row = []

for j in range(image.size[0]):

image\_row.append(pixels\_list[i \* image.size[0] + j])

image\_rs.append(pixels\_list[i \* image.size[0] + j][0])

image\_gs.append(pixels\_list[i \* image.size[0] + j][1])

image\_bs.append(pixels\_list[i \* image.size[0] + j][2])

pixels\_array.append(image\_row)

Далее с помощью роевого алгоритма найдем инициирующие позиции кластеров и приступим к выполнению метода k-means с помощью библиотеки KMeans. Полученные кластеры будем выводить на трехмерную систему координат, где одинаковый цвет у точек будет означать принадлежность к одной группе (рисунок 12).

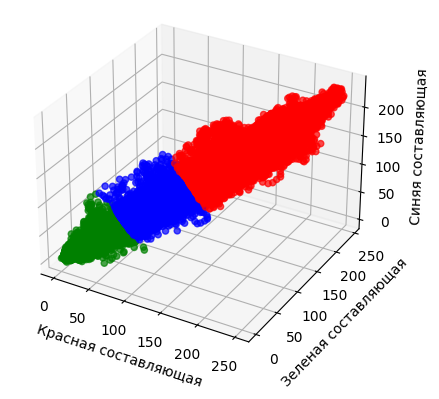


Рисунок 12. Пиксели изображения, окрашенные в цвет своего кластера.

Окрасим пиксели по их координатам так, чтобы визуально можно было отличить каждый кластер (рисунок 13).

new\_pixels\_array = []

new\_pixels = []

for i in range(image.size[1]):

temp\_pixel\_array = []

for j in range(image.size[0]):

pixel = pixels\_array[i][j]

min\_value = 0

current\_cluster = -1

# проверяем, к какому кластеру относится пиксель

for cluster\_num in range(k\_regions):

value = pow(pixel[0] - cluster\_centers[cluster\_num][0], 2) + \

pow(pixel[1] - cluster\_centers[cluster\_num][1], 2) + \

pow(pixel[2] - cluster\_centers[cluster\_num][2], 2)

if current\_cluster == -1:

current\_cluster = cluster\_num

min\_value = value

else:

if value < min\_value:

current\_cluster = cluster\_num

min\_value = value

new\_pixels.append(colors\_rgb[current\_cluster])

temp\_pixel\_array.append(colors\_rgb[current\_cluster])

new\_pixels\_array.append(temp\_pixel\_array)

После этого отсечем все лишние кластеры, оставив только кластер(-ы) с домами, перейдя в монохромное отображение (рисунок 14). После этого с помощью сканирующего окна в 50x50 пикселей будем проходить по всему снимку и подсчитывать процент «пикселей-домов» и окрашивать соответствующую зону в более красный цвет, чем больше там процент застройки (рисунок 15).

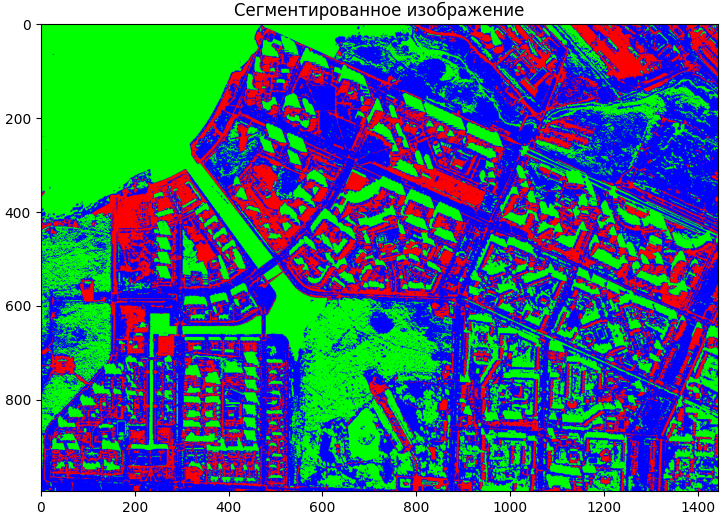


Рисунок 14. Сегментированное изображение с тремя кластерами (зеленый – вода и лес; синий – дороги и травяные покровы; красный – здания)

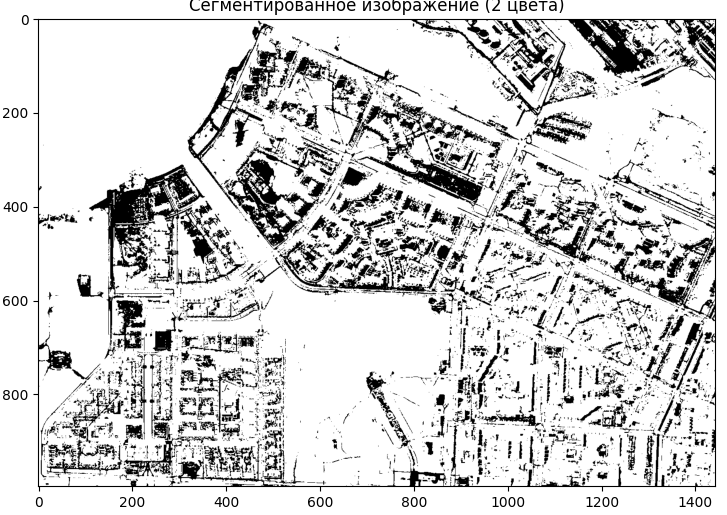


Рисунок 15. Сегментированное изображение с оставшимися домами.



Рисунок 16. Оригинальное изображение с маской плотности (более красное – более застроенное)

Далее реализуем разные операторы со сканирующим окном для уже подготовленного изображения (градации серого) и выведем его также в виде монохромного изображения (рисунок 17).



Рисунок 17. Контурный препарат, полученный с помощью оператора Превитта.

2.2. Тестирование.

1. Метод k-средних, 3 кластера.

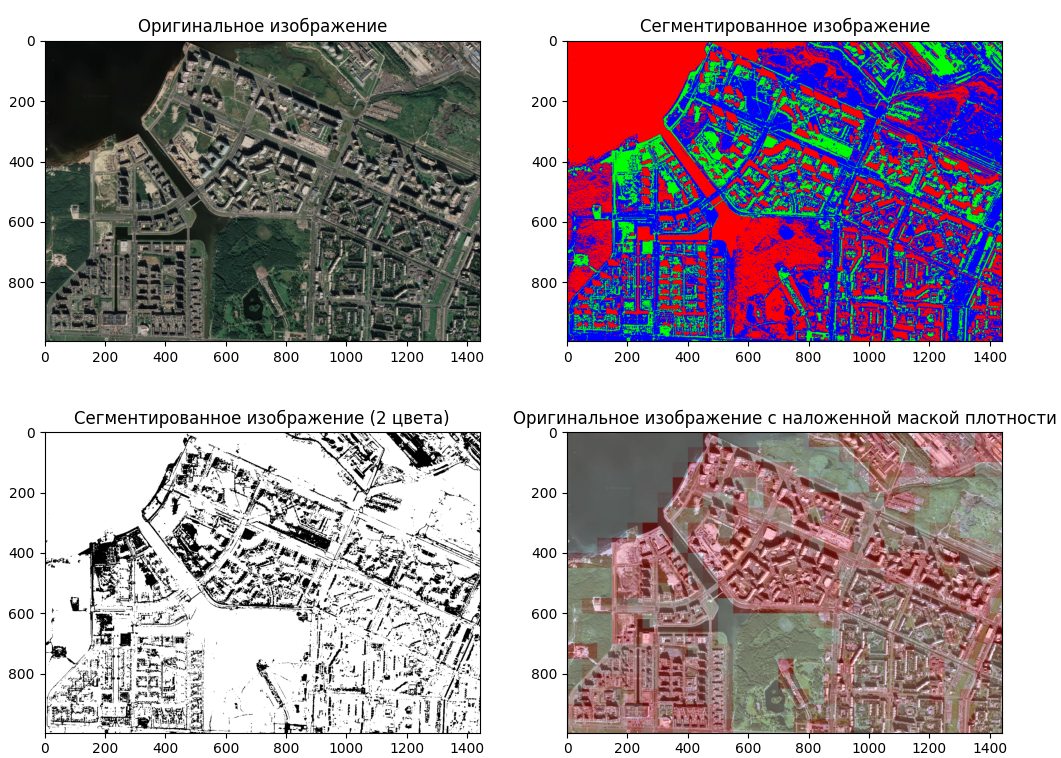


Рисунок 18. Результат выполнения сегментации методом k-средних для 3 кластеров.

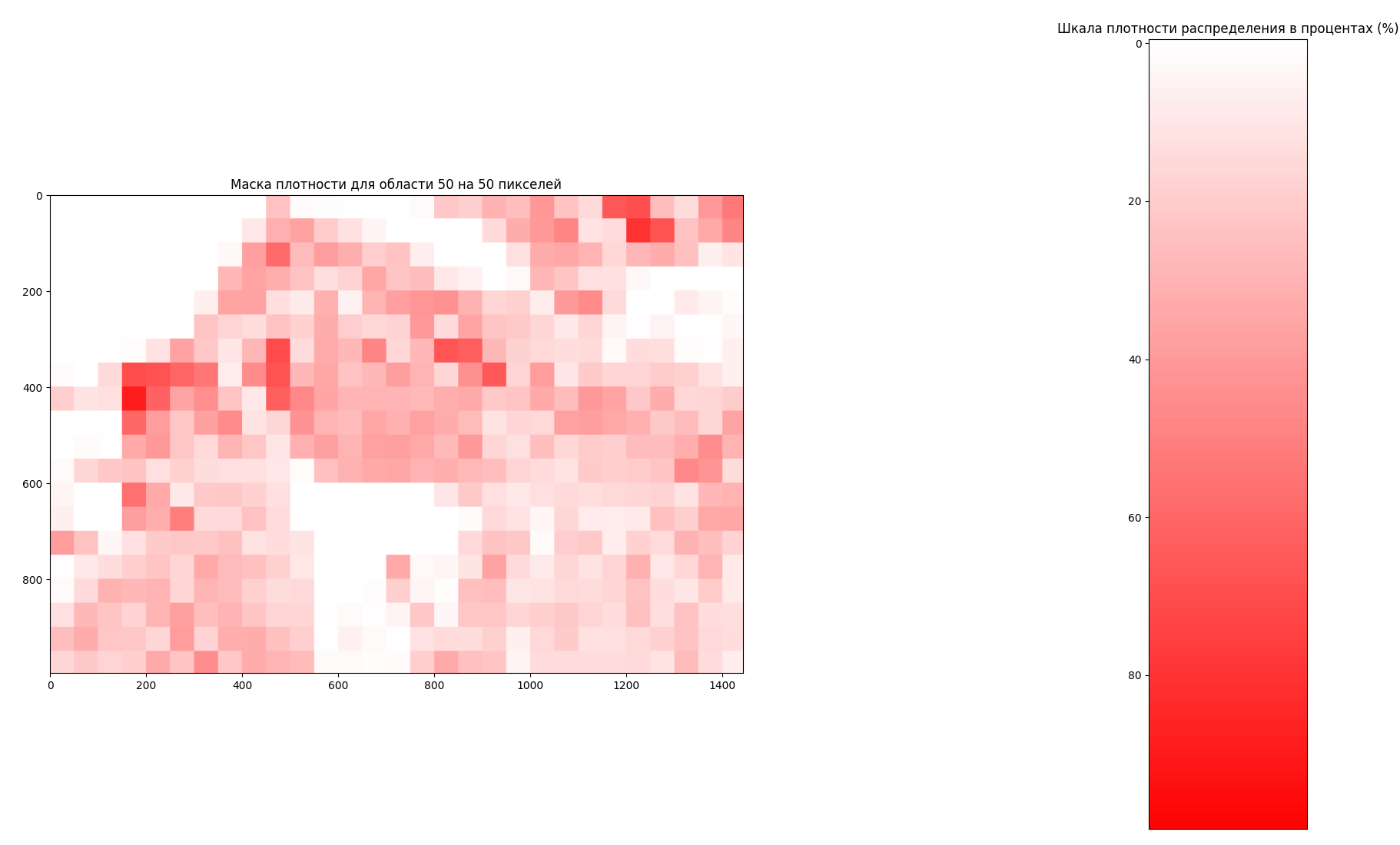


Рисунок 19. Маска плотности сегментации методом k-средних для 3 кластеров.

2. Метод k-средних, 6 кластеров

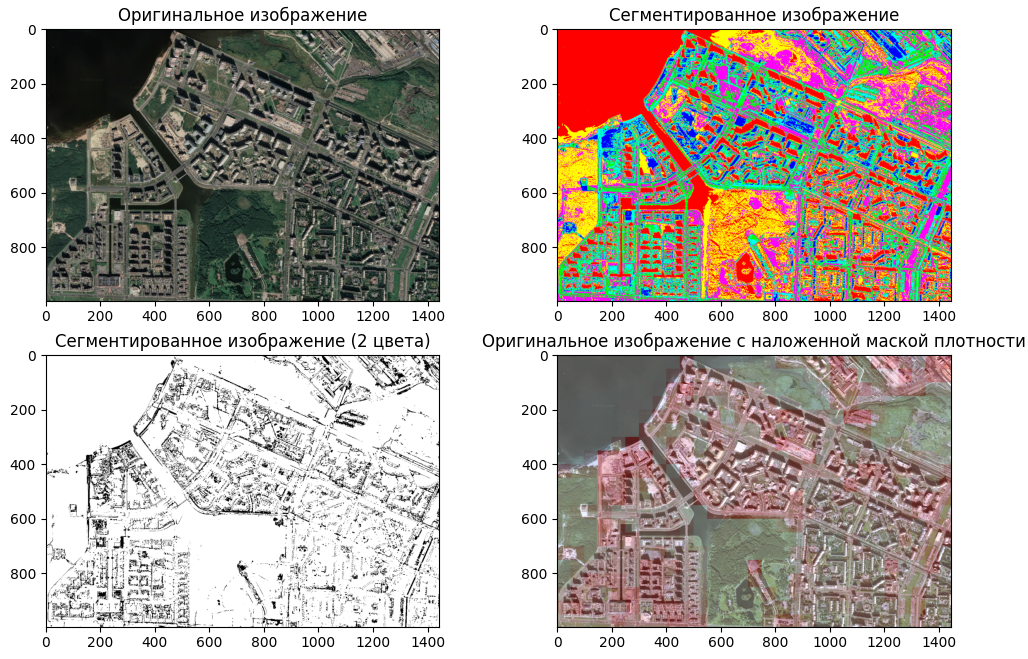


Рисунок 20. Результат выполнения сегментации методом k-средних для 3 кластеров.

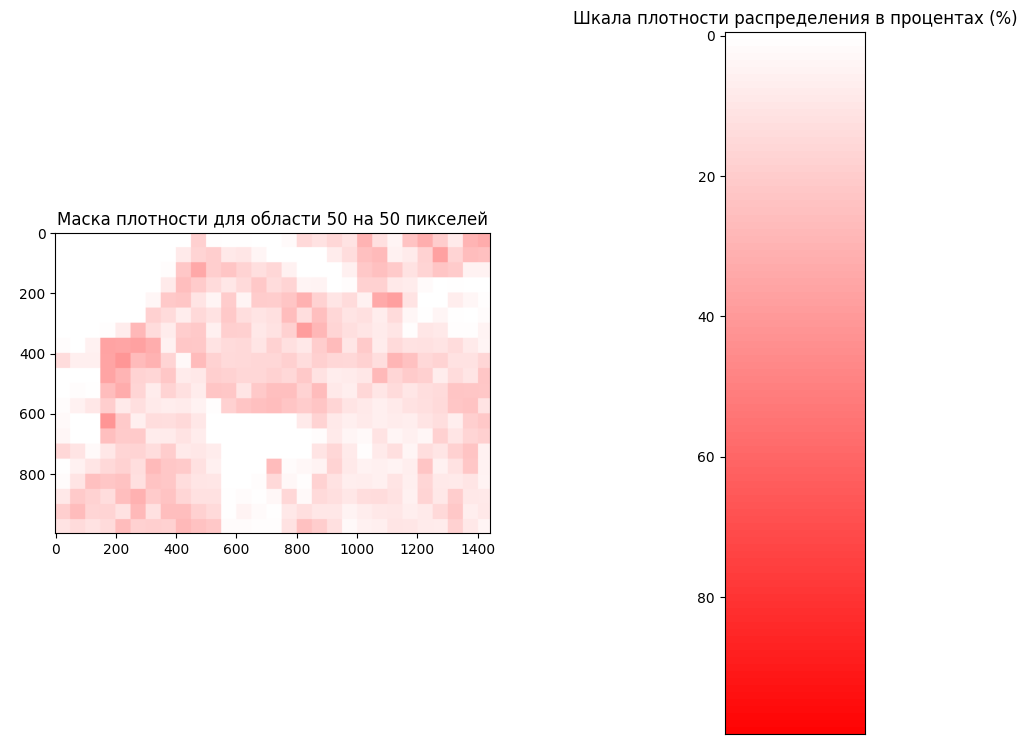


Рисунок 21. Маска плотности сегментации методом k-средних для 6 кластеров.

3. Метод выделения контуров. Оператор Превитта для порогового значения, равному 150.



Рисунок 22. Контуры зданий, полученных с помощью оператора Превитта с пороговым значением, равным 150.

4. Метод выделения контуров. Оператор Превитта для порогового значения, равному 150.



Рисунок 23. Контуры зданий, полученных с помощью оператора Превитта с пороговым значением, равным 200.

2.3. Выводы по главе 2

В данной главе была описана программа для сегментации спутниковых изображений, а также проведены тесты, которые помогли выяснить, что:

1. Методы k-средних и выделения контуров позволяют выделить наиболее четкие объекты на спутниковом снимке.

2. Чем больше количество кластеров или пороговое значение, тем меньше шума остается на результирующем изображении и тем больше вероятность отсечения какого-либо важного объекта.

3. Сам по себе метод выделения контуров объектов не может сегментировать снимки, его необходимо дорабатывать для заливки контуров домов.

4. Алгоритмы иногда выделяют дороги или побережья, считая, что это дома, что необходимо исправлять добавлением проверки на признаки прямых

5. Иногда тени могут закрывать собой другие дома, что, скорее всего может быть исправлено с помощью обработки изображения в цветовой модели HSL.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Биоэвристики: теория, алгоритмы и приложения : монография / С. И. Родзин, Ю. А. Скобцов, С. А. Эль-Хатиб. – Чебоксары: ИД «Среда», 2019. – 224 с.

2. Bing Xue, Mengjie Zhang, Will N. Browne, 2012 Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach,IEEE Trans. Cybern., 2168—2267

3. Long Tan, A Clustering K-means Algorithm Based on Improved PSO Algorithm,2015 Fifth International Conference on Communication Systems and Network Technologies

4. «ijs.uobaghdad.edu.iq» // [Электронный ресурс]. Режим доступа: https://ijs.uobaghdad.edu.iq/index.php/eijs/article/view/3825/1948 (дата обращения: 19.05.2023).

5. «hindawi.com» // [Электронный ресурс]. Режим доступа: https://www.hindawi.com/journals/mpe/2014/976486/ (дата обращения: 19.05.2023).

6. Венцов Н. Н. Обзор алгоритмов кластеризации, используемых в задачах поиска изображений по содержанию / Н. Н. Венцов, В. В. Долгов, Л. А. Подколзина // Инженерный вестник Дона. — 2016. — Т. 42. — № 3 (42)

7. «researchgate.net» // [Электронный ресурс]. Режим доступа: https://www.researchgate.net/publication/327122544\_Novel\_Image\_Segmentation\_Using\_Particle\_Swarm\_Optimization (дата обращения: 19.05.2023).

8. Р. Гонсалес, Р. Вудс «Цифровая обработка изображений» — М: Техносфера, 2005 – 1007с.

9. Е. Петров, Е. Медведева, Н. Харина, Е. Курбатова. «Методы и алгоритмы обработки цифровых изображений» // Инфокоммуникационные технологии. 2011.