



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Компьютерные системы и сети

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.04.01 Информатика и вычислительная техника

## **РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА** **К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ** **МАГИСТРА НА ТЕМУ:**

*Подсистема подготовки больших массивов  
изображений для системы геолокации объектов на  
основе их фотографий*

Студент

ИУ6-41М

(Группа)

\_\_\_\_\_  
(Подпись, дата)

В.Ю. Сидоров

\_\_\_\_\_  
(И.О. Фамилия)

Руководитель

\_\_\_\_\_  
(Подпись, дата)

Д.В. Березкин

\_\_\_\_\_  
(И.О. Фамилия)

Нормоконтролер

\_\_\_\_\_  
(Подпись, дата)

Ю.И. Бауман

\_\_\_\_\_  
(И.О. Фамилия)

2020 г.

## **ЗАДАНИЕ**

## **КАЛЕНДАРНЫЙ ПЛАН**

## АННОТАЦИЯ

Работа посвящена выработке подхода к созданию гибридной интеллектуальной системы (ГИС) определения местоположения объектов по их фотографиям. Одной из подсистем проектируемой ГИС является подсистема подготовки больших массивов фотографий с геотегами и разбиения земной поверхности на непересекающиеся сегменты.

Для этого был проведен анализ существующих способов разбиения земной поверхности на непересекающиеся множества, дан их краткий обзор и сравнительная характеристика; анализ источников фотографий, методы быстрого доступа к ним. Проанализирована целесообразность и эффективность использования указанных способов разбиения для решения задач, поставленных в рамках дипломного проекта. Проанализирована целесообразность и эффективность использования библиотеки S2 для решения задач, поставленных в рамках дипломного проекта.

Проведен анализ существующих подходов к геолокации фотографий, предложен алгоритм работы ГИС и состав входящих в нее подсистем.

Рассмотрены ключевые моменты проектирования и реализации системы. Осуществлена разработка технологии тестирования и приведены результаты её применения к разработанному прототипу.

**Ключевые слова:** гибридная интеллектуальная система, сегментирование, непересекающиеся сегменты, большие массивы изображений, Google S2, геопозиционирование, геотеги.

## ANNOTATION

The work is devoted to developing an approach to creating a hybrid intelligent system (GIS) for determining the location of objects from their photographs. One of the subsystems of the designed GIS is the subsystem for preparing large arrays of photographs with geotagging and dividing the earth's surface into disjoint segments.

For this, an analysis was made of the existing methods of dividing the earth's surface into disjoint sets, their brief overview and comparative characteristics were given; analysis of photo sources, methods of quick access to them. The feasibility and

effectiveness of using the specified methods of partitioning to solve the problems posed in the framework of the graduation project is analyzed. The feasibility and efficiency of using the S2 library for solving the tasks set in the framework of the graduation project is analyzed.

The analysis of existing approaches to geolocation of photographs is carried out, the algorithm of the GIS and the composition of its subsystems are proposed.

The key points of the design and implementation of the system are considered. The testing technology was developed and the results of its application to the developed prototype are presented.

Keywords: hybrid intelligent system, segmentation, disjoint segments, large arrays of images, Google S2, geo-positioning, geotags.

## РЕФЕРАТ

Записка 26 с., 3 табл., 9 рис., 7 ист.

### ГИБРИДНАЯ ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА, ГИБРИДНАЯ ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА, GEONASH, GOOGLE S2, КРИВАЯ ГИЛЬБЕРТА, HASH, YFCC100M, IM2GPS, WWSGID, СЕГМЕНТИРОВАНИЕ ЗЕМНОЙ ПОВЕРХНОСТИ, ПОЛИГОНЫ, GEOJSON

Объектом разработки является прототип подсистемы подготовки больших массивов изображений для системы геолокации объектов на основе их фотографий.

Успешная оценка геолокации фотографии может использоваться во многих интересных проектах, хоть это и является крайне сложной задачей. Поэтому разработка такой системы имеет актуальный характер.

Цель работы – проработка подхода к созданию подсистемы подготовки больших массивов изображений для системы геолокации объектов на основе их фотографий, с использованием данных, извлечённых из неструктурированной информации и обоснования получаемых результатов.

В процессе работы была изучена предметная область, выделены основные методы разбиения земной поверхности на непересекающиеся сегменты, привести краткую характеристику описываемых методов, выделены основные особенности; оценена оптимальность выбора того или иного метода для решения задач, поставленных в рамках дипломного проекта; выделены основные источники больших массивов фотографий с геотегами, приведена их краткая характеристика и обозначены основные особенности. Спроектирована обобщенная структурная схема разрабатываемой подсистемы. Реализованы основные модули подсистемы.

Результатом работы стала подсистема, реализующая весь список функций, описанных в техническом задании.

Проведено тестирование подсистемы в различных ситуациях. Результаты тестирования показывают соответствие программы требованиям технического задания.

Разработанная подсистема подготовки больших массивов изображений для системы геолокации объектов на основе их фотографий была использована в конкурсе ФПИ 2020 (конкурс на лучшую интеллектуальную технологию геопривязки изображений).

## СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....	13
ВВЕДЕНИЕ .....	14
1 Анализ предметной области .....	15
1.1 Актуальность исследований .....	15
1.2 Обзор существующих подходов к геолокации .....	16
1.3 Анализ способов получения больших массивов фотографий с геотегами ...	18
1.3.1 Описание поиска и выгрузки больших массивов фотографий с геотегами .....	18
1.3.2 Массив данных YFCC100M .....	19
1.3.3 Массив данных Im2GPS .....	20
1.3.4 Массив данных проекта World-Wide Scale Geotagged Image Dataset for Automatic Image Annotation and Reverse Geotagging .....	21
1.4 Анализ методов разбиения земной поверхности на непересекающиеся сегменты .....	24
1.4.1 GeoHash.....	24
1.4.1.1 Описание GeoHash.....	24
1.4.1.2 Принцип работы GeoHash.....	25
1.4.2 Google S2Geometry .....	27
1.4.2.1 Проекция .....	28
1.4.2.2 Иерархические клетки.....	29
1.4.2.3 Кривая Гильберта .....	30
1.4.2.4 Анализ быстрого способа доступа к фотографиям и их метаданным .....	32
1.4.2.5 Охватывающие регионы .....	35
1.4.2.6 Использование полигонов.....	37



1.4.3 Сравнительная характеристика описанных методов.....	39
1.5 Анализ сервисов для хранения больших массивов данных .....	42
1.5.1 Описание сервиса Dropbox .....	43
1.5.2 Описание сервиса AWS S3 .....	43
1.5.3 Сравнение времени доступа к данным, используя сервисы Dropbox и AWS S3.....	44
1.6 Выводы по исследовательской части .....	45
2 Проектирование и разработка подсистемы подготовки больших массивов фотографий для системы геолокации объектов на основе их фотографий .....	46
2.1 Структура системы геолокации объектов на основе их фотографий.....	46
2.2 Структура подсистемы подготовки больших массивов фотографий .....	48
2.3 Проектирование модуля подготовки больших массивов фотографий.....	50
2.3.1 Проектирование подмодуля поиска похожих фотографий для исходной выборки.....	51
2.3.2 Проектирование подмодуля выгрузки обновленной выборки на удаленное хранилище .....	54
2.4 Проектирование модуля разделения земной поверхности на непересекающиеся смежные сегменты.....	55
2.4.1 Проектирование подмодуля предварительного определения по координатам принадлежности фотографий к тому или иному сегменту .....	56
2.4.2 Проектирование подмодуля адаптивного сегментирования земной поверхности .....	58
2.5 Разработка диаграммы вариантов использования .....	60
2.6 Разработка диаграммы состояний интерфейса .....	68
2.7 Разработка прототипов форм графического интерфейса .....	70
2.8 Разработка диаграммы классов подсистемы .....	70

2.9 Выводы по конструкторской части .....	70
3 Тестирование спроектированной подсистемы .....	70
ЗАКЛЮЧЕНИЕ .....	71
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	73
ПРИЛОЖЕНИЕ А .....	77
ПРИЛОЖЕНИЕ Б .....	78
ПРИЛОЖЕНИЕ В .....	79
ПРИЛОЖЕНИЕ Г .....	80

## **ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ГИИС – Гибридная Интеллектуальная Информационная Система;

БД – База Данных;

GPS - Global Positioning System — система глобального позиционирования;

EXIF - Exchangeable Image File Format — стандарт, позволяющий добавлять к изображениям и прочим медиафайлам дополнительную информацию (метаданные), комментирующую этот файл, описывающий условия и способы его получения, авторство и т.п.;

API - application programming interface - программный интерфейс приложения, интерфейс прикладного программирования;

СУБД – Система управления базами данных;

## ВВЕДЕНИЕ

В последние годы интенсивно развиваются гибридные интеллектуальные системы (ГИС), позволяющие использовать преимущества традиционных средств и методов искусственного интеллекта, и, в то же время, преодолевать присущие им недостатки, способные решать задачи, которые до сих пор неудовлетворительно решались отдельными методами искусственного интеллекта. Гибридные интеллектуальные системы позволяют наиболее эффективно обрабатывать формализуемые и неформализуемые знания за счёт интеграции как традиционных методов обработки данных, так и искусственного интеллекта - экспертных и нечётких методов, искусственных нейронных сетей, генетических алгоритмов и т. д.

Определение местонахождения объектов по их фотографии требуется во многих жизненных ситуациях. Это чрезвычайно сложная задача, поскольку большинство фотографий содержат лишь несколько подсказок о местоположении, где они были сделаны.

Проектируемая ГИС должна оценивать, исходя из различных критериев, примерное местоположение сделанной фотографии. Фотографии могут быть классифицированы по области на земной поверхности, также имеет место быть неравномерное распределение плотности сделанных во всем мире фотографий. Поэтому оценивать примерное местоположение сделанных фотографий разумно с помощью некоторых областей на земной поверхности, в которых с некоторой вероятностью они могли быть сделаны.

Основную работу разрабатываемой подсистемы можно разделить на 3 этапа:

- 1) Подготовка обучающей и тестирующей выборки фотографий;
- 2) Проецирование фотографий на поверхность Земли;
- 3) Адаптивное сегментирование земной поверхности в зависимости от плотности распределения фотографий;

## **1 Анализ предметной области**

### **1.1 Актуальность исследований**

Во многих жизненных ситуациях требуется определение местонахождения объектов по их фотографиям. Если фотография сделана современным цифровым фотоаппаратом с встроенным GPS-датчиком, фиксирующим координаты объекта, то проблема решается достаточно просто. Указанная информация размещается в метаданных о фотографии, которые явно невидны, но могут быть получены с помощью специализированных программных средств.

Однако, наличие метаданных о местоположении объекта на фотографии (геолокации) зависит от нескольких факторов. Во-первых, от устройства, которым была сделана фотография. В некоторых камерах или мобильных устройствах может не быть GPS-датчика, фиксирующего координаты съемки. Во-вторых, от желания пользователей мобильных устройств: они могут отключить геолокацию из соображений приватности или уменьшения нагрузки на аккумулятор. В-третьих, наличие таких данных зависит от ресурса, на котором фотография была опубликована. Социальные сети Facebook, Twitter или Instagram удаляют метаданные фотографий во время их загрузки на серверы этих ресурсов. Проверить наличие метаданных довольно просто. Для этого можно загрузить файл фотографии и проверить его свойства. Для более быстрого и эффективного анализа метаданных фотографии, которые содержатся в EXIF-файле данных можно использовать специальные онлайн-ресурсы. Одним из таких ресурсов является Jeffrey's Exif Viewer [1]. Написанный и выложенный в открытый доступ американским программистом, этот сервис анализирует и показывает всю доступную информацию из метаданных.

Необходимо отметить, что во многих случаях метаданные о фотографии отсутствуют или искажены. В то же время определение местонахождения объекта по его фотографии необходимо для решения целого ряда задач и в первую очередь при анализе больших данных. При этом должно быть выполнено автоматическое определение местоположения объекта на фотографии, исходя

только из наличия самой фотографии и анализа её содержания. Проведенный предварительный анализ показал, что задача определения местонахождения объектов по их фотографиям является актуальной.

## **1.2 Обзор существующих подходов к геолокации**

В настоящее время интеллектуальные системы, как правило, не разрабатываются отдельно, а встраиваются в виде модулей в традиционные информационные системы для решения задач, связанных с интеллектуальной обработкой данных и знаний.

В работе [2] рассматривается вопрос об определении конкретного события на основе массива фотографий. Однако отмечается, что метаданные, в частности, содержащие геолокацию, зачастую могут быть неполны и недостоверны. Определение геолокации фотографии позволяет значительно упростить процесс кластеризации фотографий по событию, а также определить его точную геолокацию.

Есть ряд исследований, где используются фотографии с известной геолокацией для обучения математической модели, например, работа [3], в которой показано, что определение геолокации фотографии без метаданных может помочь расширить количество данных для последующего анализа, что позволяет повысить его качество.

Корпорацией Google было заявлено, что ее поисковик в скором времени сможет определять место, в котором была сделана та или иная фотография. Для этого сотрудники компании создали нейронную сеть, получившую название PlaNet, в базу данных которой было загружено более 90 миллионов снимков с указанием мест, в которых они были сняты. Во время совершения поиска места PlaNet будет сравнивать снимок с изображениями из базы данных на пиксельном уровне. В случае совпадения она выдаст предположение о том, где была сделана фотография. Основные принципы распознавания изложены в статье [4]. Следует отметить, что пока что система далека от идеала. Во время первых тестирований

программа смогла правильно угадать страну лишь в 28,4% случаев, а континент - в 48%. Внешний вид интерфейса системы представлен на рисунке 1.



Рисунок 1 – Внешний вид интерфейса системы PlaNet

Корпорация Google объявила о том, что приложение Google Maps для мобильных устройств научилось самостоятельно вычислять вероятный пункт назначения в режиме навигации. Для определения наиболее вероятного пункта назначения функция Driving Mode анализирует историю предыдущих маршрутов и историю поисковых запросов. После того, как приложение определит вероятное место прибытия, навигатор будет выдавать стандартную в таких случаях информацию о времени в пути, плотности потока и возможных маршрутах. Имеется соответствующий интерфейс программирования приложений (API) [5].

В работе [31] представлен проект Im2GPS, в котором оценка геопозиции происходит исходя из сравнения исходной фотографии с 6 миллионами подготовленных снимков из Flickr, для которых заданы координаты GPS. Сравнительные результаты оценки геопозиции проектов PlaNet и Im2GPS описаны в [4, 30].

### **1.3 Анализ способов получения больших массивов фотографий с геотегами**

Поскольку для обучения нейронных сетей, входящих в состав проектируемой ГИС, потребуется огромное количество подготовленных фотографий с геотегами, измеряемое миллионами, появляется задача о том, где взять такое количество фотографий и как их хранить, чтобы можно было оперативно выбрать ту или иную фотографию из подготовленной выборки. Далее будут рассмотрены способы загрузки больших массивов фотографий с геотегами и существующие датасеты.

#### **1.3.1 Описание поиска и выгрузки больших массивов фотографий с геотегами**

В качестве основных источников фотографий выступают различные сервисы вроде VK, Facebook, Flickr, Instagram. Большой проблемой является то, что большинство сервисов ставят ограничение на количество выгружаемых фотографий. К тому же, сервисы VK, Facebook и Instagram удаляют из фотографий EXIF-данные, в которых содержится полезная информация, в том числе, координаты геопозиции.

И только лишь Flickr предоставляет доступ к выгрузке фотографий с геотегами, правда ограниченный – 2000 фотографий в сутки для обычного пользователя ресурса. Для этого необходимо получить доступ к API Flickr-а и выкачивать фото, составляя специальный поисковый запрос к серверу. Подробная информация, как получить доступ к API и формировать поисковые запросы, представлена в [6].

Такой принцип получения больших массивов фотографий является трудоемким и неоптимальным по затраченному времени, поэтому было решено использовать готовые датасеты фотографий с возможностью дополнения их, используя исходные коды запросов из [7].



### 1.3.2 Массив данных YFCC100M

Yahoo Flickr Creative Commons 100 Mission Dataset [8]– набор данных , который является крупнейшей общедоступной мультимедийной коллекцией, которая когда-либо была выпущена, и включает в себя 100 миллионов медиа-объектов, из которых примерно 99,2 миллиона - это фотографии, а 0,8 миллиона - видео, и все они были загружены на Flickr в период с 2004 по 2014 год и опубликованы. по коммерческой или некоммерческой лицензии.

В настоящее время набор данных распространяется через Amazon AWS в виде сжатого архива объемом 12,5 ГБ, содержащего только метаданные. Подробная информация о том, как скачать архив метаданных, представлена в [9].

Фактически фотографии и видео в настоящее время загружаются в облако, чтобы гарантировать, что набор данных останется доступным и неповрежденным в течение многих лет; Как и метаданные, фото и видео данные могут быть подключены как сетевой диск только для чтения.

В наборе данных содержится 48 366 323 фотографии и 103 506 видео, которые были аннотированы географической координатой, либо вручную пользователем, либо автоматически через GPS.

Таким образом выходит, что, используя датасет YFCC100M, можно получить 48 366 323 «грязных» фотографий с геотегами. Набор данных содержит неоднозначные фотографии, например, внутренней среды, еды и людей, для которых местоположение трудно предсказать. Пример пригодной для обучения нейросетей и не представляющей практической ценности фотографий представлен на рисунке 2.



Рисунок 2 – Пример фотографий с геотегами из датасета YFCC100M

Представленный массив фотографий содержит множество различных категорий фотографий. Топ-25 категорий по количеству фотографий представлен на рисунке 3.

<i>Concept</i>	<i>Photos</i>
outdoor	44,290,738
indoor	14,013,888
people	11,326,711
nature	9,905,587
architecture	6,062,789
landscape	5,121,604
monochrome	4,477,368
sport	4,354,325
building	4,174,579
vehicle	3,869,095
plant	3,591,128
blackandwhite	2,585,474
animal	2,317,462
groupshot	2,271,390
sky	2,232,121
water	2,089,110
text	2,074,831
road	1,796,742
blue	1,658,929
tree	1,641,696
hill	1,448,925
shore	1,439,950
car	1,441,876
head	1,386,667
art	1,391,386

Рисунок 3 – ТОП-25 категорий фотографий из датасета YFCC100M

### 1.3.3 Массив данных Im2GPS

Im2GPS Dataset [7] –база данных из 6 472 304 фотографий, загруженных из Flickr. Список географических ключевых слов включает в себя каждую страну, каждый континент, топ 200 густонаселенных городов и популярные туристические места (например, «Пиза», «Никко», «Орландо»). В составе

датасета отсутствуют фотографии, которые также были помечены ключевыми словами, такими как «день рождения», «концерт», «аннотация» «кот» и «собака».

Все изображения были уменьшены до максимального размера 1024 ширины или высоты и хранятся в формате JPEG. В общей сложности датасет представляет собой 1 терабайт данных.

Распределение фотографий из указанного датасета на земной поверхности представлено на рисунке 4.

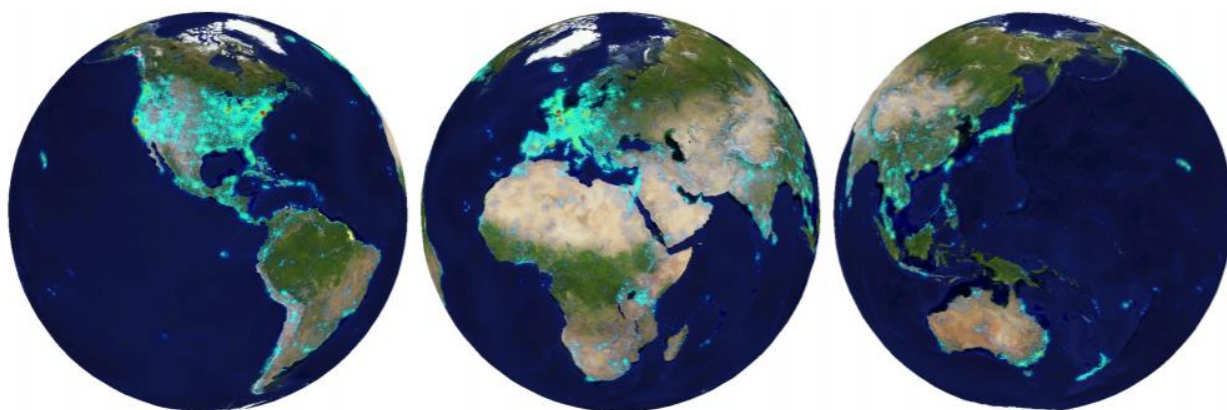


Рисунок 4 – Распределение фотографий по земной поверхности из датасета Im2GPS

В проекте IM2GPS [7], который создан для распознавания геопозиции по фотографии, также используется Flickr в качестве источника наборов фотографий. В [7] также предоставлены исходные коды для удобной загрузки фотографий с сервиса Flickr по категориям.

Изображения скачиваются напрямую с http-сервера Flickr вместо того, чтобы проходить через API, и, следовательно, не требует ключа API и нет ограничений на загрузку фотографий.

#### **1.3.4 Массив данных проекта World-Wide Scale Geotagged Image Dataset for Automatic Image Annotation and Reverse Geotagging**

Данный набор данных находится в открытом доступе и состоит из 14,1 миллиона изображений с соответствующими метаданными., загруженных из

Flickr [10]. Выборка содержит фотографии со всего мира. основополагающий принцип выбора фотографий данного проекта - стратегии сканирования данных, основанной на графе дружбы Flickr и принципе малого мира [11]. Предложенный метод основан на [12]. Однако вместо создания случайной выборки идентификаторов фотографий генерируется выборка идентификаторов, соответствующих пользователям, проживающим в разных местах мира. Начиная с начального набора пространственно хорошо распределенных пользователей, просматриваем связанные с ними графы дружбы, чтобы расширить набор пользователей. Согласно принципу маленького мира, окончательный набор пользователей будет содержать пользователей, которые сделали фотографии, покрывающие всю карту мира и с реалистичным распределением плотности.

Таким образом получается выборка фотографий по наиболее популярным городам мира (Рис. 5), с равномерным распределением фотографий, наподобие распределению людей на планете (Рис.6)

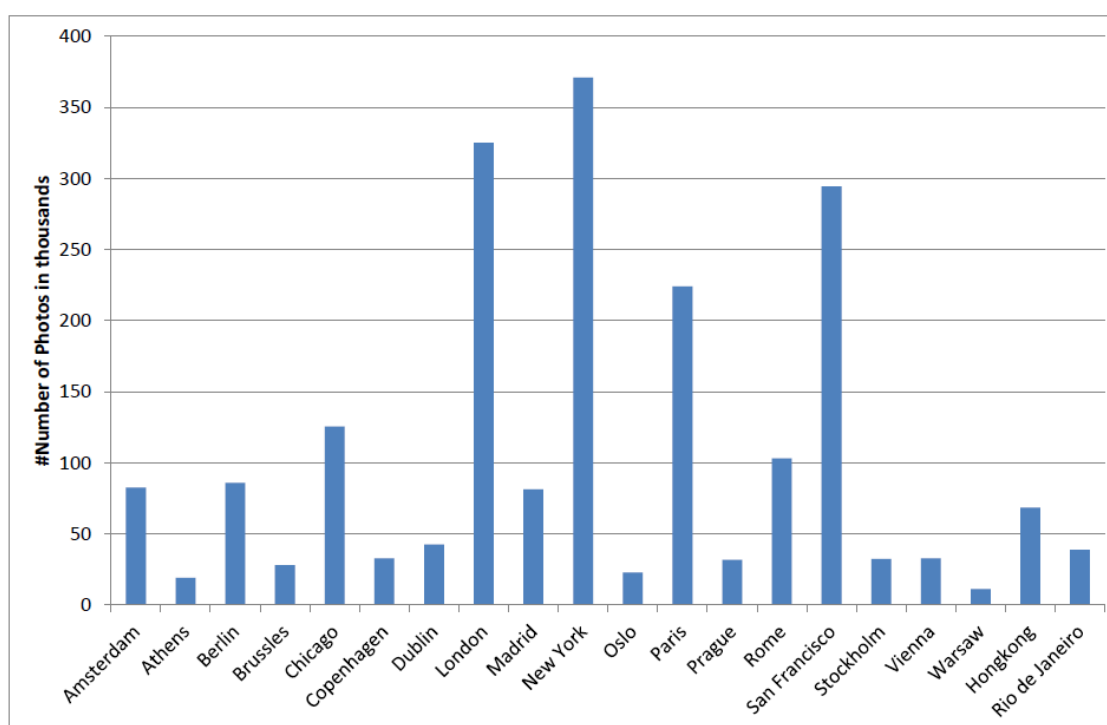


Рисунок 5 – Распределение фотографий по городам из датасета WWSGI

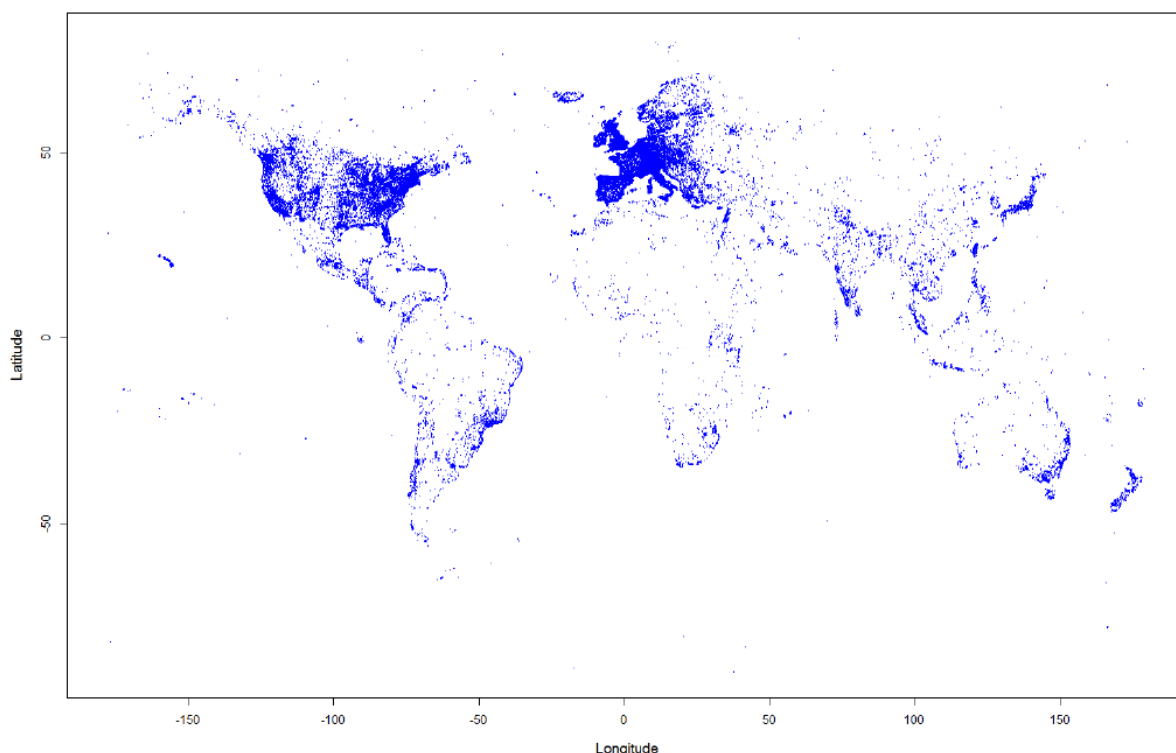


Рисунок 6 – Распределение фотографий по земной поверхности из датасета WWSGI

Данный массив данных хранится на сервере, а доступ к фотографиям предоставляется при помощи архива метаданных фотографий. Он представляет собой CSV-файл, колонки которого описаны на рисунке 7.

Field Name	Description
id	Photo Flickr identifier
userid	Flickr user identifier
tags	Image tags
latitude	Latitude of image location
longitude	Longitude of image location
views	Number of image views
date_taken	Date of image capture
date_uploaded	Date of image upload to Flickr
Accuracy*	Flickr accuracy level of the geographical coordinate
flickr_secret**	Flickr secret
flickr_server**	Flickr server name
flickr_farm**	Flickr farm name

Рисунок 7 – Формат CSV-файла метаданных о фотографиях датасета WWSGI

Формат запроса для получения доступа к фотографиям:

[http://farm{flickr\\_farm}.staticflickr.com/{flickr\\_server}/{id}\\_{flickr\\_secret}.jpg](http://farm{flickr_farm}.staticflickr.com/{flickr_server}/{id}_{flickr_secret}.jpg)

## **1.4 Анализ методов разбиения земной поверхности на непересекающиеся сегменты**

### **1.4.1 GeoHash**

Geohash - это общедоступная система геокодирования, изобретенная Густавом Нимейером, которая кодирует географическое местоположение в виде короткой цепочки букв и цифр. Это иерархическая пространственная структура данных, которая подразделяет пространство на сегменты в форме сетки, что является одним из многих применений так называемой кривой Z-порядка и, как правило, кривых заполнения пространствах [13].

#### **1.4.1.1 Описание GeoHash**

GeoHash предлагает свойства, такие как произвольная точность и возможность постепенного удаления символов из конца кода, чтобы уменьшить его размер (и постепенно потерять точность). Как следствие постепенного снижения точности, близлежащие места часто (но не всегда) представляют аналогичные префиксы. Чем длиннее общий префикс, тем ближе эти два места.

Основные применения геохешей:

- Как уникальный идентификатор.
- Для представления точечных данных, например, в базах данных.
- Геохеш также было предложено использовать для геотеггинга.

При использовании в базе данных структура данных с географической привязкой имеет два преимущества. Во-первых, данные, проиндексированные с помощью геохеша, будут иметь все точки для данной прямоугольной области в смежных срезах (количество срезов зависит от требуемой точности и наличия «линий разломов» геохеша). Это особенно полезно в системах баз данных, где запросы к одному индексу намного проще или быстрее, чем запросы с несколькими индексами. Во-вторых, эту структуру индекса можно использовать

для быстрого и «грязного» поиска близости: самые близкие точки часто находятся среди самых близких геохешей.

#### 1.4.1.2 Принцип работы GeoHash

1. У нас есть местоположение на карте мира (см. Рис.8), и мы хотим представить эту точку как значение Geohash.

Сначала мы делим мир на две половины с вертикальной линией и присваиваем каждой половине двоичное значение 0 или 1. Наша точка на карте находится в половине «0».



Рисунок 8 – Начальное разбиение карты мира при алгоритме Geohash



2. Затем каждая подобласть делится напополам снова и снова, прибавляя к нашему хэшу новые значения (см. Рис. 9)



Рисунок 9 – Дальнейшие шаги получения хэша необходимой точки по алгоритму GeoHash

Таким образом, мы можем продолжать разделять пространство до тех пор, пока не достигнем уровня улицы или выше. В этот момент наш Geohash будет немного длиннее и будет выглядеть так:

00101 10101 01110 00110 00110 00110 11110 00111

Четные биты принимаются для кода долготы, нечетные – для кода широты соответственно.

Этот двоичный файл может быть представлен в виде буквенно-цифровых символов (32-битное кодирование). Каждые 5 бит преобразуются в один символ:

5 p f 6 6 6 y 7

Более подробное описание алгоритма представлено в [14].

Таким образом, тем больше количество символов в геохэше, тем более маленькую область он обозначает. Количество символов в геохэше называется



уровнем. Соответствие уровня и размера обозначаемой им территории представлено в Таблице 1.

Таблица 1 - Соответствие уровня Geohash и размера обозначаемой им территории

Уровень GeoHash	Размер ячейки, ширина	Размер ячейки, длина
1	5,000km	5,000km
2	1,250km	1,250km
3	156km	156km
4	39.1km	39.1km
5	4.89km	4.89km
6	1.22km	1.22km
7	153m	153m
8	38.2m	38.2m
9	4.77m	4.77m
10	1.19m	1.19m
11	149mm	149mm
12	37.2mm	37.2mm

Система Geohash была разработана в 2008 году, находится в открытом доступе и на данный момент реализована на таких языках программирования, как Java, Python, C++, Ruby, JavaScript.

#### 1.4.2 Google S2Geometry

Google S2Geometry – библиотека, которая определяет структуру для разложения единичной сферы на иерархию ячеек [15]. Каждая ячейка является четырехугольником, ограниченным четырьмя геодезическими. Верхний уровень иерархии получается путем проецирования шести граней куба на единичную сферу, а нижние уровни - путем рекурсивного деления каждой ячейки на четыре дочерних элемента. Например, на следующем рисунке показаны две из шести лицевых ячеек, одна из которых была подразделена несколько раз:

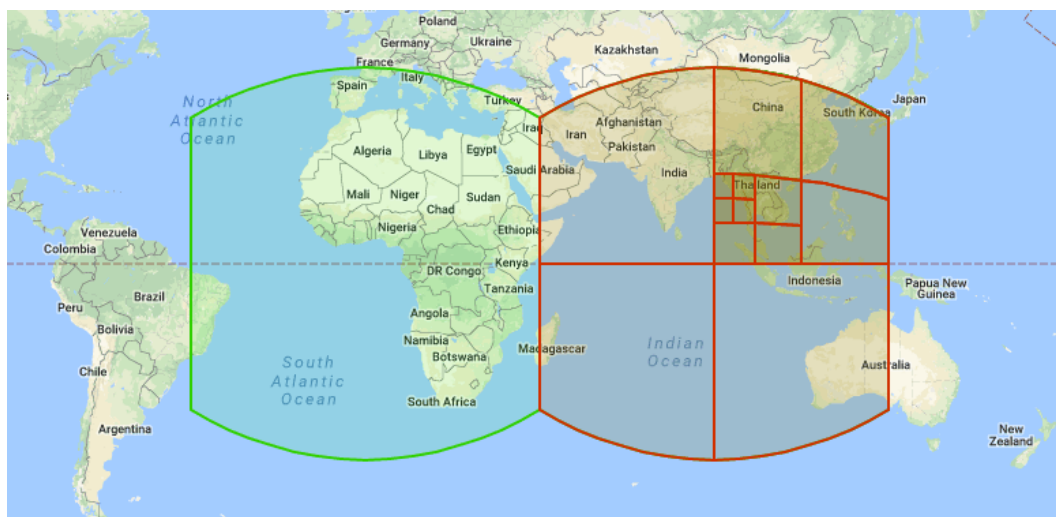


Рисунок 10 – Пример ячеек Google S2

### 1.4.2.1 Проекция

Обычно география подразумевает использование одной из распространённых проекций земного шара на плоскость. Например, всем нам известной проекции Меркатора [17]. Недостаток такого подхода заключается в том, что любая проекция так или иначе имеет искажения. Наш земной шар не выглядит как плоскость. Пример искажения проекции Меркатора представлен на рисунке 11. На картах Россия кажется огромной, но на самом деле площадь Африки в 2 раза больше площади России [16].



Рисунок 11 – Пример искажения проекции Меркатора

S2 решает эту проблему использованием исключительно сферической проекции и сферической геометрии. Конечно, Земля — тоже не совсем сфера, но искажениями в случае сферы можно пренебречь для большинства задач.

#### 1.4.2.2 Иерархические клетки

Второй (и самой важной) особенностью библиотеки является понятие иерархического разбиения земного шара на клетки (по-английски — cells).

Разбиение иерархично, то есть присутствует такое понятие, как уровень (или level) клетки. И на одном уровне клетки имеют примерно одинаковый размер.

Клетками можно задать любую точку на Земле. Если воспользоваться клеткой максимального, 30-го, уровня, которая имеет размер меньше сантиметра по ширине, то точность, как вы понимаете, будет очень высокой. Клетками более низкого уровня можно задать ту же самую точку, но точность уже будет меньше. Например, 5 м на 5 м.

Примеры размеров клеток представлены в Таблице 2.

Таблица 2 – Параметры клеток библиотеки S2Geometry

Уровень	Мин. площадь	Макс. площадь	Единицы измерения	Кол-во ячеек на уровне
0	85011012.19	85011012.19	km <sup>2</sup>	6
1	21252753.05	21252753.05	km <sup>2</sup>	24
12	3.04	6.38	km <sup>2</sup>	100M
30	0.44	0.93	cm <sup>2</sup>	7e18

Параметры всех ячеек представлены в [18].

Иерархия S2 полезна для пространственной индексации и для аппроксимации областей в виде набора ячеек. Клетки можно использовать для

представления точек и областей: точки обычно представлены в виде листовых ячеек, а области представлены в виде наборов ячеек на любом уровне (уровнях). Например, примерное представление Гавайев, как набора из 22 ячеек:

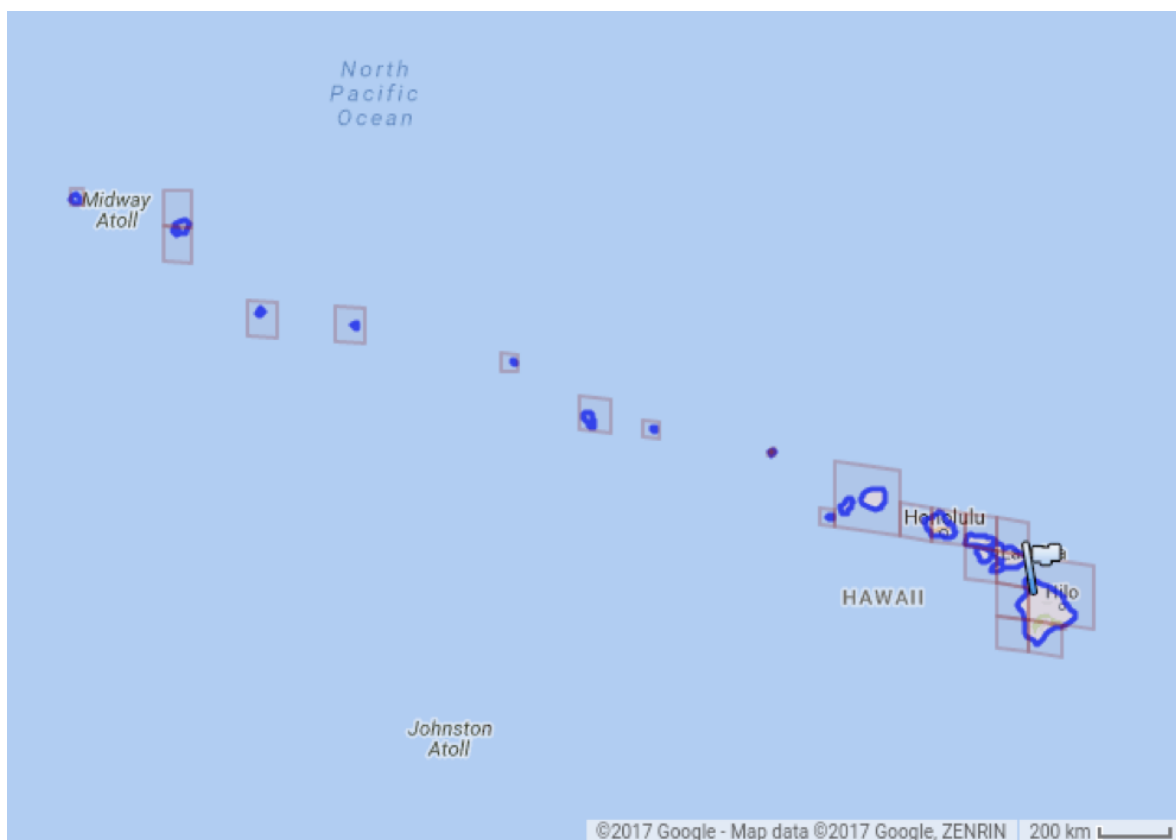


Рисунок 12 – Представление Гавайев как набора из 22 ячеек разного уровня

#### 1.4.2.3 Кривая Гильберта

Иерархическое разбиение сферы на уровни (см. Табл. 2) осуществляется с использованием кривой Гильберта.

Сначала мы заключаем сферу в куб и проецируем её на все шесть его граней, чуть-чуть подправив проекцию на ходу, чтобы убрать искажения. Это level 0 (см. Рис. 13). Затем мы каждую из шести граней куба можем разбить на четыре равные части — это level 1. И каждую из получившихся четырёх частей можно разбить ещё на четыре равные части — level 2. И так далее до level 30.

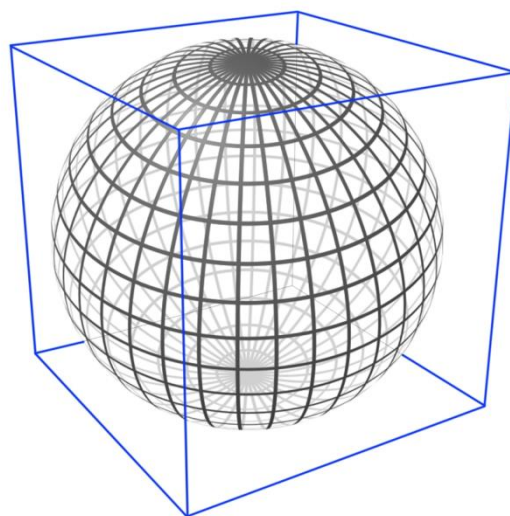


Рисунок 13 – Заключение сферы в куб

Но на данном этапе у нас всё ещё присутствует двумерность. И здесь на подмогу нам приходит математическая идея из далёкого прошлого. В конце XIX века Дейвид Гильберт придумал способ заполнения любого пространства одномерной линией, которая поворачивается, сворачивается и таким образом заполняет пространство [19]. Гильбертова кривая рекурсивна, а это значит, что точность или плотность заполнения можно выбирать. Любой мелкий кусок мы можем при необходимости заполнить плотнее (см. Рисунок 14).

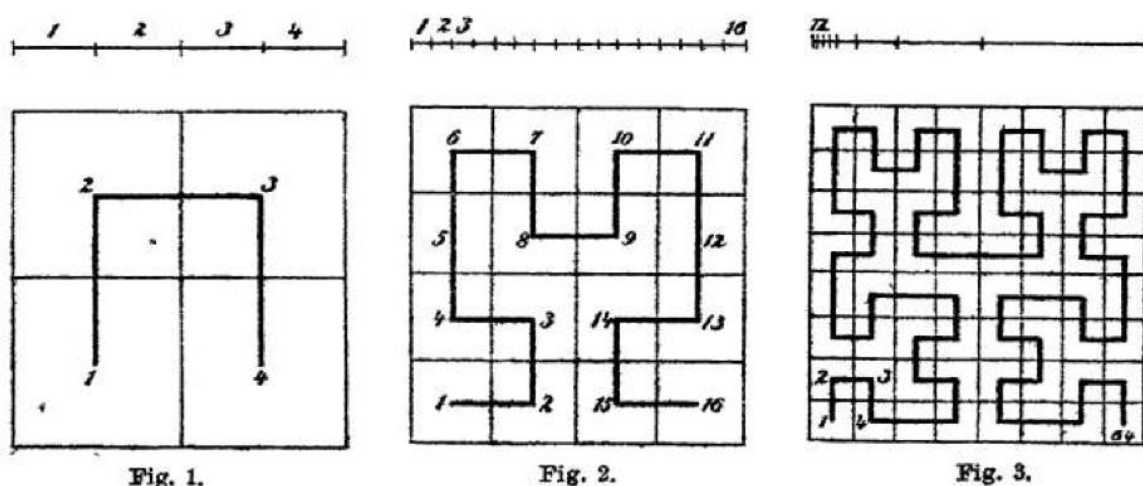


Рисунок 14 – Принцип работы Гильбертовой кривой

Мы можем заполнить такой кривой наше двумерное пространство. И если теперь взять эту кривую и растянуть её в прямую, то получится одномерный

объект, описывающий многомерный объект, — одномерный объект или координату на этой линии, с которой удобно и просто работать.

Примерно так выглядит заполнение Гильбертовой кривой Земли на пятом уровне подразделения S2Geometry:

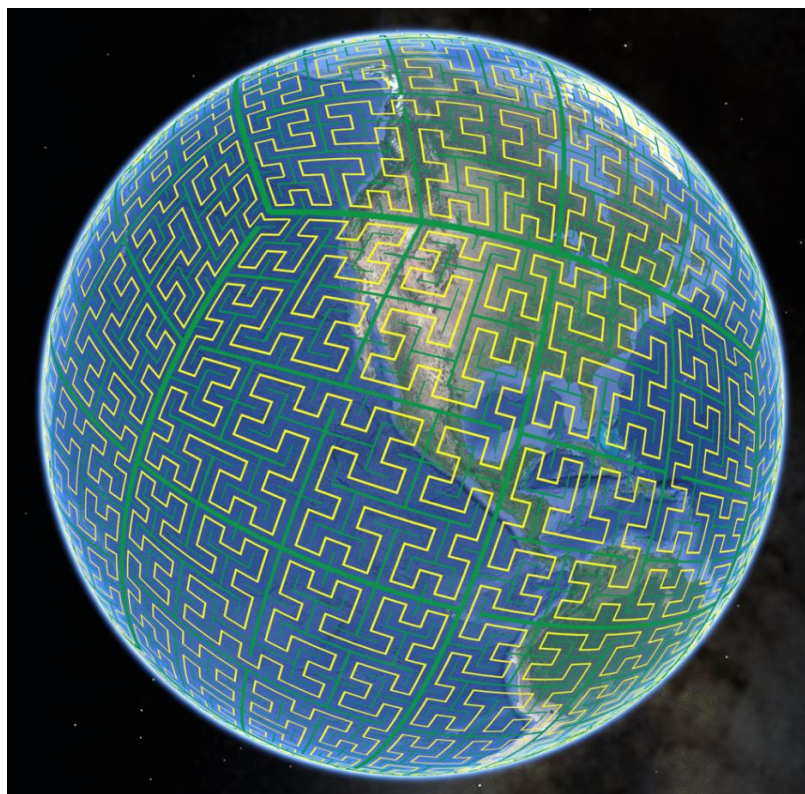


Рисунок 15 - заполнение Гильбертовой кривой Земли на пятом уровне подразделения S2Geometry

Если присмотреться к бинарному представлению значения индекса вдоль кривой Гильберта, связанной с каждой ячейкой квадродерева, то для первого порядка получим: 00, 01, 10 и 11. Для второго порядка первый четыре ячейки от 0 до 3 могут быть представлены как 0000, 0001, 0010, 0011. Следующие четыре ячейки с 4 по 7 имеют двоичное представление 0100, 0101, 0110, 0111 и так далее.

#### **1.4.2.4 Анализ быстрого способа доступа к фотографиям и их метаданным**

На основании предыдущих пунктов выделим важные параметры, рассматривая которые можно быстро определить всю существенную информацию о фотографии:

- 1) Путь к фотографии на жестком хранилище или удаленном сервере;



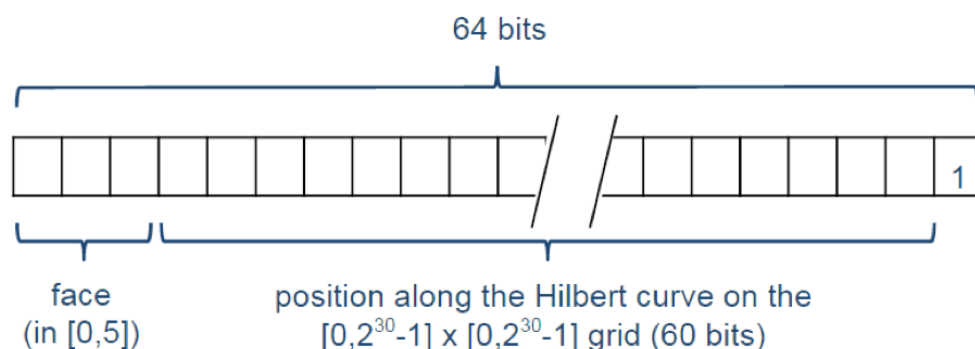
- 2) Идентификатор ячейки (см. рис. 16);
- 3) Координаты;
- 4) Страна (страны), которой(ым) принадлежит ячейка;

Каждая клетка может быть закодирована 64-битным числом (CellID). 3 первых бита используются для кодирования 6 граней куба.

Затем мы каждую из шести граней разбиваем на четыре равные части 30 раз. Чтобы задать одну из четырёх частей, на которые мы разбиваем клетку на каждом из уровней, нам хватит двух бит.

Итого  $3 + 2 \cdot 30 = 63$ . А один лишний бит мы выставим в конце, чтобы знать и быстро понимать, какого уровня клетка задаётся данным CellID (см. Рисунок 16).

#### S2 Cell ID of a **leaf** cell (level 30):



#### S2 Cell ID of a **level-2** cell:

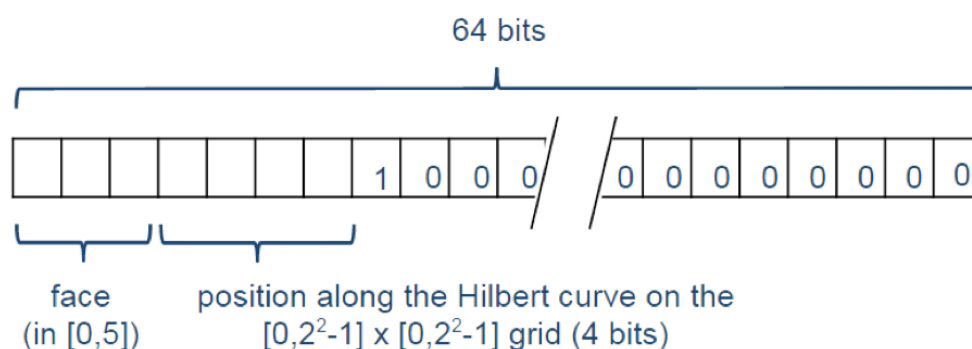


Рисунок 16 – Представление положения точки на сфере в виде 64-битного числа

Можно заметить, что двоичное представление значения положения вдоль кривой Гильберта второго порядка, связанной с дочерней ячейкой квадродерева

на уровне 2, имеет те же 2 старших бита, что и значение положения вдоль кривой Гильберта первого порядка его родительская ячейка на уровне 1.

Это верно на всех уровнях. Если взять двоичное представление ячейки  $N$  на кривой Гильберта порядка  $i$ , 4 дочерних элемента  $N$  на кривой Гильберта порядка  $o + 1$  будут иметь одинаковые старшие биты, которые равны  $N$ , до последних двух, которые определит ячейку однозначно.

Например, возьмем двоичное представление для случайной ячейки  $N$  на кривой порядка  $o$ :  $H_o(N) = 101011001... 1011$ . Четыре дочерних элемента  $n_0, n_1, n_2, n_3$  из  $H_o(N)$  порядка  $o + 1$  будут иметь представления:

$$H_{o+1}(n_0) = H_o(N) \cdot 00,$$

$$H_{o+1}(n_1) = H_o(N) \cdot 01,$$

$$H_{o+1}(n_2) = H_o(N) \cdot 11,$$

$$H_{o+1}(n_3) = H_o(N) \cdot 11,$$

где  $\cdot$  представляет конкатенацию.

Таким образом, получаем программируемое представление Гильбертовой кривой в виде 64-х битной переменной, которое можно использовать для эффективного индексирования и запроса элементов дерева квадрантов. Действительно, для каждой ячейки  $s_2$  мы можем построить ветвь дерева, начинающуюся с корневого узла, который следует двоичному представлению ячейки вдоль кривой Гильберта. Для списка ячеек мы можем объединить эти представления в одну древовидную структуру, к которой можно обращаться, используя только двоичные операции.

В узлах В-дерева мы будем хранить ячейку уровня `storageLevel` и список фотографий в данной ячейке.

Поскольку основная часть операций по отношению к набору фотографий – операции поиска, то использовать преимущества В-дерева будет целесообразно, т.к. сложность поиска (в среднем) составляет  $O(\log n)$ . Следовательно, чем больше элементов, тем лучше показывает себя данная структура.



Что нам дают все эти разбиения и кодирование Гильбертовой кривой в одно число?

1. Мы можем любую точку на земном шаре закодировать одним CellID. В зависимости от уровня мы получаем разную точность.
2. Любой двумерный объект на земном шаре мы можем закодировать одним или несколькими CellID. Аналогично мы можем играть с точностью, варьируя уровни.
3. Свойство Гильбертовой кривой, заключающееся в том, что точки, которые находятся рядом на ней, находятся рядом и в пространстве, и тот факт что CellID у нас — просто число, позволяют нам для поиска пользоваться обычным деревом типа B-дерева. В зависимости от того, что мы ищем (точку или область), мы будем пользоваться точным поиском, или range scan, то есть поиском «от» и «до».

#### **1.4.2.5 Охватывающие регионы**

Если необходимо сгенерировать ячейки, чтобы покрыть область, можно использовать метод библиотеки, в котором указывается максимальное количество ячеек, максимальный уровень ячеек и минимальный уровень ячеек, которые будут использоваться, а затем алгоритм будет приближать этот регион, используя указанные параметры. В приведенном ниже примере используется библиотека S2 с использованием ячеек 15-го уровня:

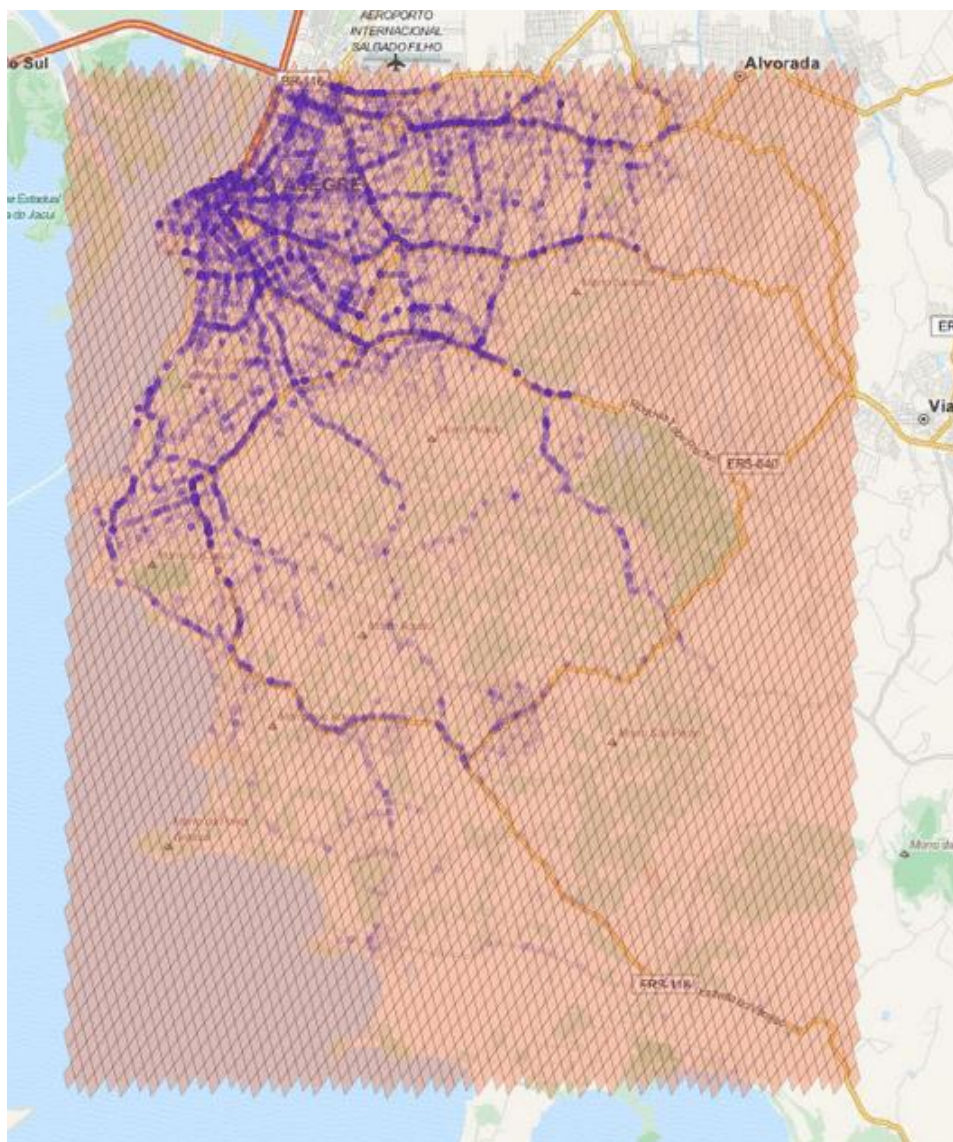


Рисунок 17 – Покрытие области ячейками 15 уровня, используя библиотеку S2 Geometry [21]

Библиотека S2 Geometry позволяет задавать минимальный и максимальный уровень ячеек. Если задать минимальный уровень, например, значением 8 (таким образом, предоставляя возможность использования более крупных ячеек), алгоритм будет аппроксимировать область таким образом, чтобы обеспечить меньшее количество ячеек, а также пытаться сохранить точность приближения, как в примере ниже:

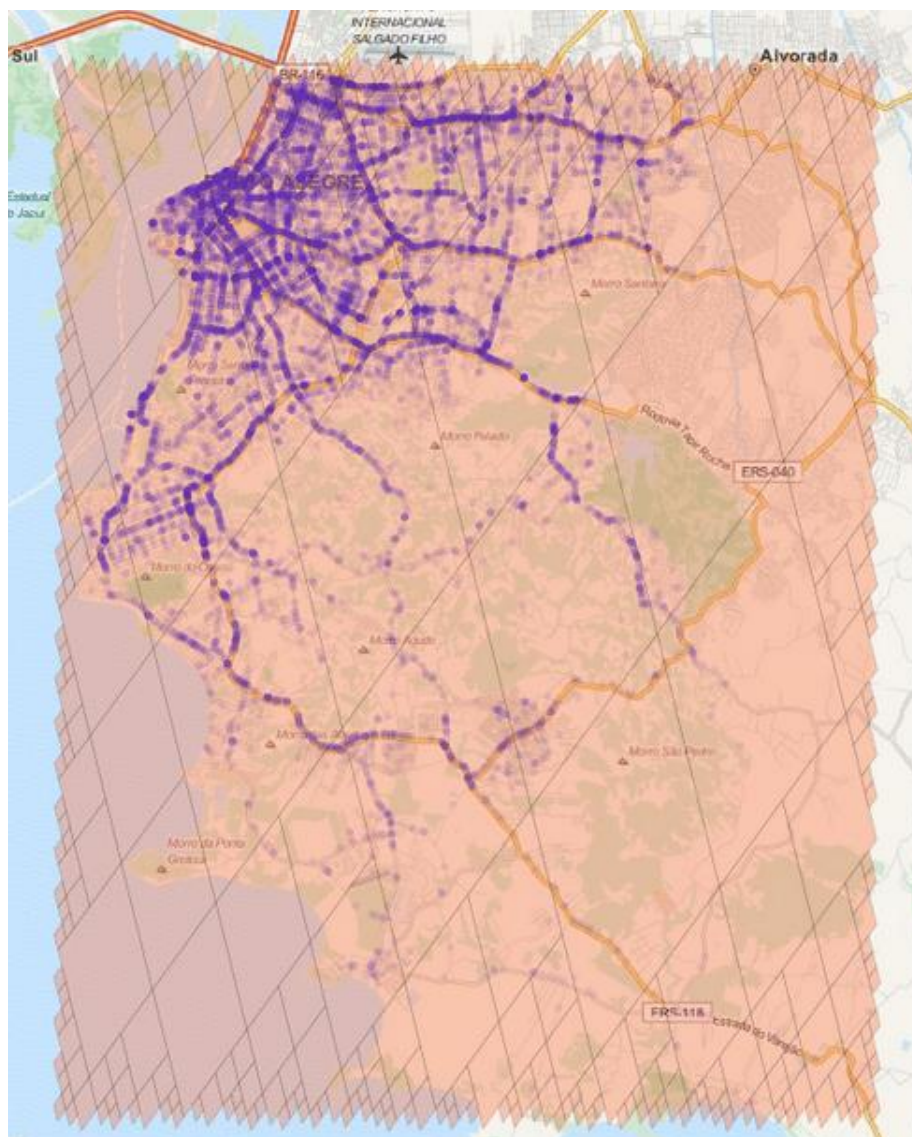


Рисунок 18 - Покрытие области ячейками 8-15 уровня, используя библиотеку S2 Geometry [21]

Как можно увидеть, у нас теперь есть покрытие, использующее более крупные ячейки в центре, и чтобы справиться с границами, есть приближение, использующее меньшие ячейки.

Исходя из [16], используя такое покрытие сегментами, скорость поиска необходимого объекта увеличивается на три порядка.

#### 1.4.2.6 Использование полигонов

Т.к. подразумевается, что проектируемая ГИС будет оперировать с фотографиями, сделанными на суше, то и учитывать сегменты, не относящиеся к материковой поверхности, нет необходимости.

Библиотека позволяет ограничить материковую территорию, используя полигоны (Рисунок 19).

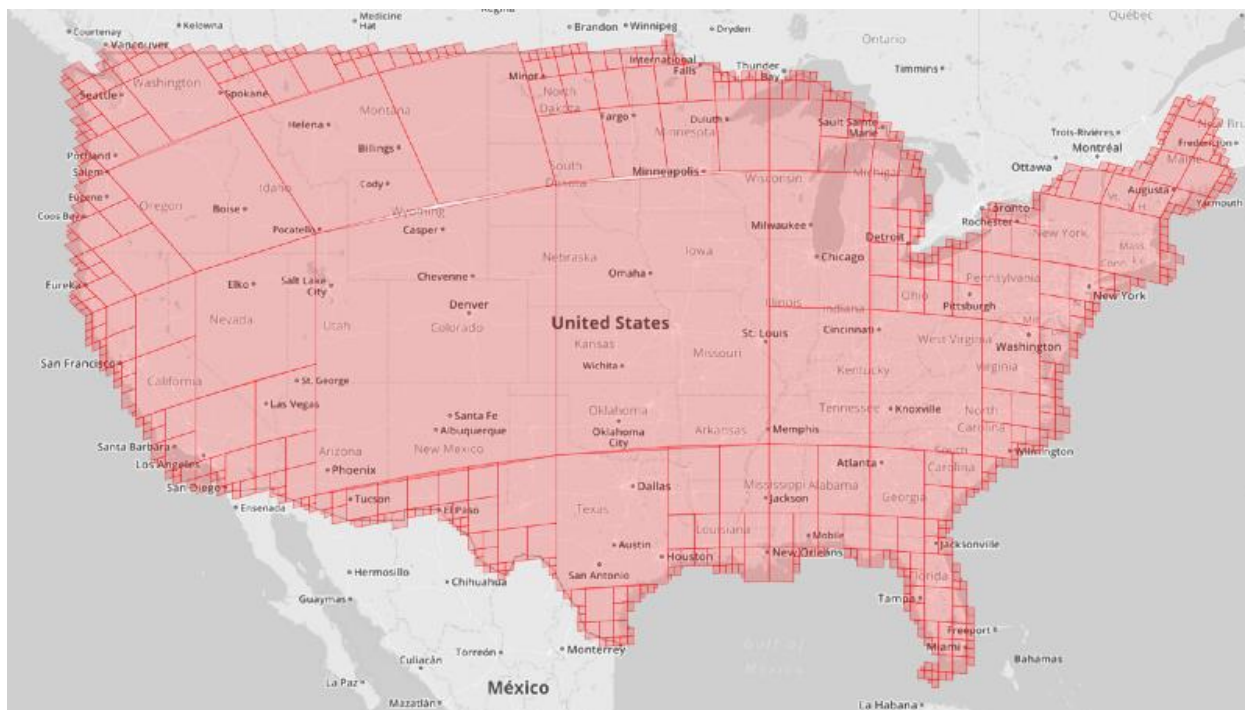


Рисунок 19 – Полигональное сегментирование территории США, используя библиотеку S2Geometry.

Для этого территория США, как показано на рисунке выше, задается массивом координат, который затем представляется в виде полигона.

С помощью формата обмена пространственными данными GeoJSON можно быстро и удобно преобразовать набор координат каждой страны в полигоны, чтобы затем использовать их для работы S2Geometry.

Пример представления набора координат в формате GeoJSON представлен в листинге ниже.

Листинг 1 – представление набора координат в формате GeoJSON

```
{ "type": "Polygon",  
  "coordinates": [  
    [ [100.0, 0.0], [101.0, 0.0], [101.0, 1.0], [100.0, 1.0], [100.0, 0.0] ]  
  ]  
}
```

Полная спецификация формата GeoJSON указана в [28].



Пакет геоданных, предоставляющий полигоны GeoJSON для всех стран мира, предоставляется сервисом Natural Earth - сообществом, созданного для создания визуально приятных, хорошо составленных карт с картографическим обеспечением в небольшом масштабе [29].

### 1.4.3 Сравнительная характеристика описанных методов

Первый явный недостаток Geohash – наличие всего 12 «уровней», разброс расстояния между которыми достигает больших значений. В случае с Google S2 Geometry – количество уровней – 31.

Разберем на примере поиска объектов в Париже, например, Эйфелевой башни. Необходимо заключить территорию города в как можно меньшее количество ячеек, при этом стараясь не задействовать территории, не относящиеся к Парижу.

Если мы будем использовать 5-й уровень ячеек для охвата города, то мы не сможем уместить левую часть города:

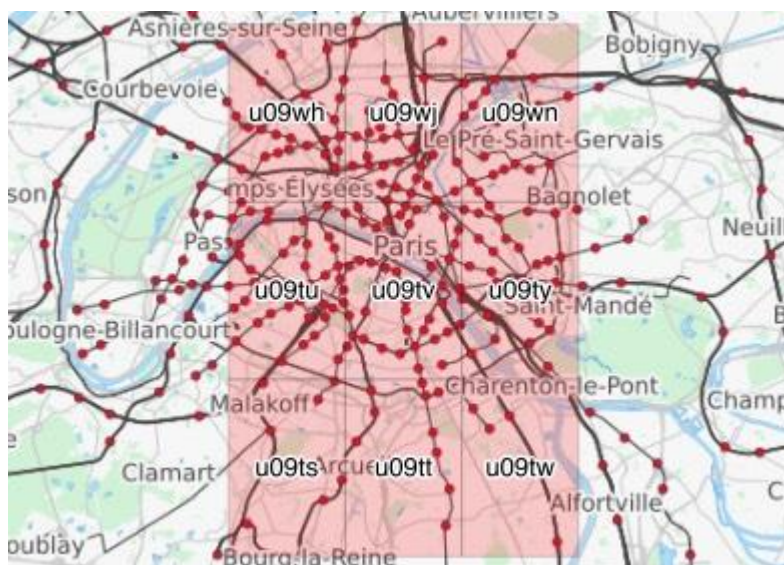


Рисунок 20 – Пример охвата г. Париж с помощью 5-го уровня ячеек GeoHash.

Можно добавить еще 3 ячейки слева (итого выйдет 12 ячеек), но большинство алгоритмов, работающих по методу Geohash, уменьшат масштаб до уровня 4 [22]. Но теперь в сегменты попадает весь регион:

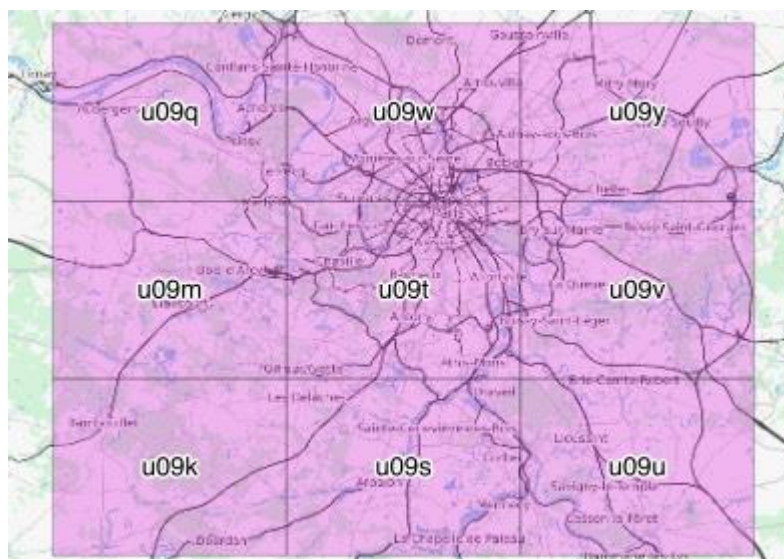


Рисунок 21 - Пример охвата г. Париж с помощью 4-го уровня ячеек GeoHash.

Тем временем, используя s2, сегментируем территорию г. Париж 9 ячейками, используя технологию сегментирования ячейками различного уровня (Описано в п. 1.4.2.4).



Рисунок 22 - Пример охвата г. Париж ячейками S2 разного уровня.

Итоговая картина выглядит так:

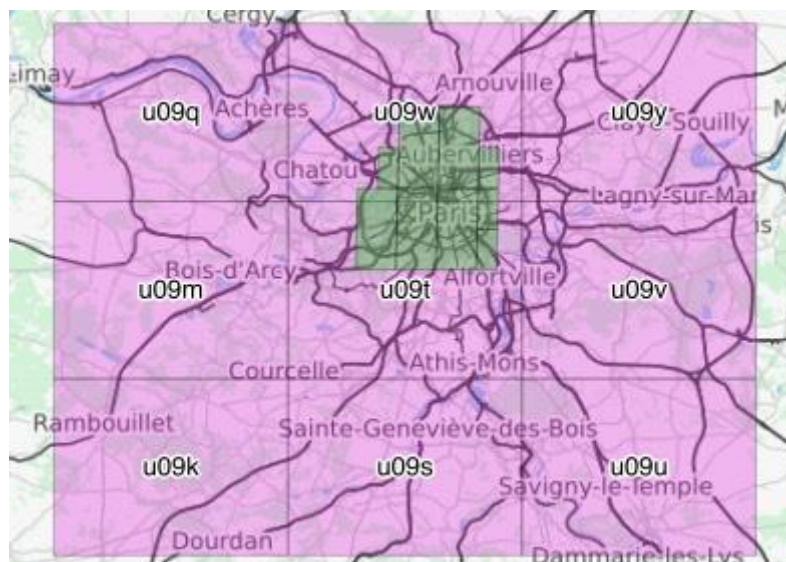


Рисунок 23 – Разделение поверхности г. Парижа на сегменты. Розовым – используя технологию Geohash, зеленым – S2 Geometry.

Таким образом, нельзя однозначно утверждать, что количество уровней GeoHash и размер ячеек, который соответствует тому или иному уровню, является оптимальным для покрытия территории различных размеров

Второй недостаток GeoHash вытекает из организации его работы. Ближайшая точка может иметь сильно различающееся значение Geohash, если местоположение объектов близко к границе квадрата сетки (см. Рисунок 24).



Рисунок 24 – Пример изначально различающегося значения GeoHash, если местоположение объектов близко к границе квадрата сетки.

В этом примере выше можем наблюдать, что эти 2 местоположения (зеленая точка и красная точка) довольно близки, но они находятся на

противоположных сторонах границы. Таким образом, представление их местоположения по формату GeoHash сразу же начнется с противоположных двоичных значений. Многие последующие горизонтальные деления будут видеть зеленую точку в крайнем правом углу («1») и красную точку в крайнем левом («0»). Поэтому очень мало корреляции в значениях GeoHash этих двух соседних точек.

Также нельзя не отметить тот факт, что для кодирования местоположения, используя систему GeoHash, потребуется от 1 до 12 байт памяти (т.к. каждый уровень GeoHash может кодироваться целочисленным и буквенным значением [13], то для представления будет задействован тип данных `char`, занимающий 1 байт памяти [23]).

Таким образом, исходя из вышеописанного, библиотека S2Geometry является более предпочтительной для использования в рамках реализации дипломного проекта.

## **1.5 Анализ сервисов для хранения больших массивов данных**

Поскольку при разработке проектируемой системы требуется проводить операции с большим массивом фотографий и метаданными к ним, целесообразно использовать удаленное хранилище для быстрого доступа ко всем данным.

Следует определить основные требования к сервисам для хранения больших массивов данных:

- Быстрая загрузка/выгрузка данных;
- Бесплатный доступ в образовательных целях;
- Открытый доступ к API и наличие подробной документации к нему.

В результате были выделены следующие сервисы:

- Dropbox;
- AWS S3.

Далее вышеуказанные сервисы будут рассмотрены детально.



### **1.5.1 Описание сервиса Dropbox**

Dropbox[24] – это файловый хостинг, который включает в себя программу-клиент, персональное облачное хранилище и синхронизацию файлов. Сервис позволяет создать специальное хранилище, которое синхронизирует удаленные данные и данные на компьютере. Доступ к файлам может осуществляться с помощью персонального компьютера, мобильного устройства, а также с помощью API и через сайт.

Для бесплатного пользования объем дискового пространства ограничивается размером 2 гигабайта [24].

Для интеграции с пользовательскими приложениями Dropbox предоставляет открытый доступ к Dropbox API. API обеспечивает доступ к таким функциям, как загрузка файлов, загрузка, совместное использование, поиск и восстановление.

Dropbox API поддерживает основные языки программирования, такие как Python, Java, C# [25]. Документация к API предоставлена достаточно подробно и на различных языках, в том числе и русском. Последняя версия API датирована 04.11.2015г.

### **1.5.2 Описание сервиса AWS S3**

Amazon Simple Storage Service (S3) — хранилище файлов любого типа, любого объема, с высокой доступностью и отказоустойчивостью. Оно разработано для хранения статического контента, пользовательских данных и бэкапов[26].

Сервис ориентирован предпочтительно для хранения статических данных и предоставляет 5 Гб хранилища для открытого пользования.

Разработчики могут пользоваться хранилищем AWS S3 с помощью веб-консоли, а также AWS S3 API.

Все файлы (фото, видео, документы) хранятся в контейнерах (bucket). Для быстрого доступа к данным доступны несколько регионов для создания своего контейнера (рис. 25), поэтому задержки доступа будут минимальными[27].

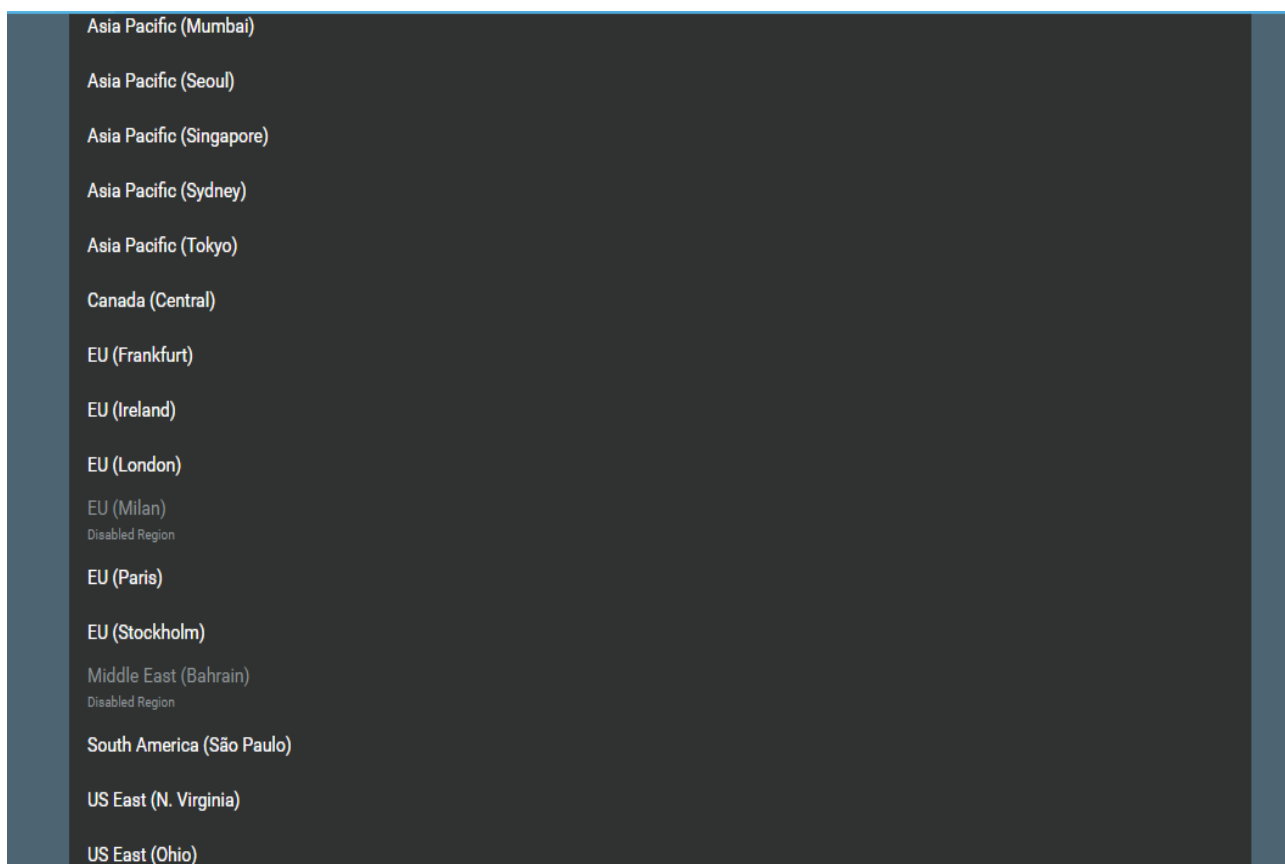


Рисунок 25 – Список доступных регионов для создания контейнера AWS S3

### 1.5.3 Сравнение времени доступа к данным, используя сервисы Dropbox и AWS S3

Т.к. основными объектами, которые предполагается загружать, хранить и выгружать, являются фотографии и CSV-файлы метаданных к ним, то целесообразно измерять время загрузки/выгрузки именно этих данных.

Результаты расчетов приведены в таблице 3.

Таблица 3 – Расчёт времени загрузки и выгрузки данных для хранилищ Dropbox и AWS S3.

<b>Сервис</b>	<b>Загрузка 100 фотографий, секунды</b>	<b>Выгрузка 100 фотографий, секунды</b>	<b>Загрузка файла метаданных, секунды</b>	<b>Выгрузка файла метаданных, секунды</b>
Dropbox	91,30	6,52	0,45	0,34
AWS S3	6,81	4,45	0,23	0,15

Как следует из расчётов, Dropbox незначительно уступает сервису AWS S3 по загрузке и выгрузке единичных файлов и сильно проигрывает по загрузке больших наборов данных.

Исходя из того, что сервис AWS S3 предоставляет 5 Гб хранилища для открытого пользования, нежели 2 Гб в случае Dropbox, а также из расчетов времени загрузки и выгрузки данных, было решено отдать предпочтение сервису AWS S3 для хранения данных.

## 1.6 Выводы по исследовательской части

В исследовательской части была доказана актуальность исследований поиска геопозиции по фотографии.

Были рассмотрены популярные существующие подходы к геолокации, такие как Google PlaNet и Im2GPS.

Проанализированы способы получения большого массива фотографий с геотегами. Рассмотрен способ загрузки с сервера Flickr, предполагающий использование Flickr API.

Такой принцип получения больших массивов фотографий является трудоемким и неоптимальным по затраченному времени, поэтому было решено

использовать готовые датасеты фотографий с возможностью дополнения их, используя исходные коды запросов из [7].

Были рассмотрены три готовых датасета фотографий с геотегами. Рассмотрена технология хранения метаданных для набора фотографий в виде CSV-файла.

Проведен сравнительный анализ методов разбиения земной поверхности на непересекающиеся сегменты, рассмотрены библиотеки Google S2 Geometry и GeoHash.

Рассмотрены популярные сервисы для удаленного хранения данных, такие как Dropbox и AWS S3. Произведен расчет времени загрузки и выгрузки данных на хранилища, предоставляемые вышеуказанными сервисами.

## **2 Проектирование и разработка подсистемы подготовки больших массивов фотографий для системы геолокации объектов на основе их фотографий**

На этапе проектирования и разработки необходимо разработать структуру проектируемой подсистемы, спроектировать и разработать алгоритмы этапов подготовки больших массивов изображений и сегментирования земной поверхности на непересекающиеся сегменты, выбрать язык и среду программирования.

### **2.1 Структура системы геолокации объектов на основе их фотографий**

Анализ известных подходов к определению местоположения объектов по фотографиям и их реализаций показал отсутствие приемлемых, с точки зрения качества распознавания, готовых решений. В связи с этим, для решения практических задач, необходимо разработать собственную систему геолокации, для которой предлагается следующий алгоритм работы.

1. По фотографии из тестового набора (у которой отсутствуют или искажены метаданные) попробовать найти похожую фотографию, затем найти ее описание (или проанализировать окружающий контекст) и метаданные, если

они есть. Затем извлечь информацию геолокации путем интеллектуального анализа собранной текстовой информации.

2. Проанализировать фотографию из тестового набора с помощью обученной нейросети (или нескольких специализированных нейросетей, предварительно обученных способами, описанными ниже). При этом могут применяться методы компьютерного зрения (фильтрация помех, удаление фона и т.п.) В результате анализа формируется вектор признаков (тегов) и оценка соответствия тега изображению. Из тегов формируется поисковый запрос и выполняется поиск изображений в имеющийся базе данных изображений или в Интернете. Определяется наиболее похожее из них, а также его название и контекст, который затем анализируется способом, приведенном в пункте 1.

3. Результат анализа изображения может проверяться и уточняться с использованием онтологических справочников. В результате проверки может быть улучшено качество геолокации, исправлены некоторые ошибки и обоснован полученный результат.

На рис. 26 показана обобщённая структура многоагентной ГИС определения местоположения объектов по их фотографиям, реализующая предлагаемый алгоритм работы.

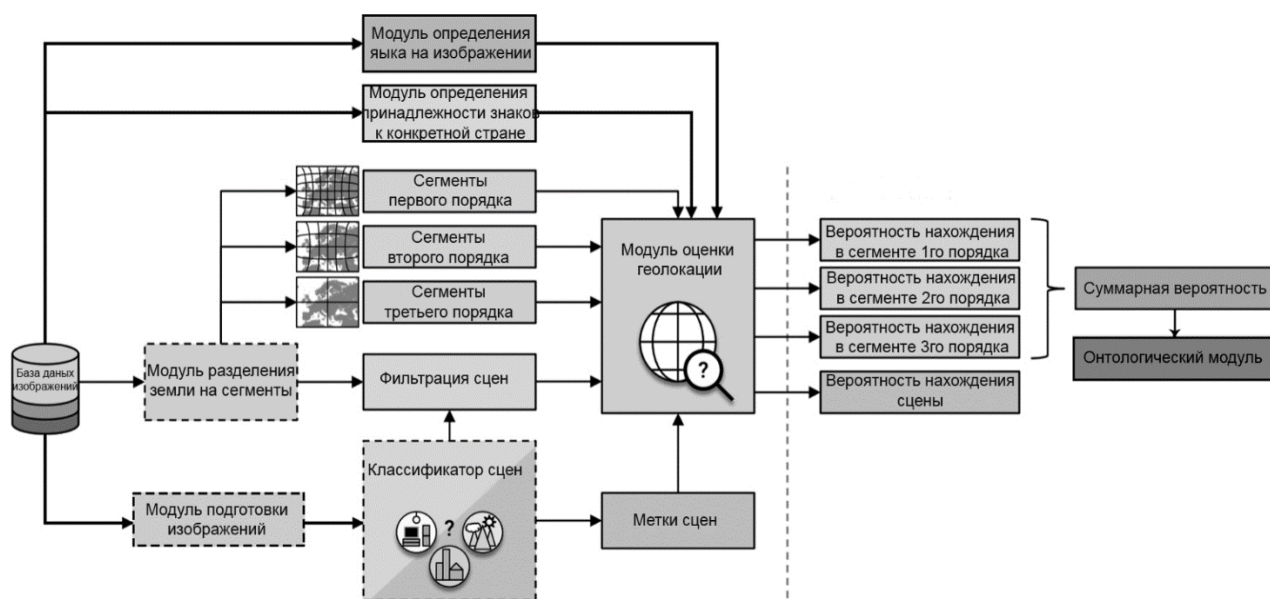


Рисунок 26 - Обобщённая структура многоагентной ГИС определения местоположения объектов по их фотографиям

## 2.2 Структура подсистемы подготовки больших массивов фотографий

Обобщенная схема функционирования проектируемой подсистемы представлена на рисунке 27.



Рисунок 27 - Обобщенная схема подсистемы подготовки больших массивов фотографий

Проектируемая подсистема больших массивов фотографий состоит из следующих элементов:

1. Модуль подготовки больших массивов фотографий;
  - 1.1. Подмодуль поиска похожих фотографий для исходной выборки;
  - 1.2. Подмодуль выгрузки обновленной выборки на удаленное хранилище;

2. Модуль разделения земной поверхности на непересекающиеся смежные сегменты;

2.1. Подмодуль предварительного определения по координатам принадлежности фотографий к тому или иному сегменту;

2.2. Подмодуль адаптивного сегментирования земной поверхности.

В Таблице 4 приведено описание модулей и подмодулей структурной схемы подсистемы.

Таблица 4– Описание модулей и подмодулей структурной схемы подсистемы

Имя элемента	Описание модуля
Модуль подготовки больших массивов фотографий	Отвечает за наполнение исходной выборки фотографий. В результате работы данного модуля получаем новый расширенный датасет фотографий, а также CSV-файл с метаданными для каждой фотографии
Подмодуль поиска похожих фотографий для исходной выборки	Отвечает за поиск похожих фотографий для каждой фотографии из исходной обучающей выборки
Подмодуль выгрузки обновленной выборки на удаленное хранилище	Компонует результаты работы предыдущего подмодуля и отправляет их на удаленное хранилище, добавляет в метаданные ссылки для каждой фотографии на удаленном хранилище

Продолжение таблицы 4

Имя элемента	Описание модуля
Модуль разделения земной поверхности на непересекающиеся смежные сегменты	Отвечает за разделение земной поверхности на непересекающиеся смежные сегменты на основе плотности распределения фотографий
Подмодуль предварительного определения по координатам принадлежности фотографий к тому или иному сегменту	Преобразует координаты для каждой фотографии в номер сегмента земной поверхности минимального размера
Подмодуль адаптивного сегментирования земной поверхности	Анализирует результат работы предыдущего подмодуля и, в зависимости от плотности фотографий в том или ином сегменте минимального размера, производит адаптивное разделение земной поверхности на непересекающиеся смежные сегменты

### 2.3 Проектирование модуля подготовки больших массивов фотографий

В качестве основного источника фотографий была взята обучающая выборка на 5000 фотографий, предоставленная организаторами конкурса ФПИ. Как определяется в [31], чем больше обучающая выборка, тем более точными будут результаты работы нейронной сети. Поэтому целесообразно максимально увеличить исходную выборку, не потеряв при этом полноты содержания.

Алгоритм работы модуля подготовки больших массивов фотографий изображен на рисунке 28





Рисунок 28 – Схема алгоритма модуля подготовки больших массивов фотографий

### 2.3.1 Проектирование подмодуля поиска похожих фотографий для исходной выборки

Для поиска похожих изображений было решено пользоваться Bing Cognitive Services Image Search API [32], поскольку для других поисковых сервисов (Google Search API, Yandex Search API) не предоставляется открытого доступа для студентов в рамках разработки своих проектов.

Для сервиса Bing Cognitive Services Image Search API есть одно ограничение – для одной фотографии можно получить только 35 похожих фотографий.

Поисковой запрос к сервису Bing Cognitive Services Image Search API выглядит следующим образом [33]:

[https://api.cognitive.microsoft.com/bing/v7.0/images/search\[?modules=similarimages\]](https://api.cognitive.microsoft.com/bing/v7.0/images/search[?modules=similarimages])

После того, того, как составлен запрос, следует отправить по полученному адресу POST-запрос, указав при этом необходимый файл.

В результате получаем JSON-файл, анализируя который получаем похожие фотографии:

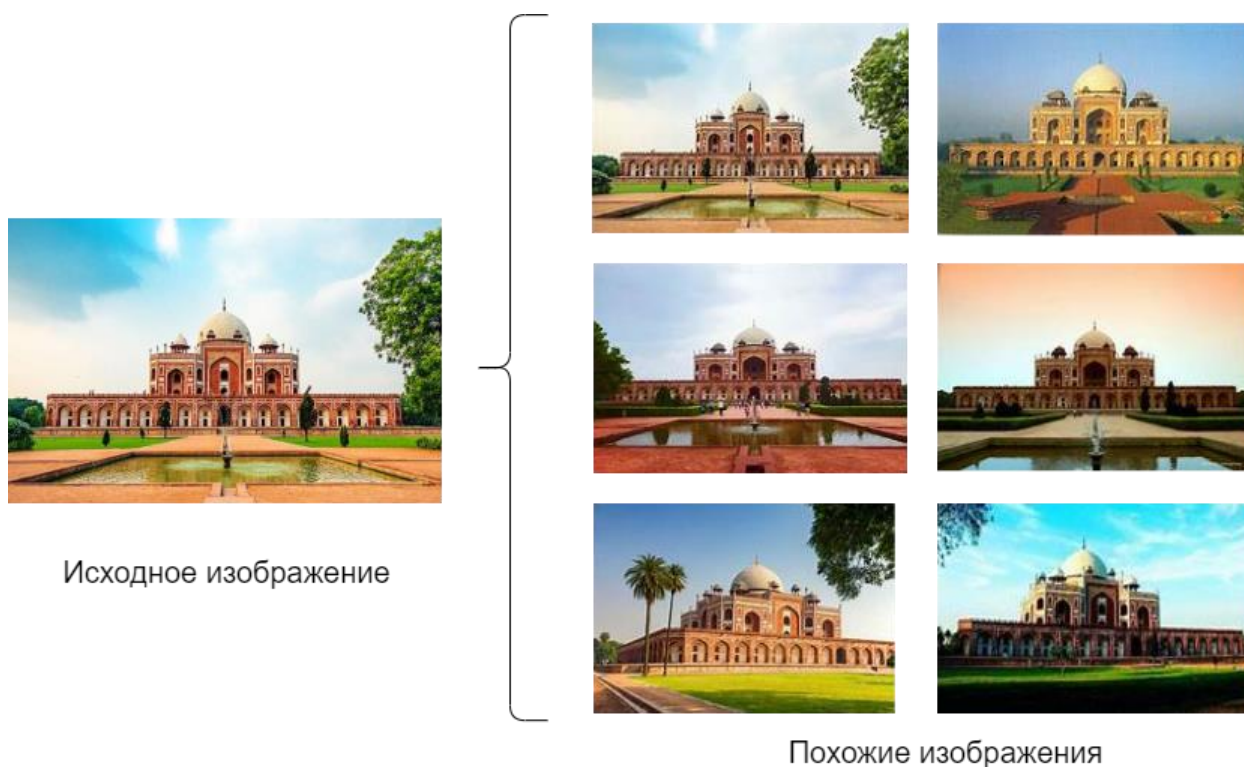


Рисунок 29 – Результат работы сервиса Bing Image Search API

Алгоритм работы подмодуля поиска похожих фотографий представлен на рисунке 30.

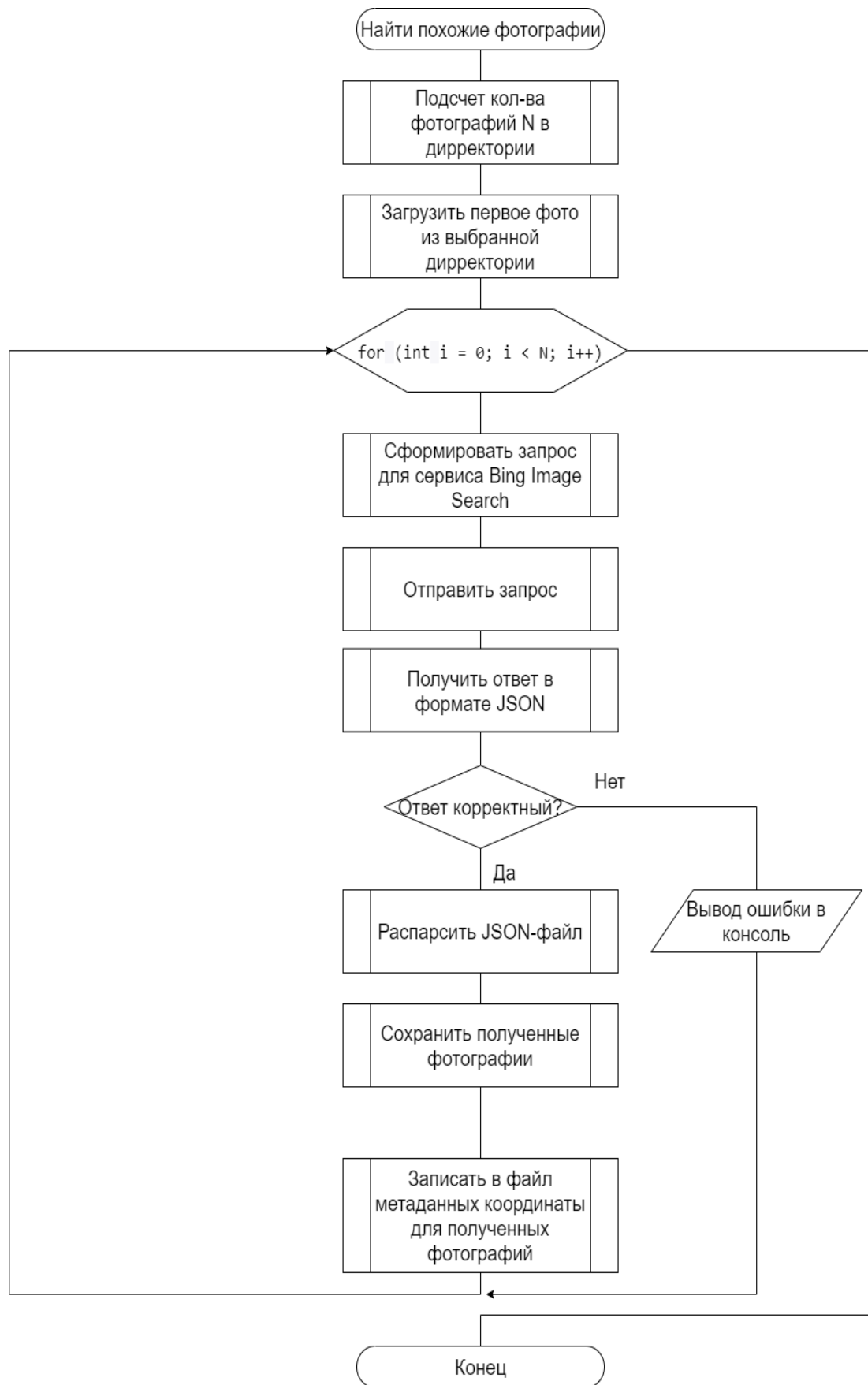


Рисунок 30 – Схема алгоритма подмодуля поиска похожих фотографий

### 2.3.2 Проектирование подмодуля выгрузки обновленной выборки на удаленное хранилище

В результате работы предыдущего модуля количество фотографий исходного датасета будет увеличено в 35 раз, поскольку для каждой фотографии найдется 35 схожих.

Для удобного и оперативного доступа к исходной и расширенной выборкам было решено использовать облачное хранилище AWS S3.

Для получения доступа к AWS S3 API необходимо создать аккаунт в системе Amazon, а также инициализировать создание нового файлового хранилища (корзины).

Когда корзина была создана, доступ к ее содержимому был ограничен участниками по проектированию и разработке системы геолокации объектов на основе их фотографий (рис.31).

Access for other AWS accounts

Canonical ID ⓘ	List objects ⓘ	Write objects ⓘ	Read bucket permissions ⓘ	Write bucket permissions ⓘ
<input type="text" value="ilay199640@gmail.com"/>	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes
<input type="text" value="dmitryb2007@yandex.ru"/>	<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes
<input type="text" value="nadya.suh.24@yandex.ru"/>	<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes	<input type="checkbox"/> Yes

Рисунок 31 – Список пользователей, имеющих доступ к файловому хранилищу.

Были созданы директории для исходного датасета, для расширенного датасета фотографий и для датасета фотографий, классифицированных по географической принадлежности (см. рис.32)

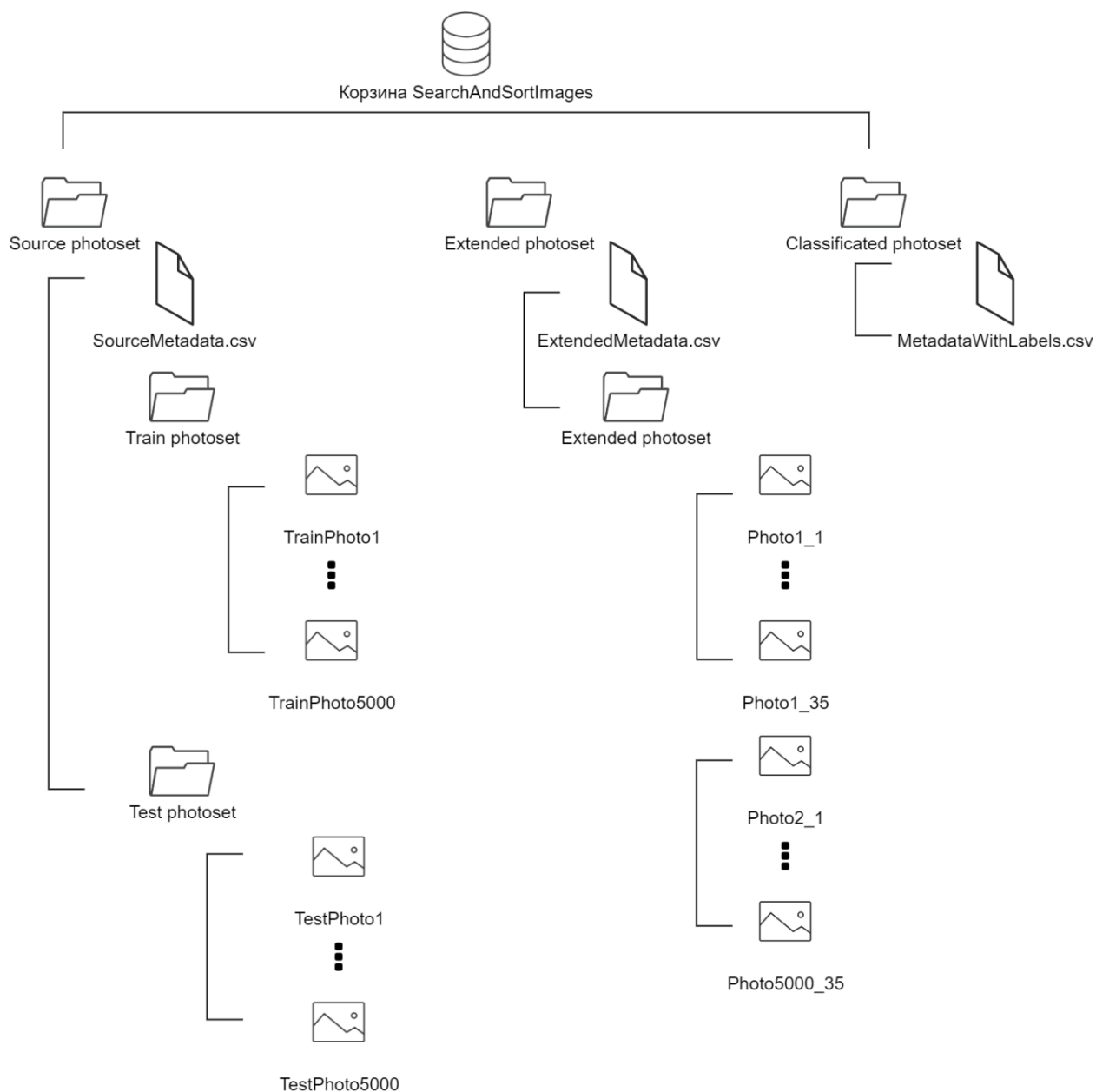


Рисунок 32 – Схема хранения данных на удаленном хранилище

## 2.4 Проектирование модуля разделения земной поверхности на непересекающиеся смежные сегменты

После расширения исходной обучающей выборки фотографий целесообразно приступить к разделению земной поверхности на непересекающиеся смежные сегменты.

Поскольку для реализации функционала данного модуля было решено использовать средства библиотеки S2Geometry, будем опираться на предлагаемые ей классы и методы.

Общая схема работы описываемого модуля представлена на Рисунке 33.

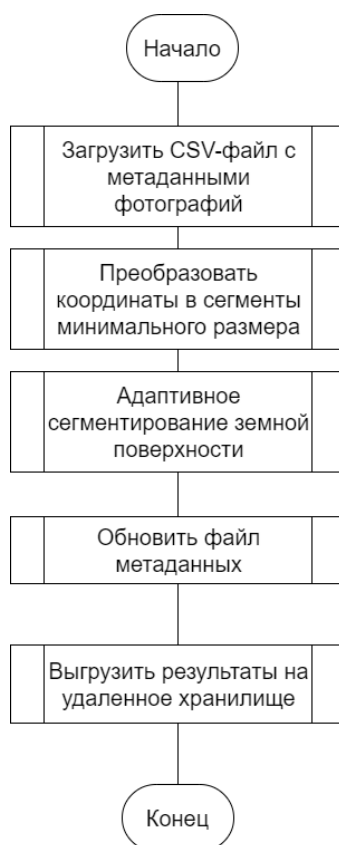


Рисунок 33 – Схема алгоритма модуля разделения земной поверхности на непересекающиеся смежные сегменты

#### 2.4.1 Проектирование подмодуля предварительного определения по координатам принадлежности фотографий к тому или иному сегменту

Описываемый подмодуль на вход получает исходный файл с метаданными о фотографиях, предварительно дополненный данными о похожих фотографиях.

Определение, к какому сегменту принадлежит та или иная фотография, происходит путем преобразования координат (ширина, долгота) сначала в радианы, а затем, с помощью средств S2Geometry - в CellID – идентификатор сегмента максимального размера (минимального уровня по классификации S2Geometry).

Алгоритм работы описываемого модуля приведен на Рисунке 34.

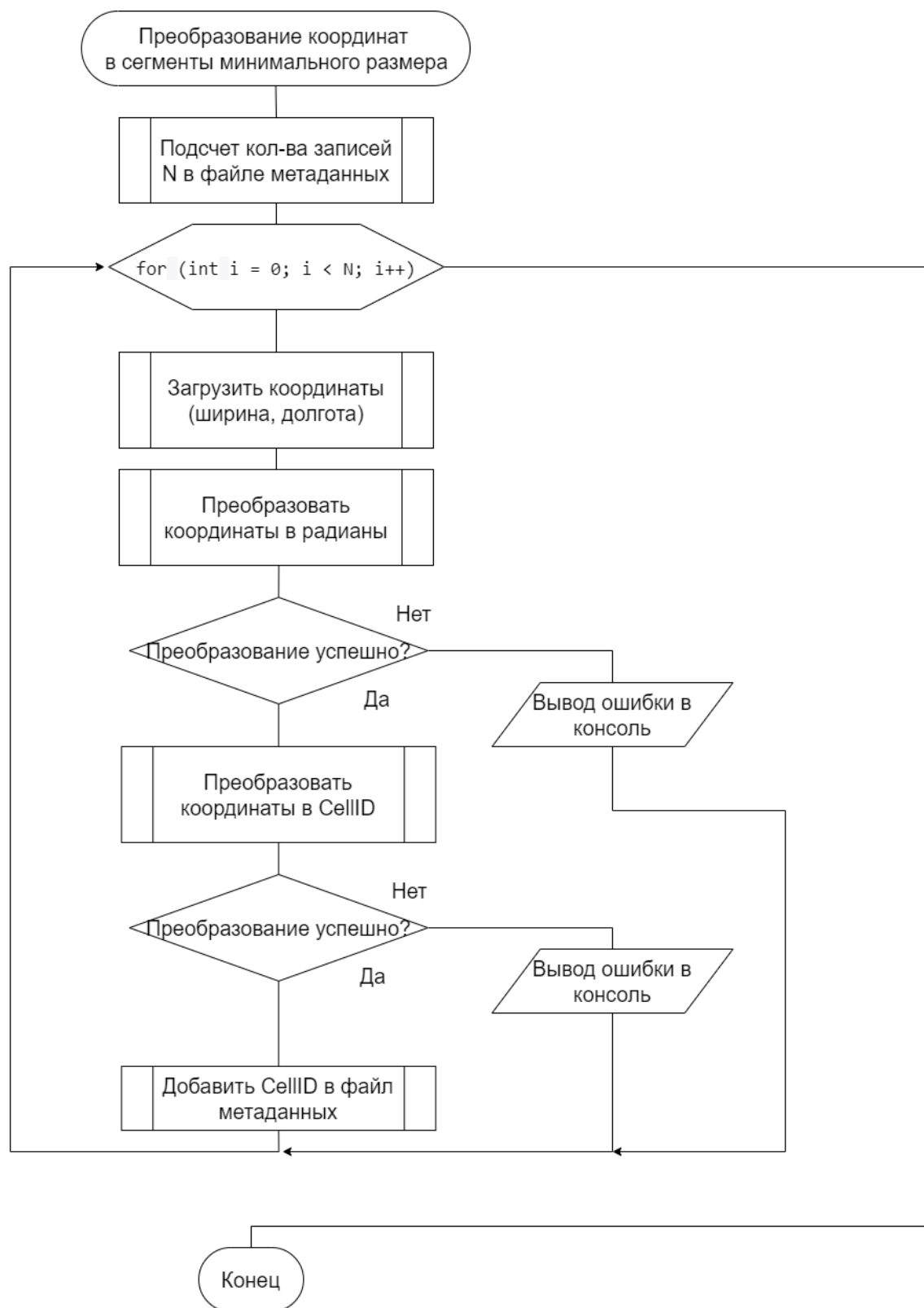


Рисунок 34 – Схема алгоритма модуля разделения земной поверхности на непересекающиеся смежные сегменты

## 2.4.2 Проектирование подмодуля адаптивного сегментирования земной поверхности

Для повышения быстродействия проектируемой системы целесообразно будет разделить поверхность Земли на непересекающиеся сегменты, причем они должны удовлетворять свойству адаптивности.

Т.к. поверхность Земли населена совсем не равномерно и фотографии из обучающей выборки распределены так же неравномерно, то и необходимо разработать алгоритм, задающий размер сегмента и необходимость его (например, в океане нет смысла выделять отдельный сегмент, т.к. плотность фото в нем будет околонулевая)

Таким образом, существование и размер сегмента прямо пропорционально должно соответствовать количеству фотографий, охватываемых им.

Опираясь на проект PlaNet от Google [4], ставший родоначальником библиотеки S2, а также на проект [30], можно выделить два необходимых параметра для сегментирования:

- 1)  $\tau_{\min}$  – минимальное необходимое количество фотографий для выделения сегмента
- 2)  $\tau_{\max}$  – максимальное количество фотографий, выше которого применяется следующий уровень сегментирования (текущий сегмент разбивается на 4 равные подсегмента)

В PlaNet использовалась обучающая выборка на 91 миллион фотографий с геотегами [30], при этом было задействовано 26 263 сегментов, причем значение  $\tau_{\min}$  и  $\tau_{\max}$  составляли соответственно 50 и 10000. Подберем необходимое значение  $\tau_{\min}$  и  $\tau_{\max}$  в зависимости от используемого датасета, чтобы сохранялось отношение прямой пропорциональности с значениями  $\tau_{\min}$  и  $\tau_{\max}$ , используемыми в решении PlaNet :



Таблица 5 – Расчет  $\tau_{\min}$  и  $\tau_{\max}$

Массив изображений	Количество изображений	$\tau_{\min}$	$\tau_{\max}$	$\tau_{\max} * 2$	$\tau_{\max}/2$
Исходная выборка	5000	1	2	4	-
Расширенная выборка	175000	1	19	38	10

Также стоит провести эксперимент, как изменится соотношение (скорость работы системы)/точность при изменении  $\tau_{\max}$  в обе стороны.

В результате получается такое сегментирование поверхности:

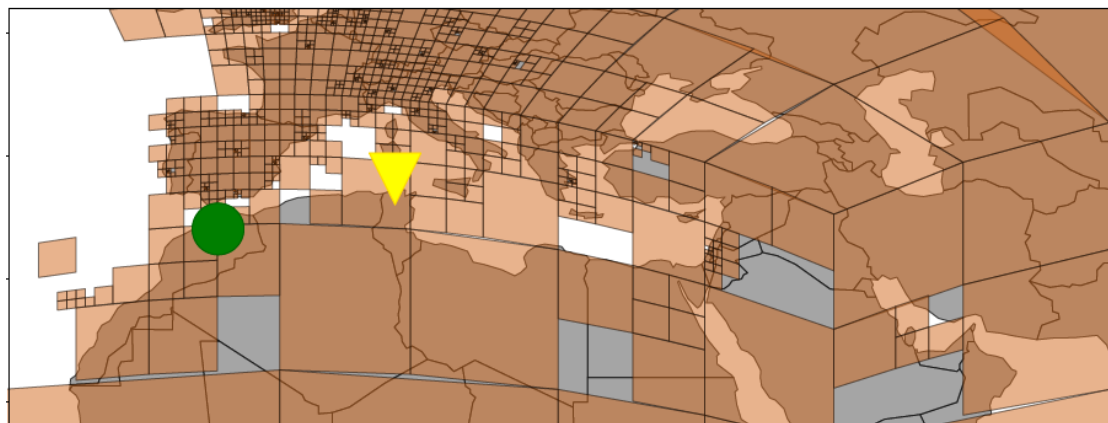


Рисунок 35 – Адаптивное разбиение земной поверхности на непересекающиеся смежные сегменты

Алгоритм процесса сегментирования представлен на рисунке ниже:

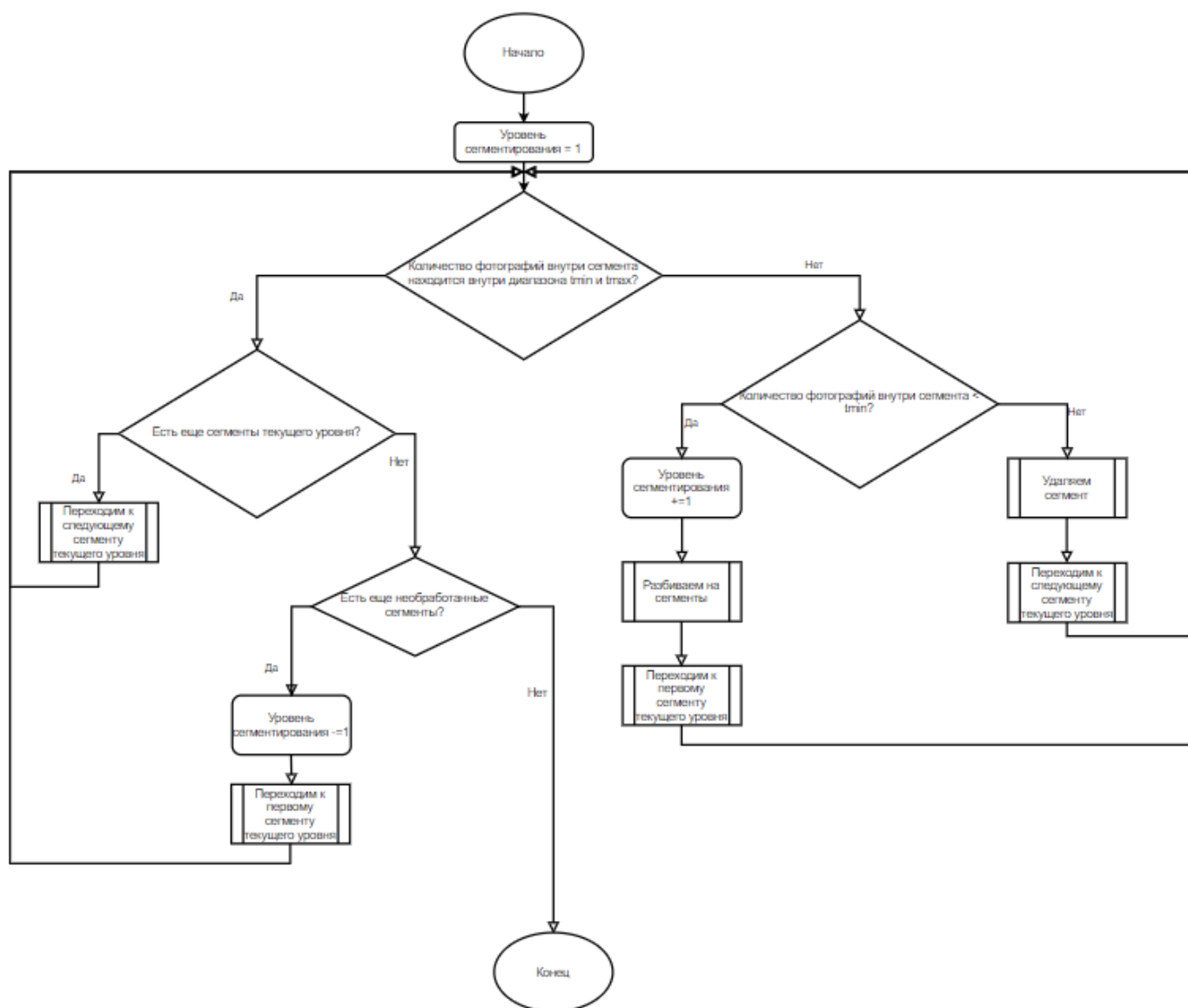


Рисунок 36 – Алгоритм разбиения земной поверхности на непересекающиеся смежные сегменты

## 2.5 Разработка диаграммы вариантов использования

"Вариант использования" - это описание поведения программы при ее взаимодействии с действующим лицом, в нашем случае действующим лицом будет администратор подсистемы.

Данная диаграмма поможет на начальном этапе разработки описать взаимодействие администратора с подсистемой, определить его возможные действия и их альтернативы.

Анализ особенностей работы проектируемой подсистемы позволил выделить следующие варианты использования:

- загрузить исходный массив данных;
- найти похожие фотографии;
- выгрузить данные на удаленное хранилище;
- определить сегмент для фотографий;
- разделить земную поверхность на непересекающиеся смежные сегменты.

Таблица 6 – Описание варианта использования *Загрузить исходный массив данных*

Название варианта	Загрузить исходный массив данных
Цель	Загрузить исходный массив данных для дальнейшей работы с ним
Действующие лица	Администратор подсистемы
Краткое описание	Загрузка исходного набора фотографий и метаданных к ним для дальнейшего наполнения.
Тип	Основной

Таблица 7– Вариант использования *Загрузить исходный массив данных* (типичный ход событий)

Действие исполнителя	Отклик подсистемы
1. Администратор выбирает пункт «Выбрать папку с фотографиями»	2. Открывается диалоговое окно стандартного проводника Windows
3. Администратор выбирает папку с фотографиями	4. Подсистема запоминает путь к папке с фотографиями
5. Администратор выбирает пункт «Выбрать файл метаданных»	6. Открывается диалоговое окно стандартного проводника Windows
7. Администратор выбирает CSV-файл метаданных	8. Подсистема запоминает путь к файлу метаданных

*Альтернатива:*

3 Администратор выбирает папку, где хотя бы один файл не соответствует по расширению файлу фотографии.

4 Подсистема уведомляет администратора, что не все файлы являются фотографиями.

*Альтернатива 2:*

5 Администратор выбирает поврежденный CVS-файл или посторонний файл.

6 Приложение уведомляет администратора, что не удалось считать файл метаданных.

Таблица 8 – Описание варианта использования *Найти похожие фотографии*

Название варианта	Найти похожие фотографии
Цель	Найти похожие фотографии для исходной фотографии
Действующие лица	Пользователь
Краткое описание	Процесс поиска похожих фотографий для исходной фотографии, используя API Bing Cognitive Services
Тип	Основной

Таблица 9 – Вариант использования *Найти похожие фотографии (типичный ход событий)*

Действие исполнителя	Отклик подсистемы
1. Администратор выбирает пункт «Найти похожие фотографии»	2. Подсистема запускает поиск похожих фотографий, используя API Bing Cognitive Services, загружает найденные фотографии в папку и обновляет файл метаданных

*Альтернатива:*

2 Подсистема выводит сообщение об ошибке при поиске фотографий, если такая ситуация происходит; поиск похожих фотографий приостанавливается.

Таблица 10 – Описание варианта использования *Выгрузить данные на удаленное хранилище*

Название варианта	Выгрузить данные на удаленное хранилище
Цель	<i>Выгрузить данные на удаленное хранилище AWS S3</i>
Действующие лица	Администратор подсистемы
Краткое описание	Процесс выгрузки данных на удаленное хранилище AWS S3
Тип	Основной

Таблица 11 – Вариант использования *Выгрузить данные на удаленное хранилище* (типичный ход событий)

Действие исполнителя	Отклик подсистемы
1. Администратор выбирает пункт «Выгрузить данные на удаленное хранилище»	2. Подсистема дописывает файл метаданных URL-ссылками для каждой картинки и выгружает все файлы на удаленное хранилище

*Альтернатива:*

2 Подсистема выводит сообщение об ошибке при выгрузке фотографий, если такая ситуация происходит.

Таблица 12 – Описание варианта использования *Определить сегмент для фотографий*

Название варианта	Определить сегмент для фотографий
Цель	Определить сегмент для каждой фотографии из указанной выборки
Действующие лица	Администратор подсистемы
Краткое описание	Определение сегмента для каждой фотографии из указанной выборки на основе геоданных (ширина, долгота)
Тип	Основной

Таблица 13 – Вариант использования *Определить сегмент для фотографий* (типичный ход событий)

Действие исполнителя	Отклик системы
1. Администратор выбирает пункт «Определить сегмент для фотографий»	2. Открывается диалоговое окно стандартного проводника Windows
3. Администратор выбирает файл метаданных	4. Подсистема запоминает путь к файлу метаданных и загружает его; появляется диалоговое окно с предложением ввести параметры $\tau_{\min}$ и $\tau_{\max}$

*Альтернатива:*

3 Администратор выбирает поврежденный CVS-файл или посторонний файл.

4 Подсистема уведомляет администратора, что не удалось считать файл метаданных.

*Альтернатива 2:*

3 Администратор не выбирает файл метаданных.

4 Подсистема использует файл метаданных, который был выбран последним; если такого нет – выводится сообщение об ошибке.

Таблица 14 – Описание варианта использования *Разделить земную поверхность на непересекающиеся смежные сегменты*

Название варианта	Разделить земную поверхность на непересекающиеся смежные сегменты
Цель	Разделить земную поверхность на непересекающиеся смежные сегменты
Действующие лица	Администратор подсистемы
Краткое описание	<i>Разделение земной поверхности на непересекающиеся смежные сегменты</i>
Тип	Основной

Таблица 15 – Вариант использования *Разделить земную поверхность на непересекающиеся смежные сегменты (типичный ход событий)*

Действие исполнителя	Отклик системы
1. Администратор выбирает пункт «Разделить земную поверхность на непересекающиеся смежные сегменты»	2. Открывается диалоговое окно стандартного проводника Windows
3. Администратор выбирает файл метаданных	4. Подсистема запоминает путь к файлу метаданных и загружает его; появляется диалоговое окно с предложением ввести параметры $\tau_{\min}$ и $\tau_{\max}$
5. Администратор задает параметры $\tau_{\min}$ и $\tau_{\max}$	6. Подсистема считывает значение параметров и выполняет сегментирование земной поверхности

*Альтернатива:*

5 Администратор не задает параметры  $\tau_{\min}$  и  $\tau_{\max}$ .

6 Подсистема использует стандартное значение параметров.

*Альтернатива 2:*

3 Администратор выбирает поврежденный CVS-файл или посторонний файл.

4 Подсистема уведомляет администратора, что не удалось считать файл метаданных.

*Альтернатива 3:*

3 Администратор не выбирает файл метаданных.

4 Подсистема использует файл метаданных, который был выбран последним; если такого нет – выводится сообщение об ошибке.

После анализа вышерассмотренных вариантов, для наглядного представления ожидаемого поведения приложения была построена диаграмма вариантов использования, приведенная на рисунке 37.



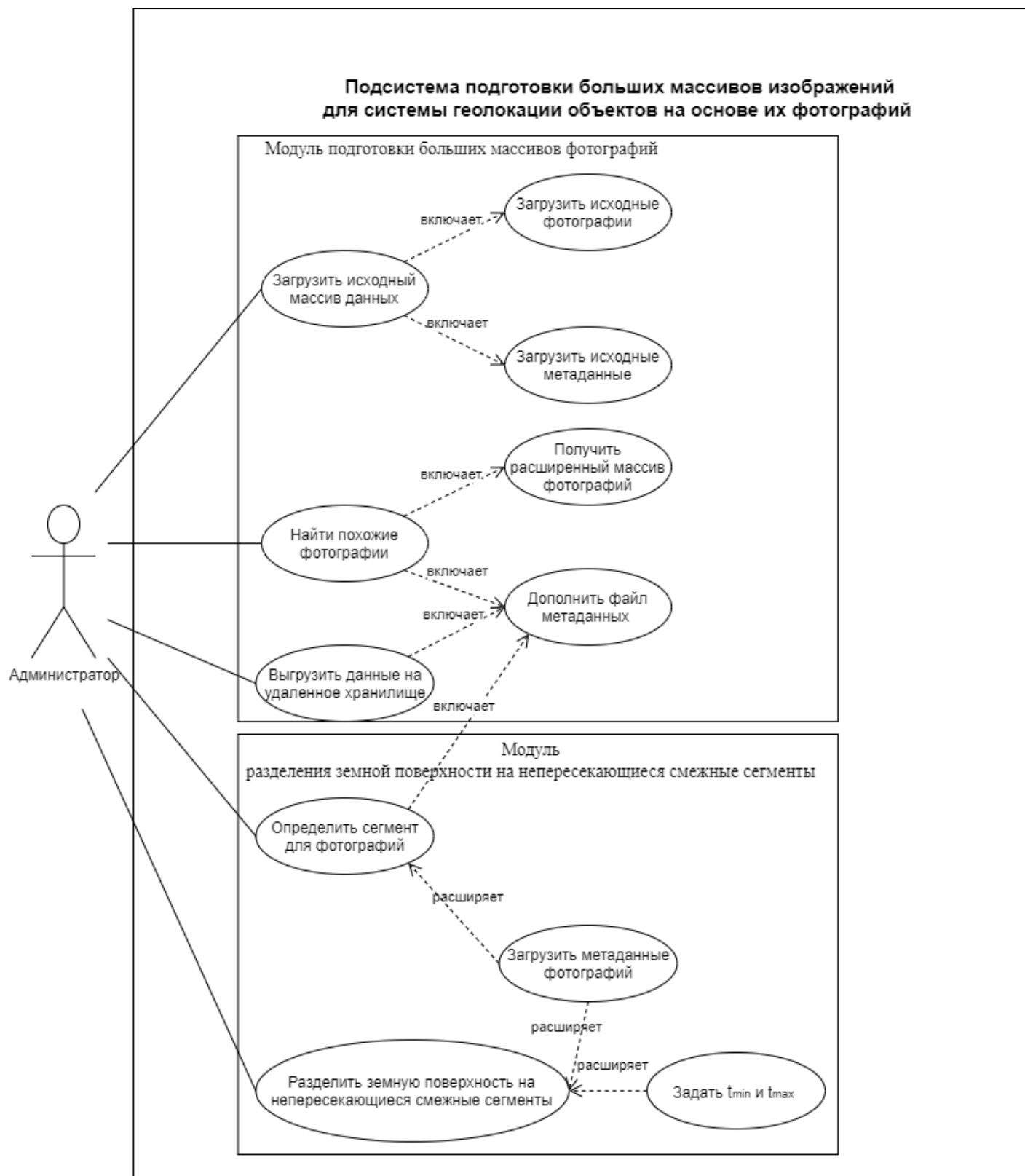


Рисунок 37 – Диаграмма вариантов использования

## 2.6 Разработка диаграммы состояний интерфейса

На основе анализа основных функций приложения и диаграммы вариантов использования была построена диаграмма состояний пользовательского интерфейса, представленная на рисунке 38.

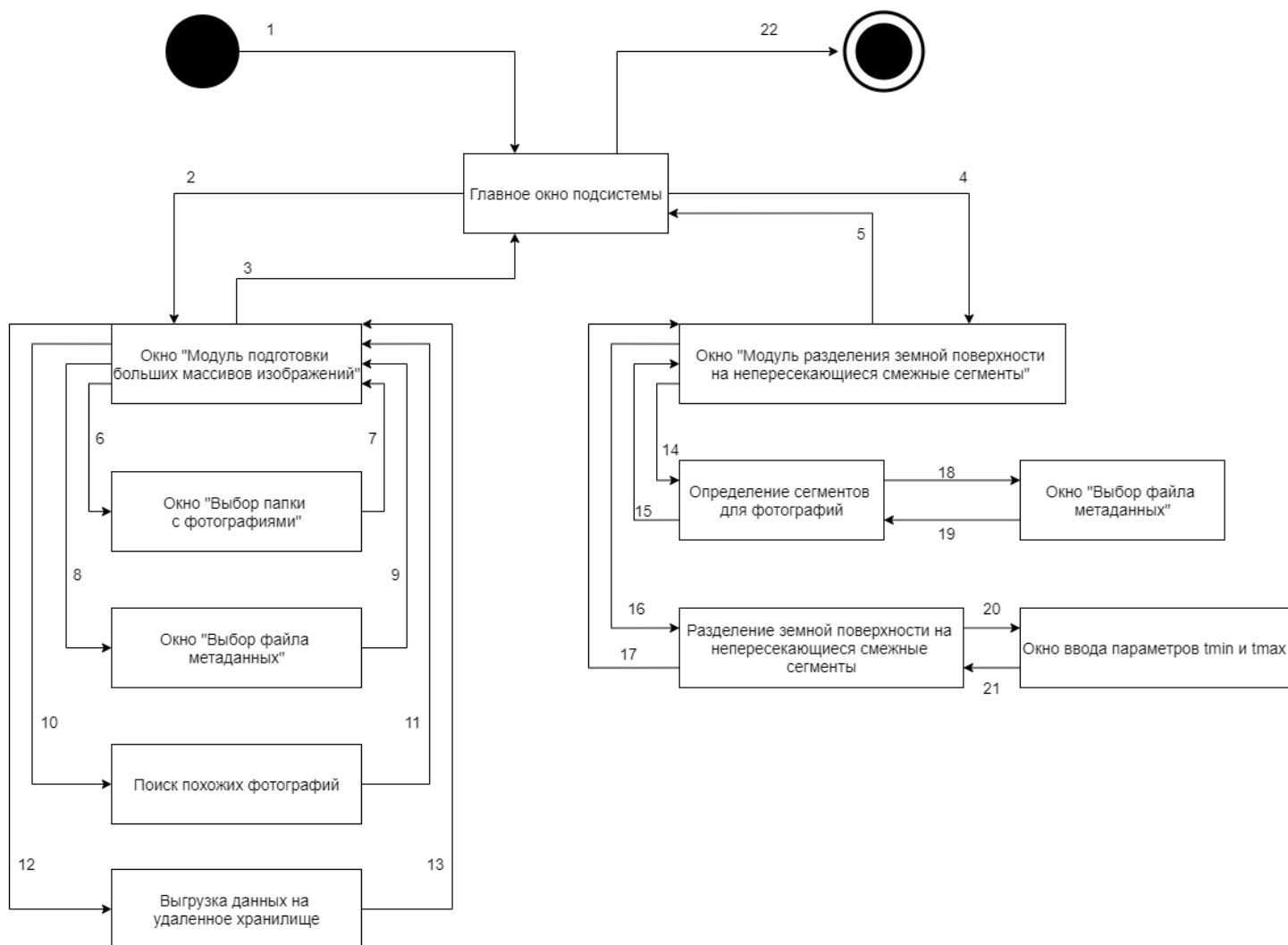


Рисунок 38 – Диаграмма состояний интерфейса

Числами 1 – 22 на диаграмме отмечены события:

- 1 - запуск подсистемы;
- 2 - нажатие вкладки «Модуль подготовки больших массивов изображений»;
- 3 - закрытие вкладки «Модуль подготовки больших массивов изображений»;

- 4 - нажатие вкладки «Модуль разделения земной поверхности на непересекающиеся смежные сегменты»;
- 5 - закрытие вкладки «Модуль разделения земной поверхности на непересекающиеся смежные сегменты»;
- 6 - нажатие кнопки «Выбрать папку с фотографиями»;
- 7 - подтверждение выбора папки с фотографиями;
- 8 - нажатие кнопки «Выбрать файл метаданных»;
- 9 - подтверждение выбора файла с метаданными;
- 10 - нажатие кнопки «Поиск похожих изображений»;
- 11 - завершение процесса поиска похожих изображений;
- 12 - нажатие кнопки «Выгрузка данных на удаленное хранилище»;
- 13 - завершение процесса выгрузки данных;
- 14 - нажатие кнопки «Определение сегментов для фотографий»;
- 15 - завершение процесса определения сегментов для фотографий;
- 16 - нажатие кнопки «Разделение земной поверхности на непересекающиеся смежные сегменты»;
- 17 - завершение процесса разделения земной поверхности на непересекающиеся смежные сегменты;
- 18 - запрос выбора файла с метаданными;
- 19 - подтверждение выбора файла с метаданными;
- 20 - запрос ввода параметров  $\tau_{\min}$  и  $\tau_{\max}$ ;
- 21 - подтверждение ввода параметров.

## **2.7 Разработка прототипов форм графического интерфейса**

## **2.8 Разработка диаграммы классов подсистемы**

## **2.9 Выводы по конструкторской части**

В конструкторской части была представлена обобщенная структура системы геолокации на основе их фотографий, спроектирована структура подсистемы подготовки больших массивов фотографий, в состав которой входят следующие компоненты:

1. Модуль подготовки больших массивов фотографий;
  - 1.1. Подмодуль поиска похожих фотографий для исходной выборки;
  - 1.2. Подмодуль выгрузки обновленной выборки на удаленное хранилище;
2. Модуль разделения земной поверхности на непересекающиеся смежные сегменты;
  - 2.1. Подмодуль предварительного определения по координатам принадлежности фотографий к тому или иному сегменту;
  - 2.2. Подмодуль адаптивного сегментирования земной поверхности.

Были разработаны алгоритмы работы вышеуказанных элементов подсистемы, спроектирована структура хранения данных, используемых подсистемой, на удаленном хранилище AWS S3.

Разработана диаграмма последовательности действий, диаграмма состояний интерфейса подсистема, диаграмма классов. Спроектированы прототипы графических форм интерфейса подсистемы.

## **3 Тестирование спроектированной подсистемы**

## ЗАКЛЮЧЕНИЕ

В результате проделанной работы было выполнено следующее:

- проведен обзор существующих подходов к геолокации, спроектировано собственное решение;
- проанализированы методы получения больших объемов данных и сегментирования земной поверхности;
- разработана структурная схема разрабатываемой подсистемы;
- разработаны рабочие компоненты подсистемы с использованием библиотеки S2 Geometry для сегментирования земной поверхности;
- проведено тестирование и отладка компонентов подсистемы.

Дальнейшее развитие подсистемы подразумевает внедрение разработанных модулей в состав системы геолокации объектов на основе их фотографий для конкурса ФПИ 2020.

Результаты, посвященные системе геолокации объектов на основе их фотографий были изложены в следующих публикациях:

- Березкин Д.В. , Сергеев И.Е., Сидоров В.Ю. Разработка элементов гибридной интеллектуальной системы определения местоположения объектов по их изображениям // Гибридные и синергетические интеллектуальные системы : материалы V Всероссийской Поспеловской конференции с международным участием / под ред. д-ра техн. наук, проф. А. В. Колесникова. — Калининград: Изд-во БФУ им. И. Канта, 2020. (принято к публикации).
- Березкин Д.В., Терехов В.И., Секирин П.А., Сергеев И. Е., Сидоров В. Ю. Разработка гибридной интеллектуальной системы определения местоположения объектов по их изображениям // XVII Всероссийская научная конференция «Нейрокомпьютеры и их применение». Тезисы докладов. – М: МГППУ, 2019. – с. 162.

- Подход к созданию гибридной интеллектуальной системы определения местоположения объектов по их фотографиям/ А. В. Пролетарский, Д. В. Березкин, В. И. Терехов, П. А. Секирин, И. Е. Сергеев, В. Ю. Сидоров // Нейрокомпьютеры: разработка, применение. – 2019. – №1 – с. 30 – 39.
- Proletarsky A., Berezkin D., Sidorov V., Sergeev I. Application of the methods of hybrid artificial intelligence to determine the location of objects by their photography // Proceedings of the 16th International conference on Applied Computing 2019, Cagliari, Italy, Nov. 7-9, 2019. – pp 245 – 250. - ISBN: 978-989-8533-95-1

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Средство для проверки метаданных фотографии Jeffrey's Image Exif Viewer  
// <http://exif-viewer.com/> (дата обращения 06.12.2018)
- 2 M. Zaharieva, M. Zeppelzauer and C. Breiteneder. “Automated social event detection in large photo collections,”// Proceedings of the ACM International Conference on Multimedia Retrieval. ACM, 2013. – С.–167-174.
- 3 Суслов Сергей Игоревич, Гаврикова Ксения Владимировна. Места притяжения граждан в городском пространстве на примере Красногвардейского района Санкт-Петербурга // Общество: политика, экономика, право. 2017. №5. С.–1-3. DOI:10.24158/per.2017.5.3.
- 4 Weyand T., Kostrikov I., Philbin J. (2016) PlaNet - Photo Geolocation with Convolutional Neural Networks. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9912. - pp 37-55, Springer, Cham - DOI:[10.1007/978-3-319-46484-8\\_3](https://doi.org/10.1007/978-3-319-46484-8_3).
- 5 Street View Static API // <https://developers.google.com/maps/documentation/streetview/intro?hl=en>.
- 6 Flickr // The App Garden [Электронный ресурс] –Режим доступа: <https://www.flickr.com/services/api/upload.api.html> (дата обращения 01.02.2020)
- 7 Im2GPS // IM2GPS: estimating geographic information from a single image [Электронный ресурс] –Режим доступа: <http://graphics.cs.cmu.edu/projects/im2gps/> (дата обращения 01.02.2020)
- 8 YFCC100M // YFCC100M: The New Data in Multimedia Research [Электронный ресурс] –Режим доступа: <https://arxiv.org/pdf/1503.01817.pdf> (дата обращения 01.02.2020)
- 9 Использование набора данных изображений с маркировкой 100M Yahoo Flickr Creative Commons 100M (YFCC100M) // Сообщество Qiita [Электронный ресурс] – Режим доступа:

- [https://qiita.com/\\_akisato/items/66deb481ea3cedf388fa](https://qiita.com/_akisato/items/66deb481ea3cedf388fa) (дата обращения 01.02.2020)
- 10 World-Wide Scale Geotagged Image Dataset for Automatic Image Annotation and Reverse Geotagging // ResearchGate [Электронный ресурс] –Режим доступа: [https://www.researchgate.net/publication/259802674\\_World-Wide\\_Scale\\_Geotagged\\_Image\\_Dataset\\_for\\_Automatic\\_Image\\_Annotation\\_and\\_Reverse\\_Geotagging](https://www.researchgate.net/publication/259802674_World-Wide_Scale_Geotagged_Image_Dataset_for_Automatic_Image_Annotation_and_Reverse_Geotagging) (дата обращения 01.02.2020)
- 11 Milgram, S. The Small World Problem. Psychology Today 2 (1967), с 60–67.
- 12 Weyand T., Kostrikov I., Philbin J. (2016) PlaNet - Photo Geolocation with Convolutional Neural Networks. In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science, vol 9912. - pp 37-55, Springer, Cham - DOI:[10.1007/978-3-319-46484-8\\_3](https://doi.org/10.1007/978-3-319-46484-8_3)
- 13 GeoHash // Wikipedia – The Free Encyclopedia [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Geohash#cite\\_note-1](https://en.wikipedia.org/wiki/Geohash#cite_note-1) (дата обращения 01.06.2019)
- 14 GeoHash Intro // Phil Whelan blog [Электронный ресурс] – Режим доступа: <https://www.bigfastblog.com/geohash-intro> (дата обращения 01.06.2019)
- 15 S2Geometry // S2 Cells [Электронный ресурс] – Режим доступа: [http://s2geometry.io/devguide/s2cell\\_hierarchy](http://s2geometry.io/devguide/s2cell_hierarchy) (дата обращения 01.06.2019)
- 16 Знакомство с библиотекой S2 от Google и пример ее использования // Хабр [Электронный ресурс] – Режим доступа: <https://habr.com/ru/company/badoo/blog/352754/> (дата обращения 01.06.2019)
- 17 Mercator Projection // Wikipedia – The Free Encyclopedia [Электронный ресурс] –Режим доступа: [https://en.wikipedia.org/wiki/Mercator\\_projection](https://en.wikipedia.org/wiki/Mercator_projection) (дата обращения 01.06.2019)
- 18 S2 Cell Statistics // S2 Cells [Электронный ресурс] – Режим доступа: [http://s2geometry.io/resources/s2cell\\_statistics](http://s2geometry.io/resources/s2cell_statistics) (дата обращения 01.06.2019)



- 19 Hilbert Curve // Wikipedia – The Free Encyclopedia [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/Hilbert\\_curve](https://en.wikipedia.org/wiki/Hilbert_curve) (дата обращения 01.06.2019)
- 20 S2 Cell Statistics // S2 Cells [Электронный ресурс] – Режим доступа: [http://s2geometry.io/resources/s2cell\\_statistics](http://s2geometry.io/resources/s2cell_statistics) (дата обращения 01.06.2019)
- 21 Google’s S2, geometry on the sphere, cells and Hilbert curve // Terra Incognita by Christian S. Perone [Электронный ресурс] – Режим доступа: <http://blog.christianperone.com/2015/08/googles-s2-geometry-on-the-sphere-cells-and-hilbert-curve> (дата обращения 01.06.2019)
- 22 A fast geo database with Google S2 // Fabrice Anecche’s blog [Электронный ресурс] – Режим доступа: [https://blog.nobugware.com/post/2016/geo\\_db\\_s2\\_geohash\\_database](https://blog.nobugware.com/post/2016/geo_db_s2_geohash_database) (дата обращения 01.06.2019).
- 23 Типы данных // METANIT.COM – сайт о программировании [Электронный ресурс] – Режим доступа: <https://metanit.com/cpp/tutorial/2.3.php> (дата обращения 01.06.2019).
- 24 Что такое Dropbox // Dropbox official page [Электронный ресурс] – Режим доступа: <https://www.dropbox.com/ru/features> (дата обращения 01.06.2019).
- 25 Dropbox API // Dropbox API for developers [Электронный ресурс] – Режим доступа: <https://www.dropbox.com/developers/reference> (дата обращения 01.06.2019).
- 26 Amazon S3: обзор хранилища // Хайлоад – построение больших приложений [Электронный ресурс] – Режим доступа: <https://ruhighload.com/amazon+s3%3A+%D0%BE%D0%B1%D0%B7%D0%BE%D1%80+%D1%85%D1%80%D0%B0%D0%BD%D0%B8%D0%BB%D0%B8%D1%89%D0%B0> (дата обращения 01.06.2019).
- 27 Amazon S3 REST API Introduction// Amazon Simple Storage Server API [Электронный ресурс] – Режим доступа: <https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html> (дата обращения 01.06.2019).

- 28 Спецификация формата GeoJSON // Gis-Lab – Географические информационные системы и дистанционное зондирование [Электронный ресурс] – Режим доступа: [http://gis-lab.info/docs/geojson\\_ru.html#MultiPolygon](http://gis-lab.info/docs/geojson_ru.html#MultiPolygon) (дата обращения 01.06.2019).
- 29 Country Polygons as GeoJSON // National Earth – free vector and raster map data [Электронный ресурс] – Режим доступа: <http://www.naturalearthdata.com/downloads/10m-cultural-vectors/> (дата обращения 01.06.2019).
- 30 Eric Müller-Budack Kader Pustu-Iren Ralph Ewerth. Geolocation Estimation of Photos Using a Hierarchical Model and Scene Classification // Computer Vision – ECCV 2018, – pp. 575–592.
- 31 В.М. Татьянкин, И.С. Дюбко. (2015). Обучающая выборка в задаче распознавания образов при использовании нейронных сетей. Вестник Югорского государственного университета, Выпуск 2 (37), 2015г. С. 94-98
- 32 Image Search API v7 // Microsoft Docs [Электронный ресурс] –Режим доступа: <https://docs.microsoft.com/en-us/rest/api/cognitiveservices-bingsearch/bing-images-api-v7-reference#imageinsightsresponse> (дата обращения 15.05.2020)
- 33 Image Search API - v7 // Microsoft Cognitive Services [Электронный ресурс] –Режим доступа: <https://azure.microsoft.com/ru-ru/services/cognitive-services/> (дата обращения 15.05.2020)
- 34 Nam N. Vo, Nathan Jacobs, James Hays. Revisiting IM2GPS in the Deep Learning Era // ICCV 2017, – pp. 2640–2649.

## **ПРИЛОЖЕНИЕ А**

**Техническое задание на выпускную квалификационную работу  
магистра на 10 листах**

## **ПРИЛОЖЕНИЕ Б**

**Графические материалы (копии) на 10 листов**

## **ПРИЛОЖЕНИЕ В**

### **Фрагмент исходного кода программного обеспечения**

## **ПРИЛОЖЕНИЕ Г**

### **Руководство администратора**