

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**Кафедра прикладной математики, механики, управления и программного
обеспечения**

Войцеховский Артём Викторович

ОПРЕДЕЛЕНИЕ ПАРАМЕТРОВ ДВИЖЕНИЯ ОБЪЕКТОВ НА ПАНО-
РАМНЫХ СНИМКАХ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

по образовательной программе подготовки специалистов
по направлению 010503.65 «Математическое обеспечение и администриро-
вание информационных систем»

Владивосток

2013

Аннотация

В данной дипломной работе описана реализация программной системы «Анализ панорамных снимков», помогающей оперативно выявлять на них объекты, а также вычислять характеристики этих объектов. Система определяет такие характеристики объектов снимков, как скорость, траектория, расстояние до камеры, размеры и gps-координаты. Для наглядного представления относительного расположения объектов также возможно построение ортогональной проекции изображения.

Оглавление

Глоссарий	5
Введение.....	6
Глава 1 Существующие системы распознавания объектов на панорамных электронных снимках	8
1.1 Методы улучшения качества снимков	8
1.1.1 Изменение контраста	9
1.1.2 Видоизменение гистограмм	9
1.1.3 Изменение тонового распределения	10
1.1.4 Удаление малых компонент.....	10
1.1.5 Анализ пространственных частот с использованием гармонических функций	11
1.2 Методы обнаружения объектов на электронном снимке	11
1.2.1 Алгоритм утоньшения для нахождения скелета.....	11
1.2.2 Алгоритм построения скелета	12
1.3 Методы поиска похожих объектов	12
1.3.1 Сравнение графов	13
1.3.2 Сравнение контуров	13
1.4 Методы определения географических координат объекта по электронному снимку	14
1.4.1 Метод Винсента	14
1.4.2 Метод Гельмерта.....	14
1.5 Методы определения расстояния до объекта по электронному снимку	15
1.6 Методы построения ортогональной проекции изображения.....	17
1.7 Использование многопоточности в системах распознавания объектов	18
1.8 Выводы из обзора.....	20
Глава 2 Анализ и построение модели ПО и решаемых задач.....	21
2.1 Анализ ПО и решаемых задач.....	21
2.2 Модель предметной области	24
2.2.1 Онтологические соглашения	30
2.3 Постановка прикладных задач	37
Глава 3 Спецификация требований и проектирование	41
3.1 Требования	41
3.1.1 Общие требования	41
3.1.2 Пользовательские требования	42
3.1.3 Функциональные требования	43
3.1.4 Требования к входным данным	43
3.1.5 Требования к выходным данным	44

3.1.6 Требования к надёжности и мобильности системы	44
3.1.7 Требования к серверу БД	44
3.2 Проект.....	45
3.2.1 АКД	45
3.2.2 Диаграмма вариантов использования	48
3.2.3 Диаграмма взаимодействия	50
3.2.4 Диаграмма потоков данных	51
3.2.5 Схема базы данных	52
3.2.6 Проект данных	54
Глава 4 Реализация и тестирование.....	57
4.1 Средства реализации.....	57
4.2 Тестирование	57
4.3 Выводы.....	58
Глава 5 Испытания системы «Анализ панорамных снимков»	59
5.1 Описание предметной области испытаний.....	59
5.2 Автоматизация предметной области	59
5.3 Преимущества автоматизации	60
Заключение	61
Литература	63

Глоссарий

Азимут – угол направления камеры относительно направления на север.

Наклон - угол относительно вертикали.

Крен - угол наклона относительно горизонта.

Большая полуось эллипсоида – большая полуось сферы, представляющей модель Земли.

Коэффициент сжатия эллипсоида – понятие в геодезии; один из параметров, помогающий однозначно определить форму Земного эллипсоида.

Точка на экваторе – точка пересечения экватора с меридианом.

Введение

Для определения параметров движения объектов прибрежных акваторий часто используются оптические методы с расположением приборов, удалённых от поверхности океана. При таком положении средства наблюдения в меньшей мере подвержены вредному воздействию внешних факторов, таких как мощные волны, обрастание биоматериалом и износ конструкций и датчиков прибора наблюдения при неблагоприятных условиях. С помощью методов обработки изображений можно количественно оценить параметры любого прибрежного явления, которое различается визуально. Кроме того, использование видеооборудования для измерения физических величин получило значительное распространение в области компьютерного зрения и робототехники. Специфическим требованием средств видеонаблюдения является наличие широкого канала передачи данных. Еще одной проблемой этого типа датчиков является то, что точность измерения зависит от условий освещения, а в тёмное время суток необходимо наличие хотя бы небольшого освещения.

В ДВО РАН ведутся работы по развертыванию системы оперативного наблюдения побережья и акваторий залива Петра Великого. Спектр возможных применений системы видеомониторинга достаточно широк – от задач охраны акваторий и мест установки приборов наблюдения до решения фундаментальных задач океанологии. В частности, она может быть полезной для оценки параметров движения объектов на акватории. Эта задача состоит в оценке скорости объекта, траектории его движения, размеров, географических координат и расстояния до места создания анализируемого снимка. Также систему видеомониторинга планируется применять для построения ортогональной проекции местности, что позволит получить горизонтальный снимок акватории с обнаруженными на ней объектами. В настоящее время в системе видеомониторинга имеются стационарные поворотные камеры и база данных, в которую эти камеры сохраняют снимки акватории.

Однако для программного средства, которое анализировало бы снимки и объекты на них, методы не подобраны и не скомпонованы. Поэтому *актуальным* является разработка программного средства, которое позволяло бы анализировать снимки, определять на них объекты, вычислять их различные характеристики и строить ортогональную проекцию снимков.

Целью дипломной работы является разработка и реализация программного средства обнаружения на сериях панорамных снимков движущихся объектов, определения элементов их движения и построения ортогональной проекции снимков с обнаруженными объектами. Для достижения поставленной цели необходимо решить следующие **задачи**:

- а) проанализировать существующие алгоритмы определения параметров движения объектов и построения ортогональной проекции и выбрать подходящие для программной системы «Анализ панорамных снимков»;
- б) разработать недостающие алгоритмы определения параметров движения объектов и построения ортогональной проекции;
- в) сформулировать требования и спецификации к программному средству «Анализ панорамных снимков» и разработать его проект;
- г) реализовать систему «Анализ панорамных снимков» и провести её тестирование;
- д) провести экспериментальное исследование созданного программного средства «Анализ панорамных снимков».

Дипломная работа состоит из введения, четырёх глав, заключения, одного приложения и списка литературы из 33 источников. Во введении обосновывается актуальность темы исследования, обосновывается цель и выделяются задачи исследования. Первая глава посвящена обзору литературы на тему «Определение параметров движения объектов на панорамных снимках», вторая – анализу предметной области, третья - проектированию и реализации системы. В четвёртой главе описаны испытания системы в реальных условиях.

Глава 1 Существующие системы распознавания объектов на панорамных электронных снимках

Целью представленного обзора литературы является формулировка требований к программному средству и выбор тех алгоритмов, которые должны быть заложены в программное средство, решающее поставленную задачу. В обзоре рассматриваются существующие способы улучшения качества изображения для повышения эффективности его дальнейшего анализа, способы нахождения объектов снимка и вычисления их пространственных характеристик.

1.1 Методы улучшения качества снимков

При считывании аналоговой информации и её оцифровке всегда возникает шум из-за особенностей работы матрицы, особенностей отражающей поверхности, точечных шумов, дефектов при изготовлении объектива, избыточной яркости (блуждания) геометрической не идеальности линзы объектива, неоднородностей пзс-матрицы и эффектов дискретизации [10]. Перед анализом изображений целесообразно применять к ним преобразования, которые устраняли бы его, а также изменяли изображение так, чтобы анализирующие алгоритмы могли предоставить наиболее точные результаты работы.

Возможны следующие способы улучшения качества цифровой фотографии:

- Изменение контраста
- Видоизменение гистограмм
- Изменение тонового распределения
- Удаление малых компонент
- Анализ пространственных частот с использованием гармонических функций

1.1.1 Изменение контраста

Процедура улучшения изображений сводится к выполнению комплекса операций с целью либо улучшения визуального восприятия изображения, либо преобразования его в форму, более удобную для визуального или машинного анализа [11; 12].

Слабый контраст — наиболее распространенный дефект фотографических и телевизионных изображений, обусловленный ограниченностью диапазона воспроизводимых яркостей, нередко сочетающейся с нелинейностью характеристики передачи уровней.

Во многих случаях контраст можно повысить, изменяя яркость каждого элемента изображения.

1.1.2 Видоизменение гистограмм

Гистограмма распределения яркостей типичного изображения естественно-го происхождения, подвергнутого линейному квантованию, обычно имеет ярко выраженный перекося в сторону малых уровней; яркость большинства элементов изображения ниже средней [13]. На темных участках подобных изображений детали часто оказываются неразличимыми. Одним из методов улучшения таких изображений является видоизменение гистограммы. Этот метод предусматривает преобразование яркостей исходного изображения, с тем чтобы гистограмма распределения яркостей обработанного изображения приняла желаемую форму. Эндрюс, Холл и другие исследователи получили ряд улучшенных изображений путем выравнивания гистограмм распределения, т. е. в каждом случае они добивались равномерности распределения яркости обработанного изображения [14]. Фрей исследовал метод видоизменения гистограмм, который обеспечивал экспоненциальную или гиперболическую форму распределения яркостей улучшенного изображения [15]. Кетчам улучшил этот метод, применив «локальные», подсчитанные по части изображения, гистограммы [16].

1.1.3 Изменение тонового распределения

Улучшение изображений выполняется путем изменения значений интенсивности (полутоновых уровней) пикселей [13]. В большинстве программных пакетов для обработки изображений предусмотрено несколько команд для изменения вида изображения посредством преобразования значений пикселей с использованием некоторой функции, задающей способ замены входных значений интенсивности на новые выходные значения. Данный метод легко расширить таким образом, чтобы пользователь мог указать несколько различных областей изображения и применить к каждой из них различные функции преобразования — функции тонового распределения. Преобразование значений интенсивности часто называется растяжением (stretching), так как такая операция часто применяется для расширения динамического диапазона значений интенсивности слишком темных изображений на весь доступный диапазон уровней интенсивности.

1.1.4 Удаление малых компонент

Существуют способы выделения связных компонент на бинарном изображении и большого количества признаков, которые можно вычислять по множеству пикселей, составляющих одну связную компоненту [13]. Описание, сформированное для изображения, может содержать множество компонент, каждая из которых представляет область, выделенную на фоне изображения, и множество признаков, вычисленных для каждой компоненты. С учетом вычисленных признаков алгоритм может удалить из этого описания любую компоненту. Например, могут быть удалены компоненты, содержащие небольшое количество пикселей, или очень тонкие компоненты. Подобная обработка приводит к удалению ряда шумовых областей вблизи границы объекта.

1.1.5 Анализ пространственных частот с использованием гармонических функций

Теория рядов Фурье в математическом анализе показывает, как большинство действительных поверхностей и функций можно представить с помощью базиса гармонических функций [13]. Энергия, соответствующая базисным векторам, интерпретируется в терминах структуры представляемой поверхности (функции). Этот подход может применяться для обработки всего изображения или его отдельных фрагментов. Он заключается в использовании базиса Фурье для последующей фильтрации изображения или для принятия решений относительно изображения на основе распределения энергии между различными базисными векторами.

Например, для удаления высокочастотного шума можно устранить из изображения все компоненты, соответствующие высокочастотным гармоническим колебаниям.

1.2 Методы обнаружения объектов на электронном снимке

При фотографировании на снимках образуются шумы из-за особенностей работы матрицы, отражающей поверхности; также добавляются точечные шумы.

Для улучшения алгоритма выделения предлагается учитывать не только размеры объекта, но и его форму, а именно, строить скелет объекта и анализировать полученный граф [17].

Рассмотрены следующие методы:

- Алгоритм утоньшения для нахождения скелета
- Алгоритм построения скелета

1.2.1 Алгоритм утоньшения для нахождения скелета

Алгоритм основан на применении к изображению восьми масок, полученных из двух (см. табл.) путем поворота каждой на 90, 180 и 270 градусов [17].

Каждая маска отвечает за удаление горизонтальных, вертикальных и диагональных линий, наклоненных под углом 45 градусов.

0	0	0	2	0	0
2	1	2	1	1	0
1	1	1	2	1	2

Рисунок 1 – Маски утоньшения

Алгоритм заключается в следующем. Каждый пиксель изображения и его ближайшие соседи подвергаются тесту на соответствие наложенной маски. Маска накладывается таким образом, что текущий пиксель становится в ее середину, а ближайшие соседи покрываются остальной частью маски. Если сам пиксель и его соседи удовлетворяют условию, что под ячейкой маски с номером 0 находится фон объекта, а под ячейками с номером 1 – сам объект, то пиксель, лежащий под центральной ячейкой маски, удаляется на изображении.

1.2.2 Алгоритм построения скелета

Для анализа скелета объекта необходимо представить его в виде взвешенного графа [17]. Для дальнейшего анализа скелета необязательно иметь всю ветвистую структуру. Достаточно выделить самый длинный путь в скелете и связанные с ним относительно большие дуги (пути) в графе. Применительно к летательным объектам в таком случае будут выделены фюзеляж самолета и оба его крыла. По соотношению весов самого длинного пути и связанных с ним дуг можно отсеивать мешающие распознаванию объекты (деревья, мелкие водоемы и др.).

1.3 Методы поиска похожих объектов

Рассмотрены следующие методы:

- Сравнение графов
- Сравнение контуров

1.3.1 Сравнение графов

После достаточного упрощения скелетов сравниваются полученные и эталонные графы [18]. Полное соответствие графов совершенно необязательно, так что, если веса дуг или их отношения мало отличаются, графы считаются похожими. Для решения задачи можно применить метод сравнения иерархий графов. Для этого вершины одного графа упорядочиваются относительно какой-нибудь конечной вершины, а для другого графа подбирается подобная комбинация вершин и дуг. Упорядочение вершин визуально можно представить как подвешивание дерева за одну из конечных вершин. Если находится подобная комбинация вершин и дуг, полученный и эталонный объекты схожи по своей структуре.

1.3.2 Сравнение контуров

Сравнение контуров — распространённая задача, возникающая, например, при решении проблемы поиска заданного объекта на изображении (template matching) [19].

Самый простой вариант сравнения пары контуров — это рассчитать их моменты.

Момент — это суммарная характеристика контура, рассчитанная интегрированием (суммированием) всех пикселей контура.

Однако моменты имеют существенные недостатки:

- они не позволяют сравнить контуры одинаковой формы, но разных размеров, поэтому их, сначала нужно нормализовать (операция эквализации контуров (приведение к единой длине) — позволяет добиться инвариантности к масштабу).

- зависят от системы координат, а значит не позволят определить повернутую фигуру.

Поэтому, лучше использовать нормализованные инвариантные моменты.

- вычисление центральных моментов (central moments)

– вычисление нормализованных центральных моментов (normalized central moments)

Для поставленной задачи данный метод не пригоден из-за возможного резкого изменения контура объекта на соседнем снимке в результате того, что он может слишком сильно развернуться относительно своего предыдущего положения.

1.4 Методы определения географических координат объекта по электронному снимку

Рассмотрены следующие методы:

- метод Винсента
- метод Гельмерта

1.4.1 Метод Винсента

Рекомендуемые прямые и обратные решения были разработаны на основе обратной формулы Рейнсфорда [20; 21]. Прямое решение было получено от обратного, выполняемого в обратном порядке, используя подход Раппа [22].

По известной точке (φ_1, L_1) , заданному азимуту a_1 и дистанции s прямой метод Винсента находит координаты конечной точки (φ_2, L_2) .

1.4.2 Метод Гельмерта

Применяется адаптированный метод Гельмерта (1880) [23]. Работа выходит за рамки метода Винсента из-за увеличенной точности, чтобы соответствовать стандартной точности большинства компьютеров. Это относительно простая задача сохранения достаточного количества членов ряда разложения, и она может быть достигнута при небольших вычислительных затратах.

Геодезические задачи наиболее легко решаются с помощью вспомогательной сферы, которая позволяет сделать точное соответствие между геодезическим

и большим кругом на сфере. На сфере, широта ϕ заменяется уменьшенной широтой β , а азимут α сохраняется.

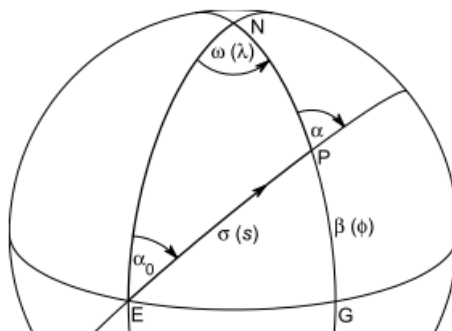


Рисунок 2 – Сфера

Из рисунка 2 очевидно, что уравнение Клеро, $\sin \alpha_0 = \sin \alpha \cos \beta$ – правило синуса, применённое к сторонам NE и NP треугольника NEP и их противолежащим углам. Третья сторона, сферическая длина дуги σ , и ее противоположный угол, сферическая долгота ω , связаны с эквивалентными величинами на эллипсоиде: расстояние x и долгота λ , в силу уравнение Рапп.

1.5 Методы определения расстояния до объекта по электронному снимку

Используется текущая математическая модель камеры с некоторыми известными мировыми координатами [24].

Практические ограничения на мировые координаты при использовании цифровых изображений возможно, не очевидны, но, во многих случаях, практически реализуемы. Один из способов состоит в том, чтобы представить образы в виде системы, в которой одна пространственная переменная **может быть задана непосредственно**, таким образом, добавляя третье уравнение и получая уникальное решение для преобразования изображения.

Например, можно зафиксировать z , как среднюю высоту над уровнем моря, полученную с помощью приборов на месте установки камеры наблюдения. Учитывая текущую высоту, можно вычислить высоту пикселей на изображении по их интенсивности. Для камер, ориентируемых вдоль берега, ошибка горизонтального расположения привносит погрешность в определение масштаба. Однако, относи-

тельная ошибка интервала (различие между спроектированными взвешенными шкалами расстояний) почти нечувствительна к ошибке оценки высоты и обычно меньше пиксела. Пример определения морфологических шкал расстояний от прямых прибрежных изображений показан на Рис. 3. Нормализованная экспозиция изображений показала отличные результаты для исследований морфологических изменений, дающих логически простое представление об изменениях огромных областей местности.

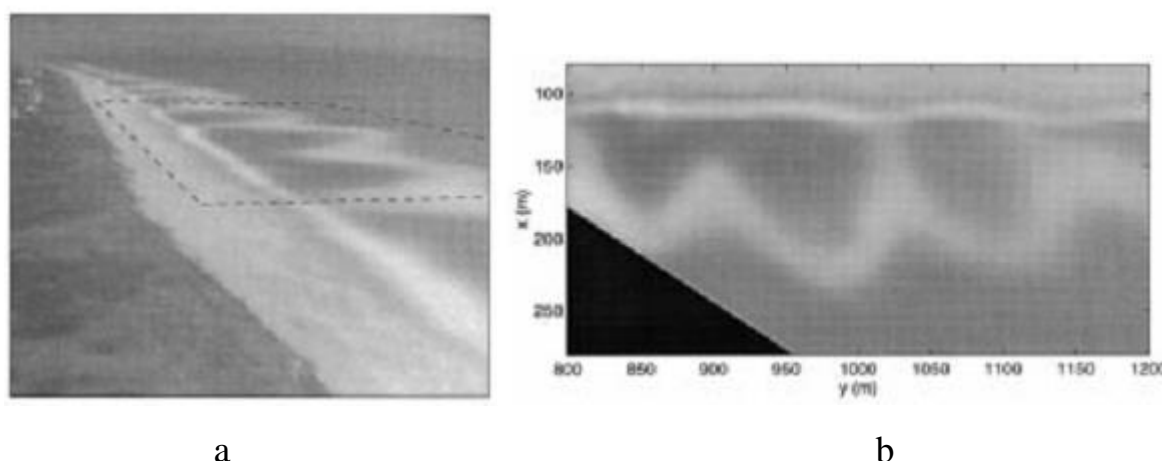


Рисунок 3 – Преобразование фотографии прибрежной зоны

Другая альтернатива определения мировых координат, используя снимки от одной камеры, состоит в том, чтобы ограничить множество мировых координат с помощью определённой поверхности. Пример этого метода, связанный с определением неизвестных мировых координат, заключается в отслеживании природных дрейфтеров (например, устройства защиты от мин для военных) в зоне, вдающейся в береговую линию. Этот случай схож с морфологическими методами измерений, за исключением того, что пространственное измерение не должно ограничиваться вертикальной или горизонтальной плоскостью (или границами вообще). Вместо этого координаты местности, соответствующие измеряемым положениям пикселей, выводятся путём математического определения координат поверхности земли. Координаты изображения, определённые вручную, соответствующие положе-

нию дрифтера, были преобразованы в мировые координаты с помощью коэффициентов калибровки камеры.

1.6 Методы построения ортогональной проекции изображения

Фотограмметрические задачи наиболее просто решаются по горизонтальным снимкам [25, с. 67]. Получить такой снимок при нынешнем состоянии средств аэрофотосъемки невозможно, да в этом нет и необходимости, поскольку при известных угловых элементах внешнего ориентирования измеренные на наклонном снимке координаты можно перевычислить на строго горизонтальный снимок. Этот процесс в фотограмметрии называется трансформированием координат.

Пусть из точки S получены горизонтальный P^0 и наклонный P снимки с изображениями m и m^0 точки местности M . Эти снимки пересекаются по линии неискаженных масштабов, и их фокусные расстояния одинаковы ($S_o = S_o^0 = f$). Точка m^0 горизонтального снимка имеет координаты x^0, y^0 , а точка m наклонного снимка - координаты x и y .

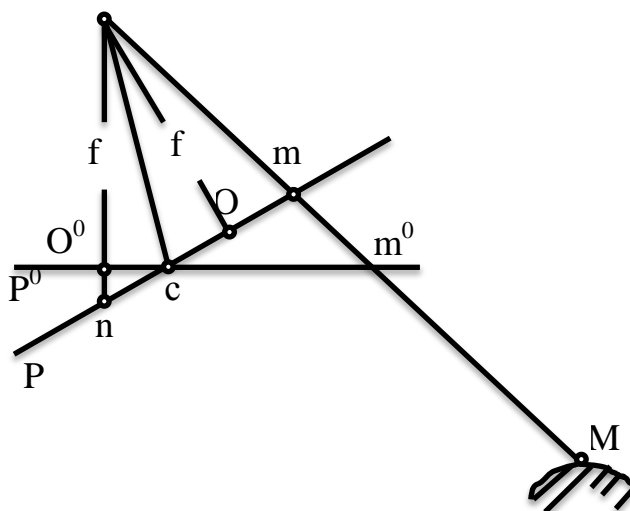


Рисунок 4 – Схема зависимости координат точек наклонного и горизонтального снимков

Рассматривая горизонтальный снимок P° вместе с расположенными на нем точками как плоскую местность и полагая отметки Z_M всех точек одинаковыми и равными нулю, а $X_S = Y_S = 0$, $X_M = Y_M = x^0$ и $Z_S = f$, становится возможным установить связь между координатами соответственных точек плоскостей P и P° .

Это позволяет преобразовать координаты точек наклонного снимка к горизонтальному случаю съемки при любых значениях угловых элементов внешнего ориентирования снимка.

1.7 Использование многопоточности в системах распознавания объектов

Многочисленные исследования показывают, что применение многопоточности при реализации алгоритмов в области компьютерного зрения может значительно увеличить быстродействие программного средства [26 ; 27]. В связи с тем, что решения в прикладных областях, подобных приведенным в списке, сводятся к ряду известных задач дискретной математики, большой практический интерес представляют параллельные алгоритмы их решения.

Алгоритмы, позволяющие выполнять параллельное суммирование с помощью многопроцессорных систем с общей памятью, могут выполняться с помощью каскадной схемы суммирования, модифицированной каскадной схемы [28]. В названном первоисточнике [28] приводится также схема вычисления всех частных сумм.

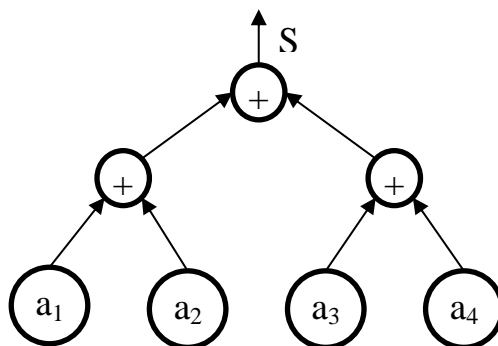


Рисунок 5 – Каскадная схема

Каскадная схема суммирования имеет вид бинарного дерева, в листьях которого записаны члены ряда, а во всех остальных узлах – операции сложения.

Для правильного построения многопоточных систем задачи, которые решают эти системы, принято разделять на следующие классы:

- Комбинаторные задачи
- Задачи линейной алгебры
- Вычисления в узлах сеток и решеток. Краевые задачи

Среди задач комбинаторики можно обратить внимание, в первую очередь, на параллельные методы сортировки и поиска данных [26].

Суть многих параллельных алгоритмов сортировки состоит в том, что упорядочиваемые данные разбиваются на какие-либо части в зависимости от метода, каждая из которых сортируется по отдельности разными процессорами [29]. Затем упорядоченные части объединяются некоторым образом, опять же в зависимости от метода. Эти два этапа могут повторяться необходимое число раз.

Типичным представителем задач, относящихся к линейной алгебре, является умножение матрицы на матрицу [30].

Методы решения этой задачи зависят от выбранной топологии вычислительной сети и от способа распределения матриц A , B , C по процессорам. Если сеть содержит p процессоров, то каждую из трех матриц можно распределить между ними одним из 4-х способов: на p горизонтальных полос, на p вертикальных полос, на сетку размера p на p , совсем не разбивать на части. Тогда, в зависимости от способов распределения матриц, получается $4^3 = 64$ варианта решения задачи.

Задачи, связанные с вычислениями в узлах сеток и решеток, решение краевых задач относятся к числу задач, для которых технология параллельного программирования также хорошо разработана [28].

Для иллюстрации рассмотрим численное решение двумерной краевой задачи для уравнения теплопроводности методом простой итерации с использованием явного итерационного алгоритма. Перед началом вычислений задаются гранич-

ные условия (значения на границе плоской области – прямоугольной сетки) и начальные значения во внутренних точках в начальный момент времени.

На каждом этапе вычислений определяются приращения значений функции во внутренних точках сетки в предположении, что новые значения функций равны одной четверти от суммы значений в 4-х соседних точках.

1.8 Выводы из обзора

На основании анализа литературы сформулируем следующие основные требования к будущему программному средству:

1. Система должна применять алгоритмы обработки изображений для улучшения их качества (п. 2.1. – 2.5.).
2. Система должна находить объекты на соседних снимках с помощью алгоритмов сравнения скелетов объектов (п. 3.1. – 3.2., 4.1 – 4.2.).
3. Система должна определять gps-координаты каждого объекта на снимке (п. 5.1. – 5.2.).
4. Система должна определять расстояние до каждого объекта на снимке с помощью используемой математической модели камеры с учётом известной высоты камеры над уровнем моря и её оптических характеристик (эффективное фокусное расстояние, угол наклона к горизонту) (п. 6.).
5. Система должна строить горизонтальную проекцию каждого полученного снимка (п. 7.).
6. Система должна быть многопоточной: выполнять алгоритмы улучшения качества изображений и сравнения объектов с помощью параллельных потоков (п. 8.1. – 8.3.).

Глава 2 Анализ и построение модели ПО и решаемых задач

2.1 Анализ ПО и решаемых задач

Профессионалом данной предметной области является человек, решающий перечисленные далее задачи.

- 1) Поиск одинаковых объектов на снимках.
- 2) Расчёт следующих характеристик каждого объекта по известным оптическим и пространственным параметрам камеры: расстояние, gps-координаты, скорость, размеры, траектория.
- 3) Составление плана местности каждого снимка с отмеченными объектами.

Для решения перечисленных задач имеется стационарная поворотная камера и созданные ею цифровые снимки акваторий. Для камеры известны оптические и пространственные характеристики.

Таким образом, выделены следующие задачи.

- 1) Автоматизировать анализ снимков и расчёты полученных характеристик объектов.
- 2) В случае ошибок выводить информативные сообщения об ошибках.
- 3) Программа должна строить ортогональную проекцию выбранного изображения.

Перечислим выделенные объекты и их свойства.

- Объект *Камера* имеет указанные далее характеристики
 - оптические характеристики
 - эффективное фокусное расстояние (м) $(0, \infty)$
 - горизонтальный угол обзора (градус) $(0, 180)$
 - вертикальный угол обзора (градус) $(0, 180)$
 - пространственные характеристики
 - высота над уровнем моря (м) $[0, \infty)$
 - gps-координаты

- широта (градус) $[-90; 90]$
- долгота (градус) $[0; 180]$
- Объект *Снимок* имеет указанные далее характеристики
 - пиксельные размеры
 - ширина (пиксель) $[352, 10320]$
 - высота (пиксель) $[288, 8956]$
 - время создания
 - год $[2000, 3000]$
 - месяц $[1, 12]$
 - день $[1, 31]$
 - час $[0, 23]$
 - минута $[0, 59]$
 - секунда $[0, 59]$
 - наклон (градус) $[0, 180]$
 - азимут (градус) $[0, 359]$
 - крен (градус) $[-180, 180]$
 - объект снимка (идентификаторы) $[0, \infty)$
 - ортогональная проекция (изображение)
 - направляющие косинусы $m_{ij}, i, j = 1..3$ $(-\infty, \infty)$
 - коэффициенты прямого линейного преобразования $L, L_{ij}, i, j = 1..12$ $(-\infty, \infty)$

— Объект *Объект на снимке* имеет указанные далее характеристики

- размеры
 - длина (м) $[0, \infty)$
 - высота (м) $[0, \infty)$
 - ширина (м) $[0, \infty)$
- расстояние до камеры (м) $(0, \infty)$
- расстояние до объекта по горизонтали (м) $(0, \infty)$
- расстояние до объекта по длине (м) $(0, \infty)$

- gps-координаты – последовательность
 - gps-координаты на снимке
 - широта (градус) $[-90; 90]$
 - долгота (градус) $[0; 180]$
- скорость на снимке (м/с) $(0, \infty)$
- траектория () последовательность координат на снимке
- пиксельные координаты – последовательность
 - пиксель
 - x-координата
 - y-координата
- Объект *Модель Земли* имеет указанные далее характеристики
 - большая полуось эллипсоида $a = 6\,378\,136$ (м) [31]
 - малая полуось эллипсоида b
 - коэффициент сжатия эллипсоида $f = 1/298,257\,84$
 - третий коэффициент сжатия $(0, \infty)$ n
 - первый эксцентриситет $(0, \infty)$ e
 - второй эксцентриситет $(0, \infty)$ e'
 - точка на экваторе
 - азимут точки на экваторе с направлением на начальную точку α_0
 - сферическая длина дуги от точки на экваторе до начальной точки σ_1
 - расстояние от точки на экваторе до начальной точки s_1
 - расстояние от точки на экваторе до конечной точки s_2
 - сферическая длина дуги от точки на экваторе до конечной точки σ_2
 - расстояние от начальной точки до конечной s_{12}
 - начальная точка
 - азимут начальной точки по направлению вдоль параллели $[-90; 90]$ α_1

- широта начальной точки $[-90; 90]$ φ_1
- сферическая широта начальной точки β_1
- сферическая долгота начальной точки ω_1
- долгота начальной точки λ_1
- конечная точка
 - азимут конечной точки по направлению вдоль параллели α_2
 - сферическая широта конечной точки β_2
 - широта конечной точки φ_2
 - долгота конечной точки λ_2
 - сферическая долгота конечной точки ω_2
- параметр растяжения k
 - параметр растяжения ε
- коэффициенты преобразования Лагранжа $C_{11} \dots C_{16}$
- коэффициент A_1
- коэффициент τ

Для всех перечисленных выше задач **входными данными** являются группа последовательных снимков, оптические и пространственные характеристики камеры, а **выходными** - скорость, ускорение, траектория, размер объекта либо сообщение о невозможности анализа (несоответствие снимков друг другу).

2.2 Модель предметной области

Запишем модель предметной области с использованием языка прикладной логики [32 ; 33].

сорт объекты: $\{ \}N$ каждый объект имеет уникальный идентификатор

сорт снимки: $\{ \}N$ каждый снимок имеет уникальный идентификатор

сорт ортогональное изображение: $\{ \}N$ каждое ортогональное изображение имеет уникальный идентификатор

сорт камера: $\{ \}N$

длина $\equiv \mathbb{R} (0, \infty)$

ширина $\equiv \mathbb{R} (0, \infty)$

высота $\equiv \mathbb{R} (0, \infty)$

размеры $\equiv (\text{х длина, ширина, высота})$

сорт размеры объекта: объекты \rightarrow размеры

сорт последовательность скоростей объекта: объекты $\rightarrow \text{seq } \mathbb{R}[0, \infty)$

координата $\equiv (\text{х } \mathbb{I}[0, 10320], \mathbb{I}[0, 8956])$

сорт координаты объекта на снимке: объекты $\rightarrow \text{seq}$ координата

сорт траектория: объекты $\rightarrow \text{seq}$ координаты объекта на снимке

gps-координаты $\equiv (\text{х широта } \rightarrow \mathbb{R}[0,180], \text{ долгота } \rightarrow \mathbb{R}[0,180], \text{ высота } \rightarrow \mathbb{R}[0, \infty))$

сорт gps-координаты объекта: объекты $\rightarrow \text{seq}$ gps-координаты объекта на снимке

цифровая ширина $\equiv \mathbb{I}[352, 10320]$

цифровая высота $\equiv \mathbb{I}[288, 8956]$

размеры снимка $\equiv (\times \text{ цифровая ширина, цифровая высота})$

сорт пиксельные размеры: снимки \rightarrow размеры снимка

год $\equiv \mathbb{I}[2000, 3000]$

месяц $\equiv \mathbb{I}[1, 12]$

день $\equiv \mathbb{I}[1, 31]$

час $\equiv \mathbb{I}[0, 23]$

минута $\equiv \mathbb{I}[0, 59]$

секунда $\equiv \mathbb{I}[0, 59]$

время $\equiv (\text{х год, месяц, день, час, минута, секунда})$

сорт время получения снимка: снимки \rightarrow время

сорт камера снимка: снимки \rightarrow камера

сорт объект снимка: снимки $\rightarrow \{\}$ объекты

сорт число объектов снимка: снимки $\rightarrow I [0, 255]$
сорт азимут: снимки \rightarrow угол к направлению на север
сорт угол наклона к горизонту: снимки \rightarrow угол
сорт содержание льда: снимки $\rightarrow I[0, \infty)$
сорт τ : снимки \rightarrow наклон
сорт φ : снимки \rightarrow азимут
сорт σ : снимки \rightarrow крен
сорт m_{11} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{12} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{13} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{21} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{22} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{23} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{31} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{32} : снимки $\rightarrow R(-\infty, \infty)$
сорт m_{33} : снимки $\rightarrow R(-\infty, \infty)$
сорт L : снимки $\rightarrow R(-\infty, \infty)$
сорт L_1 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_2 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_3 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_4 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_5 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_6 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_7 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_8 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_9 : снимки $\rightarrow R(-\infty, \infty)$
сорт L_{10} : снимки $\rightarrow R(-\infty, \infty)$
сорт L_{11} : снимки $\rightarrow R(-\infty, \infty)$
сорт L_{12} : снимки $\rightarrow R(-\infty, \infty)$

сорт модель Земли: $\{\}N$

сорт большая полуось эллипсоида a : модель Земли $\rightarrow I[6\,300\,000, 6\,400\,000]$

сорт коэффициент сжатия эллипсоида f : модель Земли $\rightarrow R(0, 1)$

сорт малая полуось эллипсоида b : модель Земли $\rightarrow I[635\,000, 636\,000]$

сорт третий коэффициент сжатия n : (model \rightarrow модель Земли, $n \rightarrow$ коэффициент сжатия эллипсоида (model)) $\rightarrow R(0, \infty)$

сорт первый эксцентриситет e : (model \rightarrow модель Земли, $f \rightarrow$ коэффициент сжатия эллипсоида (model)) $\rightarrow R(0, \infty)$

сорт второй эксцентриситет e' : (model \rightarrow модель Земли, $f \rightarrow$ коэффициент сжатия эллипсоида (model), $e \rightarrow$ первый эксцентриситет (model, f)) $\rightarrow R(0, \infty)$

сорт точка на экваторе: модель Земли $\rightarrow \{\}N$

сорт азимут точки на экваторе с направлением на начальную точку α_0 : (model \rightarrow модель Земли, $E \rightarrow$ точка на экваторе (model), $A \rightarrow$ начальная точка (model), $\beta_1 \rightarrow$ сферическая широта начальной точки (model, A)) $\rightarrow R[-90; 90]$

сорт сферическая длина дуги от точки на экваторе до начальной точки σ_1 : (модель Земли, точка на экваторе, начальная точка) $\rightarrow R[0; 360]$

сорт расстояние от точки на экваторе до начальной точки s_1 : (model \rightarrow модель Земли, $E \rightarrow$ точка на экваторе, $A \rightarrow$ начальная точка) $\rightarrow I(0; \infty)$

сорт расстояние от начальной точки до конечной s_{12} : (модель Земли, начальная точка, конечная точка) $\rightarrow I(0, \infty)$

сорт расстояние от точки на экваторе до конечной точки s_2 : (расстояние от точки на экваторе до начальной точки s_1 , расстояние от начальной точки до конечной s_{12}) $\rightarrow I(0; \infty)$

сорт сферическая длина дуги от точки на экваторе до конечной точки σ_2 : (модель Земли, точка на экваторе, начальная точка, конечная точка) $\rightarrow R[0; 360]$

сорт начальная точка: модель Земли $\rightarrow \{\}N$

сорт азимут начальной точки по направлению вдоль параллели α_1 : начальная точка $\rightarrow R[-90; 90]$

сорт широта начальной точки φ_1 : начальная точка $\rightarrow R[-90; 90]$

сорт сферическая широта начальной точки β_1 : (модель Земли, начальная точка) $\rightarrow R[0; 360]$

сорт сферическая долгота начальной точки ω_1 : (модель Земли, начальная точка, азимут точки на экваторе с направлением на начальную точку, сферическая длина дуги от точки на экваторе до начальной точки) $\rightarrow R[0; 360]$

сорт долгота начальной точки λ_1 : (модель Земли, начальная точка) $\rightarrow R[0; 180]$

сорт конечная точка: модель Земли $\rightarrow \{ \} N$

сорт азимут конечной точки по направлению вдоль параллели α_2 : (модель Земли, точка на экваторе, конечная точка, азимут точки на экваторе с направлением на начальную точку) $\rightarrow R[-90; 90]$

сорт сферическая широта конечной точки β_2 : (модель Земли, конечная точка, азимут точки на экваторе с направлением на начальную точку, сферическая длина дуги от точки на экваторе до конечной точки) $\rightarrow R[0; 360]$

сорт широта конечной точки φ_2 : (модель Земли, конечная точка) $\rightarrow R[-90; 90]$

сорт долгота конечной точки λ_2 : (модель Земли, конечная точка, азимут точки на экваторе с направлением на начальную точку, сферическая длина дуги от точки на экваторе до конечной точки) $\rightarrow R[0; 180]$

сорт сферическая долгота конечной точки ω_2 : (модель Земли, конечная точка, азимут точки на экваторе с направлением на начальную точку) $\rightarrow R[0; 360]$

сорт параметр растяжения k : (модель Земли, второй эксцентриситет) $\rightarrow R(0; \infty)$

сорт параметр растяжения ε : модель Земли $\rightarrow R(0; \infty)$

сорт коэффициенты преобразования Лагранжа $C_{11} \dots C_{16}$: модель Земли $\rightarrow \{\} R[0; 360]$

сорт коэффициент A_1 : модель Земли $\rightarrow R[0; 360]$

сорт коэффициент τ : модель Земли $\rightarrow R[0; 360]$

сорт координата объекта на снимке: ($\text{img} \rightarrow \text{снимки}$, $\text{obj} \rightarrow \text{объект снимка}$ (img)) \rightarrow координата

сорт gps-координаты объекта на снимке: ($\text{model} \rightarrow \text{модель Земли}$, $\text{img} \rightarrow \text{снимки}$, $\text{obj} \rightarrow \text{объект снимка}$ (img)) \rightarrow gps-координаты

сорт расстояние до камеры: ($\text{img} \rightarrow \text{снимки}$, $\text{obj} \rightarrow \text{объект снимка}$ (img)) $\rightarrow R[0, \infty)$

сорт расстояние до объекта по горизонтали: ($\text{img} \rightarrow \text{снимки}$, $\text{obj} \rightarrow \text{объект снимка}$ (img)) $\rightarrow R [0, \infty)$

сорт расстояние до объекта по длине: ($\text{img} \rightarrow \text{снимки}$, $\text{obj} \rightarrow \text{объект снимка}$ (img)) $\rightarrow R [0, \infty)$

сорт ортогональная проекция изображения: снимки \rightarrow ортогональное изображение

горизонтальный угол обзора $\equiv I(0, 360]$

вертикальный угол обзора $\equiv I(0, 360]$

наклон $\equiv I[0, 180]$

азимут $\equiv I[0, 360]$

крен $\equiv I[0, 90]$

высота над уровнем моря $\equiv R [0, \infty)$

эффективное фокусное расстояние $\equiv R(0, \infty)$

оптические характеристики камеры \equiv (х горизонтальный угол обзора \rightarrow угол, вертикальный угол обзора \rightarrow угол, эффективное фокусное расстояние)

пространственные характеристики камеры \equiv (х высота камеры, gps-координаты)

сорт высота камеры: камера \rightarrow высота над уровнем моря

сорт характеристики камеры: камера \rightarrow (х оптические характеристики камеры, пространственные характеристики камеры)

2.2.1 Онтологические соглашения

(img: снимки) (obj1: объекты) (obj2: объекты) (obj1 \neq obj2) (x: координаты объекта на снимке (obj1)) (y: координаты объекта на снимке (obj2)) $\Rightarrow x \neq y$

(img: снимки) (obj1: объекты) (obj2: объекты) (obj1 \neq obj2) \Rightarrow gps-координаты объекта на снимке (img, obj1) \neq gps-координаты объекта на снимке (img, obj2)

(img: снимки) (obj1: объект снимка (img)) (obj2: объект снимка (img)) (x1: расстояние до объекта по горизонтали (img, obj1)) (x2: расстояние до объекта по горизонтали (img, obj2)) (y1: расстояние до объекта по длине(img, obj1)) (y2: расстояние до объекта по длине(img, obj2)) (obj1 \neq obj2) $\Rightarrow (\times x1, y1) \neq (\times x2, y2)$

(obj: объекты) (t: траектория (obj)) $\Rightarrow t \neq \emptyset$

(imgs: последовательность снимков) (n: $I[2, \text{length}(\text{imgs})]$) (obj: объекты)
скорость (imgs, n, obj) = (пройденное объектом расстояние ($\pi(n-1, \text{imgs})$, $\pi(n, \text{imgs})$, obj) / (время получения снимка ($\pi(n, \text{imgs})$) - время получения снимка ($\pi(n-1, \text{imgs})$)))

(img1, img2: снимки) (obj: объекты) **пройденное объектом расстояние**
\sqrt{(\text{расстояние до камеры (img1, obj)}^2 + \text{расстояние до камеры (img2, obj)}^2 - 2 * \text{расстояние до камеры (img1, obj)} * \text{расстояние до камеры (img2, obj)} * \cos(\text{угол положения камеры относительно направления на север (img2) - угол положения камеры относительно направления на север (img1)}))}

(img: снимки) (obj: объекты) (cam: камера) **расстояние до камеры** (img, obj) =
$$\sqrt{\text{высота камеры(cam)}^2 + \text{расстояние до объекта по длине(img, obj)}^2 + \text{расстояние до объекта по горизонтали(img, obj)}^2}$$

(cam: камера) (model: модель Земли) (obj: объект (gps_obj)) (gps_cam: gps-координаты (cam)) (φ_cam: широта (gps_cam)) (λ_cam: долгота(gps_cam)) (h_cam: высота(gps_cam)) (gps_obj: gps-координаты (obj)) (φ_obj: широта (gps_obj)) (λ_obj: долгота(gps_obj)) (h_obj: высота(gps_obj)) (a: большая полуось эллипсоида) (f: коэффициент сжатия эллипсоида) gps-координаты объекта на снимке (model, img, obj) = (× широта конечной точки ((model: модель Земли) (E: точка на экваторе) (A: начальная точка) (α₁: азимут начальной точки по направлению вдоль параллели) (α₀: азимут точки на экваторе с направлением на начальную точку) (φ₁: широта начальной точки) (β₁: сферическая широта начальной точки) (σ₁: сферическая длина дуги от точки на экваторе до начальной точки (model, E, A)) → (E = точка на экваторе(model)) & (A = начальная точка (model)) & (α₁ = азимут начальной точки по направлению вдоль параллели (A)) & (α₀ = азимут точки на экваторе с направлением на начальную точку (model, E, A, α₁)) & (φ₁ = широта начальной точки (A)) & (β₁ = сферическая широта начальной точки (model, A, φ₁))), долгота конечной точки())

(model: модель Земли) (E: точка на экваторе) (A: начальная точка) (α₁: азимут начальной точки по направлению вдоль параллели) (α₀: азимут точки на экваторе с направлением на начальную точку) (φ₁: широта начальной точки) (f: коэффициент сжатия эллипсоида) (β₁: сферическая широта начальной точки) (σ₁: сфе-

рическая длина дуги от точки на экваторе до начальной точки) (b: малая полуось эллипсоида) (s_1 : расстояние от точки на экваторе до начальной точки (σ_1, b)) \rightarrow (E = точка на экваторе(model)) & (A = начальная точка (model)) & (α_1 = азимут начальной точки по направлению вдоль параллели (A)) & (α_0 = азимут точки на экваторе с направлением на начальную точку (model, E, A, α_1)) & (φ_1 = широта начальной точки (A)) & (f = коэффициент сжатия эллипсоида (model)) & (β_1 = сферическая широта начальной точки (model, A, φ_1)) & (σ_1 = сферическая длина дуги от точки на экваторе до начальной точки (α_0, β_1)) & (b = малая полуось эллипсоида (a, f)) (model: модель Земли) (A: начальная точка) (B: конечная точка) (s_{12} : расстояние от начальной точки до конечной (A, B)) \rightarrow (A = начальная точка (model)) & (B = конечная точка (model)) (model: модель Земли) (E: точка на экваторе) (b: малая полуось эллипсоида) (α_0 : азимут точки на экваторе с направлением на начальную точку) (A: начальная точка) (α_1 : азимут начальной точки по направлению вдоль параллели) (φ_1 : широта начальной точки) (f: коэффициент сжатия эллипсоида) (β_1 : сферическая широта начальной точки) (σ_1 : сферическая длина дуги от точки на экваторе до начальной точки) (s_1 : расстояние от точки на экваторе до начальной точки) (B: конечная точка) (s_{12} : расстояние от начальной точки до конечной) (s_2 : расстояние от точки на экваторе до конечной точки (s_1, s_{12})) \rightarrow (E = точка на экваторе(model)) & (A = начальная точка (model)) & (α_1 = азимут начальной точки по направлению вдоль параллели (A)) & (α_0 = азимут точки на экваторе с направлением на начальную точку (model, E, A, α_1)) & (φ_1 = широта начальной точки (A)) & (β_1 = сферическая широта начальной точки (model, A, φ_1)) & (σ_1 = сферическая длина дуги от точки на экваторе до начальной точки (α_0, β_1)) & (b = малая полуось эллипсоида (a, f)) & (s_1 = расстояние от точки на экваторе до начальной точки (σ_1, b))

(model: модель Земли) (E: точка на экваторе) (τ : коэффициент τ) (C: коэффициенты преобразования Лагранжа $C_{11} \dots C_{16}$) (B: конечная точка) (b: малая полуось эллипсоида) (α_0 : азимут точки на экваторе с направлением на начальную точку) (A: начальная точка) (α_1 : азимут начальной точки по направлению вдоль

параллели) (φ_1 : широта начальной точки) (f : коэффициент сжатия эллипсоида) (β_1 : сферическая широта начальной точки) (σ_1 : сферическая длина дуги от точки на экваторе до начальной точки) (s_1 : расстояние от точки на экваторе до начальной точки) (B : конечная точка) (s_{12} : расстояние от начальной точки до конечной) (s_2 : расстояние от точки на экваторе до конечной точки) (σ_2 : сферическая длина дуги от точки на экваторе до конечной точки) ($model, E, A, B$) \rightarrow (E = точка на экваторе($model$)) & (A = начальная точка ($model$)) & (α_1 = азимут начальной точки по направлению вдоль параллели (A)) & (α_0 = азимут точки на экваторе с направлением на начальную точку ($model, E, A, \alpha_1$)) & (φ_1 = широта начальной точки (A)) & (β_1 = сферическая широта начальной точки ($model, A, \varphi_1$)) & (σ_1 = сферическая длина дуги от точки на экваторе до начальной точки (α_0, β_1)) & (b = малая полуось эллипсоида (a, f)) & (s_1 = расстояние от точки на экваторе до начальной точки (σ_1, b)) & (s_2 = расстояние от точки на экваторе до конечной точки (s_1, s_{12}))

($model$: модель Земли) (A : начальная точка) (f : коэффициент сжатия эллипсоида) (φ_1 : широта начальной точки) (β_1 : сферическая широта начальной точки($model, A$)) \rightarrow (A = начальная точка ($model$)) & (f = коэффициент сжатия эллипсоида ($model$)) & (φ_1 = широта начальной точки (A))

($model$: модель Земли) (E : точка на экваторе) (A : начальная точка) (f : коэффициент сжатия эллипсоида) (α_0 : азимут точки на экваторе с направлением на начальную точку) (α_1 : азимут начальной точки по направлению вдоль параллели) (φ_1 : широта начальной точки) (β_1 : сферическая широта начальной точки($model, A$)) (ω_1 : сферическая долгота начальной точки ($model, E, A, \sigma_1$)) \rightarrow (A = начальная точка ($model$)) & (f = коэффициент сжатия эллипсоида ($model$)) & (φ_1 = широта начальной точки (A)) (E = точка на экваторе($model$)) & (A = начальная точка ($model$)) & (α_1 = азимут начальной точки по направлению вдоль параллели (A)) & (α_0 = азимут точки на экваторе с направлением на начальную точку ($model, E, A, \alpha_1$)) & (φ_1 = широта начальной точки (A)) & (β_1 = сферическая широта начальной точки ($model, A, \varphi_1$)) & (σ_1 = сферическая длина дуги от точки на экваторе до начальной точки($model, E, A$))

(model: модель Земли) (E: точка на экваторе) (A: начальная точка) (B: конечная точка) (α_1 : азимут начальной точки по направлению вдоль параллели) (α_0 : азимут точки на экваторе с направлением на начальную точку) (σ_2 : сферическая длина дуги от точки на экваторе до конечной точки) (α_2 : азимут конечной точки по направлению вдоль параллели (model, B, α_0 , σ_2)) \Rightarrow (E = точка на экваторе (model)) & (B = конечная точка (model)) & (A = начальная точка(model)) & (α_1 = азимут начальной точки по направлению вдоль параллели (A)) (α_0 = азимут точки на экваторе с направлением на начальную точку (model, E, A, α_1))

(model: модель Земли) (E: точка на экваторе) (A: начальная точка) (b: малая полуось эллипсоида) (α_0 : азимут точки на экваторе с направлением на начальную точку) (α_1 : азимут начальной точки по направлению вдоль параллели) (φ_1 : широта начальной точки) (f: коэффициент сжатия эллипсоида) (β_1 : сферическая широта начальной точки) (σ_1 : сферическая длина дуги от точки на экваторе до начальной точки) (s_1 : расстояние от точки на экваторе до начальной точки) (τ : коэффициент τ) (C: коэффициенты преобразования Лагранжа $C_{11} \dots C_{16}$) (B: конечная точка) (s_{12} : расстояние от начальной точки до конечной) (s_2 : расстояние от точки на экваторе до конечной точки) (σ_2 : сферическая длина дуги от точки на экваторе до конечной точки) (β_2 : сферическая широта конечной точки (model, B, α_0 , σ_2)) \rightarrow (E = точка на экваторе(model)) & (A = начальная точка (model)) & (α_1 = азимут начальной точки по направлению вдоль параллели (A)) & (α_0 = азимут точки на экваторе с направлением на начальную точку (model, E, A, α_1)) & (φ_1 = широта начальной точки (A)) & (β_1 = сферическая широта начальной точки (model, A, φ_1)) & (σ_1 = сферическая длина дуги от точки на экваторе до начальной точки (α_0 , β_1)) & (b = малая полуось эллипсоида (a, f)) & (s_1 = расстояние от точки на экваторе до начальной точки (σ_1 , b)) & (s_2 = расстояние от точки на экваторе до конечной точки (s_1 , s_{12})) (σ_2 = сферическая длина дуги от точки на экваторе до конечной точки (model, E, A, B))

(model: модель Земли) (E: точка на экваторе) (A: начальная точка) (b: малая полуось эллипсоида) (α_0 : азимут точки на экваторе с направлением на начальную точку) (α_1 : азимут начальной точки по направлению вдоль параллели) (φ_1 : широта начальной точки) (f: коэффициент сжатия эллипсоида) (β_1 : сферическая широта начальной точки) (σ_1 : сферическая длина дуги от точки на экваторе до начальной точки) (s_1 : расстояние от точки на экваторе до начальной точки) (τ : коэффициент τ) (C: коэффициенты преобразования Лагранжа $C'_{11} \dots C'_{16}$) (B: конечная точка) (s_{12} : расстояние от начальной точки до конечной) (s_2 : расстояние от точки на экваторе до конечной точки) (σ_2 : сферическая длина дуги от точки на экваторе до конечной точки) (β_2 : сферическая широта конечной точки) \rightarrow (E = точка на экваторе(model)) & (A = начальная точка(model)) & (α_1 = азимут начальной точки по направлению вдоль параллели (A)) & (α_0 = азимут точки на экваторе с направлением на начальную точку(model, E, A, α_1)) & (φ_1 = широта начальной точки (A)) & (β_1 = сферическая широта начальной точки(model, A, φ_1)) & (σ_1 = сферическая длина дуги от точки на экваторе до начальной точки(α_0 , β_1)) & (b = малая полуось эллипсоида(a, f)) & (s_1 = расстояние от точки на экваторе до начальной точки(σ_1 , b)) & (s_2 = расстояние от точки на экваторе до конечной точки(s_1 , s_{12})) (σ_2 = сферическая длина дуги от точки на экваторе до конечной точки(model, E, A, B)) (β_2 = сферическая широта конечной точки(model, B, α_0 , σ_2))

(model: модель Земли) (E: точка на экваторе) (τ : коэффициент τ) (C: коэффициенты преобразования Лагранжа $C'_{11} \dots C'_{16}$) (b: малая полуось эллипсоида) (α_0 : азимут точки на экваторе с направлением на начальную точку) (A: начальная точка) (α_1 : азимут начальной точки по направлению вдоль параллели) (φ_1 : широта начальной точки) (f: коэффициент сжатия эллипсоида) (β_1 : сферическая широта начальной точки) (σ_1 : сферическая длина дуги от точки на экваторе до начальной точки) (s_1 : расстояние от точки на экваторе до начальной точки) (B: конечная точка) (s_{12} : расстояние от начальной точки до конечной) (s_2 : расстояние от точки на экваторе до конечной точки) (σ_2 : сферическая длина дуги от точки на экваторе до

конечной точки) (s_2 : расстояние от точки на экваторе до конечной точки (s_1, s_{12}))
 \rightarrow (E = точка на экваторе(model)) & (A = начальная точка (model)) & (α_1 = азимут
 начальной точки по направлению вдоль параллели (A)) & (α_0 = азимут точки на
 экваторе с направлением на начальную точку (model, E, A, α_1)) & (φ_1 = широта
 начальной точки (A)) & (β_1 = сферическая широта начальной точки (model, A, φ_1))
 & (σ_1 = сферическая длина дуги от точки на экваторе до начальной точки (α_0, β_1))
 & (b = малая полуось эллипсоида (a, f)) & (s_1 = расстояние от точки на экваторе до
 начальной точки (σ_1, b)) & (s_2 = расстояние от точки на экваторе до конечной точ-
 ки (s_1, s_{12})) & (σ_2 = сферическая длина дуги от точки на экваторе до конечной точ-
 ки (model, E, A, B))

(model: модель Земли) (E: точка на экваторе) (τ : коэффициент τ) (C: коэф-
 фициенты преобразования Лагранжа $C_{11} \dots C_{16}$) (b: малая полуось эллипсоида)
 (α_0 : азимут точки на экваторе с направлением на начальную точку) (A: начальная
 точка) (α_1 : азимут начальной точки по направлению вдоль параллели) (φ_1 : широта
 начальной точки) (f: коэффициент сжатия эллипсоида) (β_1 : сферическая широта
 начальной точки) (σ_1 : сферическая длина дуги от точки на экваторе до начальной
 точки) (s_1 : расстояние от точки на экваторе до начальной точки) (B: конечная точ-
 ка) (s_{12} : расстояние от начальной точки до конечной) (s_2 : расстояние от точки на
 экваторе до конечной точки) (σ_2 : сферическая длина дуги от точки на экваторе до
 конечной точки) (s_2 : расстояние от точки на экваторе до конечной точки) (ω_2 :
 сферическая долгота конечной точки (model, B, α_0)) \rightarrow (E = точка на
 экваторе(model)) & (A = начальная точка (model)) & (α_1 = азимут начальной точки
 по направлению вдоль параллели (A)) & (α_0 = азимут точки на экваторе с направ-
 лением на начальную точку (model, E, A, α_1)) & (φ_1 = широта начальной точки
 (A)) & (β_1 = сферическая широта начальной точки (model, A, φ_1)) & (σ_1 = сфери-
 ческая длина дуги от точки на экваторе до начальной точки (α_0, β_1)) & (b = малая по-
 луось эллипсоида (a, f)) & (s_1 = расстояние от точки на экваторе до начальной
 точки (σ_1, b)) & (s_2 = расстояние от точки на экваторе до конечной точки (s_1, s_{12}))

& (σ_2 = сферическая длина дуги от точки на экваторе до конечной точки (model, E, A, B)) & (s_2 = расстояние от точки на экваторе до конечной точки (s_1, s_{12}))

2.3 Постановка прикладных задач

2.3.1.1 Задача определения характеристик объектов

Дано:

- последовательность снимков
- высота камеры
- оптические характеристики камеры

Найти:

- число объектов
- скорость каждого объекта
- расстояние до каждого объекта
- траекторию каждого объекта на каждом снимке
- географические координаты каждого объекта

Алгоритм решения задачи:

- 1) Пользователь загружает в систему снимки для анализа.
- 2) Если снимки удовлетворяют условиям (для каждого снимка, кроме одного, должен существовать снимок, созданный не позже, чем через 30 секунд с момента создания первого; в последовательности таких снимков угол камеры относительно направления на север должен меняться в одну и ту же сторону), то программная система применяет методы улучшения изображений снимков. Иначе система выдаёт сообщение о невозможности произвести анализ снимков.
- 3) Система определяет число объектов.
- 4) Система определяет приблизительные размеры объектов.
- 5) Система определяет скорость каждого объекта.
- 6) Система определяет траекторию каждого объекта.

- 7) Система определяет GPS-координаты каждого объекта.
- 8) Система выводит полученные данные.

2.3.1.2 Задача построения ортогональной проекции изображения

Дано:

- последовательность снимков
- высота камеры
- оптические характеристики камеры

Найти:

- ортогональную проекцию местности на каждом снимке

Алгоритм решения задачи:

- 1) Пользователь загружает в систему снимки для анализа.
- 2) Система строит ортогональную проекцию указанного пользователем изображения.
- 3) Система выводит полученный результат.

Направляющие косинусы:

$$\alpha_0 = ph(|\cos \alpha + i \sin \alpha \sin \beta| + i \sin \alpha \cos \beta) \quad (1)$$

$$m_{11} = \cos \varphi \cos \sigma + \cos \varphi \cos \tau \sin \sigma \quad (2)$$

$$m_{12} = -\sin \varphi \cos \sigma + \cos \varphi \cos \tau \sin \sigma \quad (3)$$

$$m_{13} = \sin \tau \cos \sigma \quad (4)$$

$$m_{21} = -\cos \varphi \sin \sigma + \sin \varphi \cos \tau \cos \sigma \quad (5)$$

$$m_{22} = \sin \varphi \sin \sigma + \cos \varphi \cos \tau \cos \sigma \quad (6)$$

$$m_{23} = \sin \tau \cos \sigma \quad (7)$$

$$m_{31} = \sin \varphi \sin \tau \quad (8)$$

$$m_{32} = \cos \varphi \sin \tau \quad (9)$$

$$m_{33} = -\cos \tau \quad (10)$$

Коэффициенты прямого линейного преобразования:

$$L = -(x_c m_{31} + y_c m_{32} + z_c m_{33}) \quad (11)$$

$$L_1 = (u_0 m_{31} + f m_{11}) / (\lambda_u L) \quad (12)$$

$$L_2 = (u_0 m_{32} + f m_{12}) / (\lambda_u L) \quad (13)$$

$$L_3 = (u_0 m_{33} + f m_{13}) / (\lambda_u L) \quad (14)$$

$$L_4 = -(L_1 x_c + L_2 y_c + L_3 z_c) \quad (15)$$

$$L_5 = m_{31} / L \quad (16)$$

$$L_6 = m_{32} / L \quad (17)$$

$$L_7 = m_{33} / L \quad (18)$$

$$L_8 = (u_0 m_{31} + f m_{21}) / (\lambda_v L) \quad (19)$$

$$L_9 = (u_0 m_{32} + f m_{22}) / (\lambda_v L) \quad (20)$$

$$L_{10} = (u_0 m_{33} + f m_{23}) / (\lambda_v L) \quad (21)$$

$$L_{11} = -(L_8 x_c + L_9 y_c + L_{10} z_c) \quad (22)$$

Связь координат изображения с мировыми координатами

$$u = \frac{L_1 x + L_2 y + L_3 z + L_4}{L_9 x + L_{10} y + L_{11} z + 1} \quad (23)$$

$$v = \frac{L_5 x + L_6 y + L_7 z + L_8}{L_9 x + L_{10} y + L_{11} z + 1} \quad (24)$$

Матрица связи мировых координат (x, y, z) с координатами снимка (u, v)

$$\begin{bmatrix} [L_1 - L_9 u] & [L_2 - L_{10} u] & [L_3 - L_{11} u] \\ [L_5 - L_9 v] & [L_6 - L_{10} v] & [L_7 - L_{11} v] \\ a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} [u - L_4] \\ [v - L_8] \\ d \end{bmatrix} \quad (25)$$

расстояние до объекта по длине (26)

$$= \frac{v - L_8 + ((L_9 v - L_5) \frac{u - L_4 - (L_3 - L_{11} u)}{(L_1 - L_9 u)} + (L_{11} v - L_7)) z}{L_6 - L_{10} v + \frac{(L_9 v - L_5)(L_2 - L_{10} u)}{L_1 - L_9 u}}$$

расстояние до объекта по горизонтали (27)

$$= \frac{u - L_4 - (L_2 - L_{10} u) y - (L_3 - L_{11} u) z}{L_1 - L_9 u}$$

Где: u, v – координаты пикселя снимка.

$$x^0 = -f \frac{xm_{11} + ym_{12} + fm_{13}}{xm_{31} + ym_{32} + fm_{33}} \quad (28)$$

$$y^0 = -f \frac{xm_{21} + ym_{22} + fm_{23}}{xm_{31} + ym_{32} + fm_{33}} \quad (29)$$

Где: f – фокусное расстояние.

$$\alpha_0 = ph(|\cos \alpha + i \sin \alpha \sin \beta| + i \sin \alpha \cos \beta) \quad (30)$$

$$\sigma = ph(\cos \alpha \cos \beta + i \sin \beta) \quad (31)$$

$$\omega = ph(\cos \sigma + i \sin \alpha_0 \sin \sigma) \quad (32)$$

$$\frac{s}{b} = \int_0^\sigma \sqrt{1 + k^2 (\sin \sigma)^2 \sigma'} d\sigma' = I_1(\sigma) \quad (33)$$

$$\begin{aligned} \lambda &= \omega - f \sin \alpha_0 \int_0^\sigma \frac{2 - f}{1 + (1 - f) \sqrt{\sqrt{1 + k^2 (\sin \sigma)^2 \sigma'}}} d\sigma' \\ &= \omega - f \sin \alpha_0 I_3(\sigma) \end{aligned} \quad (34)$$

$$I_1(\sigma) = A_1 \left(\sigma + \sum_{l=1}^{\infty} C_{1l} \sin 2l\sigma \right) \quad (35)$$

$$I_3(\sigma) = A_3 \left(\sigma + \sum_{l=1}^{\infty} C_{3l} \sin 2l\sigma \right) \quad (36)$$

$$\sigma = \tau + \sum_{l=1}^{\infty} C'_{1l} \sin 2l\tau \quad (37)$$

$$C'_{11} = \frac{1}{2} \varepsilon - \frac{9}{32} \varepsilon^3 + \frac{205}{1536} \varepsilon^5 + \dots \quad (38)$$

$$C'_{12} = \frac{5}{16} \varepsilon^2 - \frac{37}{96} \varepsilon^4 + \frac{1335}{4096} \varepsilon^6 + \dots \quad (39)$$

$$C'_{13} = \frac{29}{96} \varepsilon^3 - \frac{75}{128} \varepsilon^5 + \dots \quad (40)$$

$$C'_{14} = \frac{539}{1536} \varepsilon^4 - \frac{2391}{2590} \varepsilon^6 + \dots \quad (41)$$

$$C'_{15} = \frac{3467}{7680} \varepsilon^5 + \dots \quad (42)$$

$$C'_{15} = \frac{38081}{61440} \varepsilon^6 + \dots \quad (43)$$

$$\omega = ph(\cos \sigma + i \sin \alpha_0 \sin \sigma) \quad (44)$$

$$\beta = ph(|\cos \alpha_0 \cos \sigma + i \sin \alpha_0| + i \cos \alpha_0 \sin \sigma) \quad (45)$$

$$\alpha = ph(\cos \alpha_0 \cos \sigma + i \sin \alpha_0) \quad (46)$$

$$\varepsilon = \frac{\sqrt{1+k^2} - 1}{\sqrt{1+k^2} + 1} \quad (47)$$

$$A_1 = (1 - \varepsilon)^{-1} \left(1 + \frac{1}{4} \varepsilon^2 + \frac{1}{64} \varepsilon^4 + \frac{1}{256} \varepsilon^6 + \dots \right) \quad (48)$$

$$\tau_2 = s_2 / (bA_1) \quad (49)$$

Глава 3 Спецификация требований и проектирование

3.1 Требования

3.1.1 Общие требования

3.1.1.1 Система должна применять алгоритм увеличения контраста изображения перед его анализом.

3.1.1.2 Система должна применять алгоритм увеличения яркости изображения с учётом гистограммы яркости перед его анализом.

3.1.1.3 Система должна применять алгоритм изменения тонового распределения изображения перед его анализом.

3.1.1.4 Система должна использовать математическую модель камеры с коэффициентами прямого линейного преобразования.

3.1.1.5 Система должна использовать алгоритм 8-ми масок для утоньшения скелетов на изображении.

3.1.1.6 Система должна строить скелеты объектов на основе изображений скелетов, полученных после алгоритма наложения 8-ми масок.

3.1.1.7 Система должна сравнивать скелеты графов с помощью метода сравнения иерархий графов.

3.1.1.8 Система должна обрабатывать цифровые снимки в формате jpg.

3.1.1.9 Система должна обрабатывать цифровые снимки размером от 352 до 10320 пикселей по ширине и от 288 до 8956 пикселей по высоте.

3.1.2 Пользовательские требования

3.1.2.1 Интерфейс системы должен быть русскоязычным.

3.1.2.2 Система должна обеспечивать ввод группы цифровых снимков.

3.1.2.3 Система должна обеспечивать вывод количества найденных объектов на серии снимков.

3.1.2.4 Система должна обеспечивать вывод размеров, скорости, gps-координат, расстояния до камеры для каждого объекта каждого снимка.

3.1.2.5 Система должна обеспечивать вывод ортогональной проекции любого снимка.

3.1.2.6 Система должна обеспечивать возможность сохранения рассчитанных размеров, скорости, gps-координат, расстояния до камеры для любого объекта любого снимка в базу данных.

3.1.2.7 Система должна обеспечивать возможность сохранения в базу данных рассчитанной занимаемой льдом площади для любого изображения.

3.1.2.8 Система должна обеспечивать возможность сохранения полученной ортогональной проекции любого изображения.

3.1.2.9 Система должна обеспечивать ввод имени базы данных, в которую будет сохраняться информация из требований [3.1.2.6](#) и [3.1.2.7](#).

3.1.2.10 Система должна позволять удалять из базы данных информацию, которая была сохранена ею в соответствии с требованиями [3.1.2.6](#) и [3.1.2.7](#).

3.1.3 Функциональные требования

3.1.3.1 Система должна принимать цифровые снимки для проведения их анализа.

3.1.3.2 Система должна выделять движущиеся объекты на серии снимков (см. [3.1.3.1](#)).

3.1.3.3 Система должна определять расстояние от камеры до объекта (см. [3.1.3.2](#)) для каждого снимка (см. [3.1.3.1](#)).

3.1.3.4 Система должна определять скорость объекта (см. [3.1.3.2](#)) для каждого снимка из серии (см. [3.1.3.1](#)), кроме первого.

3.1.3.5 Система должна определять gps-координаты объекта (см. [3.1.3.2](#)) для каждого снимка (см. [3.1.3.1](#)).

3.1.3.6 Система должна определять размеры объекта (см. [3.1.3.2](#)) для каждого снимка (см. [3.1.3.1](#)).

3.1.3.7 Система должна строить ортогональную проекцию каждого снимка (см. [3.1.3.1](#)).

3.1.3.8 Система должна взаимодействовать с базой данных MySQL версии не ниже 5.

3.1.3.9 Система должна по требованию пользователя сохранять ортогональную проекцию (см. [3.1.3.7](#)) изображения в файл формата jpg.

3.1.3.10 Система должна по требованию пользователя сохранять информацию об объектах в базу данных, полученную в соответствии с требованиями [3.1.3.3](#) – [3.1.3.6](#).

3.1.3.11 Система должна по требованию пользователя удалять информацию об объектах из базы данных.

3.1.4 Требования к входным данным

3.1.4.1 Изображение – цифровое изображение, представленное в стандарте jpg.

3.1.4.2 Метаинформация в изображении (см. [3.1.4.1](#)) должно содержать информацию о gps-координатах камеры, горизонтальном и вертикальном углах обзора, азимуту камеры, высоте камеры над уровнем моря и время создания снимка.

3.1.5 Требования к выходным данным

3.1.5.1 Обнаруженные объекты на каждом снимке должны обводиться зелёным контуром.

3.1.5.2 Внутри каждого контура из требования [3.1.5.1](#) система должна рисовать уникальный номер объекта, который идентифицирует этот объект для всей серии снимков.

3.1.5.3 Система должна по требованию пользователя сохранять в базу данных (см. [3.1.3.8](#)) расстояние до объекта (см. [3.1.3.3](#)), скорость объекта (см. [3.1.3.4](#)), gps-координаты объекта (см. [3.1.3.5](#)), размеры объекта (см. [3.1.3.6](#)).

3.1.6 Требования к надёжности и мобильности системы

3.1.6.1 Система должна быть совместимой с операционными системами Windows XP, Windows 7 и Windows 8.

3.1.6.2 Система должна обеспечивать целостность базы данных при внесении изменений в неё.

3.1.7 Требования к серверу БД

3.1.7.1 СУБД должна являться MySQL версии не ниже 5.

3.1.7.2 БД не должна содержать базу данных, название которой совпадает с названием базы из требования [3.1.2.9](#).

3.1.7.3 СУБД должна быть установлена на IBM-совместимый компьютер с объёмом ОЗУ не менее 1024 Мб.

3.1.7.4 СУБД должна быть установлена на IBM-совместимый компьютер с тактовой частотой ЦП не менее 1,5 ГГц.

3.2 Проект

3.2.1 АКД

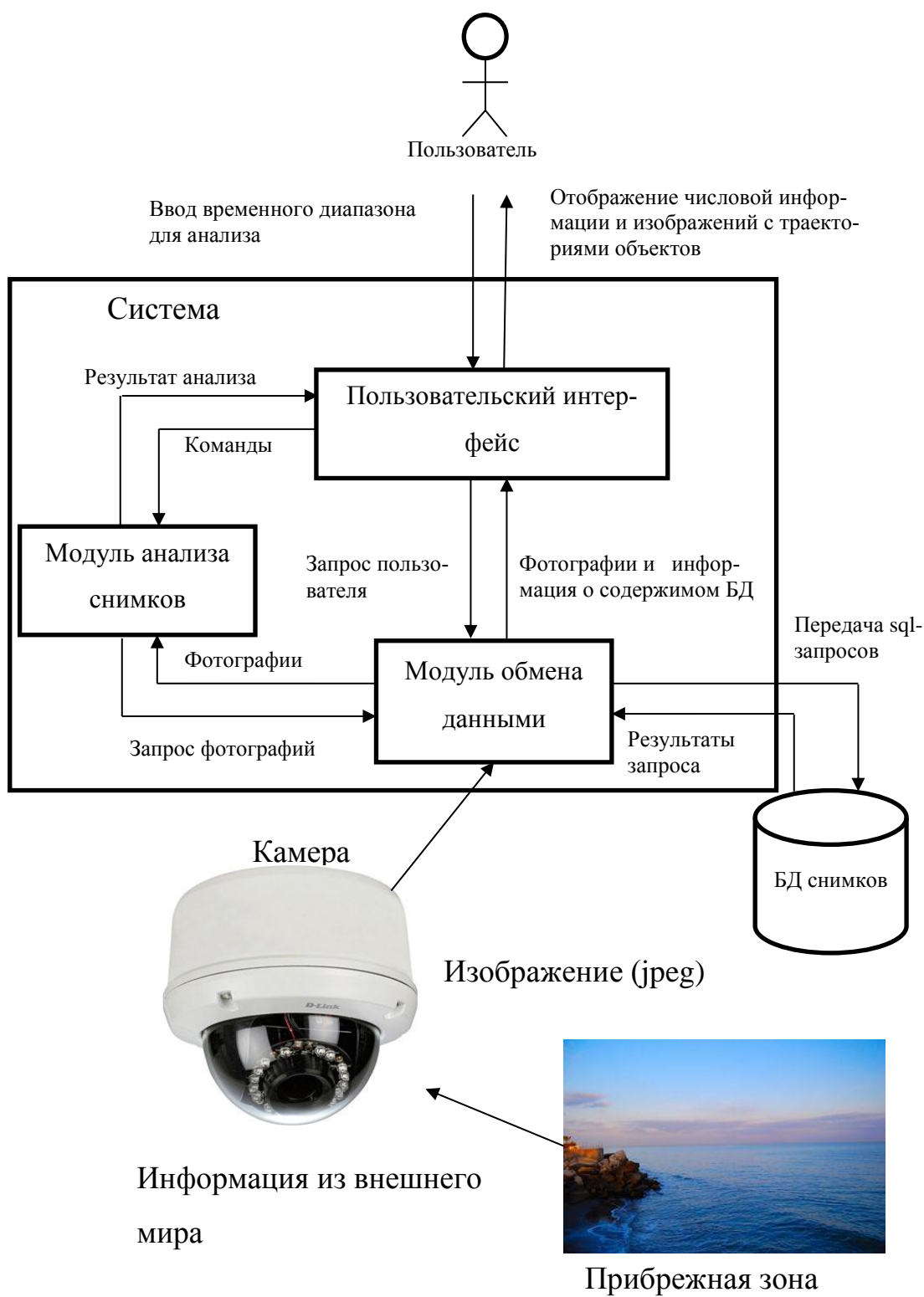


Рисунок 6 – Архитектурно-контекстная диаграмма

Архитектурно-контекстная диаграмма представляет схему взаимодействия компонентов системы между собой, а так же схему взаимодействия пользователя с системой.

Компоненты системы:

- a) Пользовательский интерфейс – компонент взаимодействия пользователя с системой
- b) Модуль анализа снимков – компонент, анализирующий снимки. Выделяет объекты на снимках и производит вычисление их характеристик.
- c) Модуль обмена данными – компонент связи других компонентов системы. Даёт возможность передавать данные между компонентами, а также просматривать и удалять содержимое БД снимков.
- d) БД снимков – система хранения результатов анализа снимков и объектов, выделенных на них.

Пользователь – человек, который через Пользовательский интерфейс указывает системе снимки, которые необходимо проанализировать с помощью Модуля анализа снимков. Также может просматривать и редактировать содержимое БД снимков.

Навыки:

- 1) Понимает русский язык.
- 2) Умеет работать с компьютерной мышью и клавиатурой.

Для начала работы программы пользователь вводит данные в систему с помощью пользовательского интерфейса. Пользовательский интерфейс передаёт введённые данные и команду начала работы Модулю анализа снимков. Тот посылает запрос на получение данных Модулю обмена данными. Модуль обмена данными посылает запрос БД снимков для их получения. В ответ БД снимков посылает данные Модулю обмена данными; он передаёт эти данные Модулю анализа снимков. Модуль анализа снимков производит вычисления и отправляет резуль-

таты Интерфейсу взаимодействия с системой; он отображает эти результаты пользователю.

Пользователь может просмотреть или удалить содержимое из БД снимков. Для этого пользователь вводит данные и обращается с соответствующей командой к Пользовательскому интерфейсу. Пользовательский интерфейс передаёт запрос Модулю обмена данными, который передаёт этот запрос БД снимков. Затем БД снимков обрабатывает полученный запрос и передаёт результат выполнения запроса модулю обмена данными. Тот в свою очередь передаёт эти данные Пользовательскому интерфейсу, который их отображает пользователю.

Камера служит для получения снимков местности и записи их в базу данных

3.2.2 Диаграмма вариантов использования

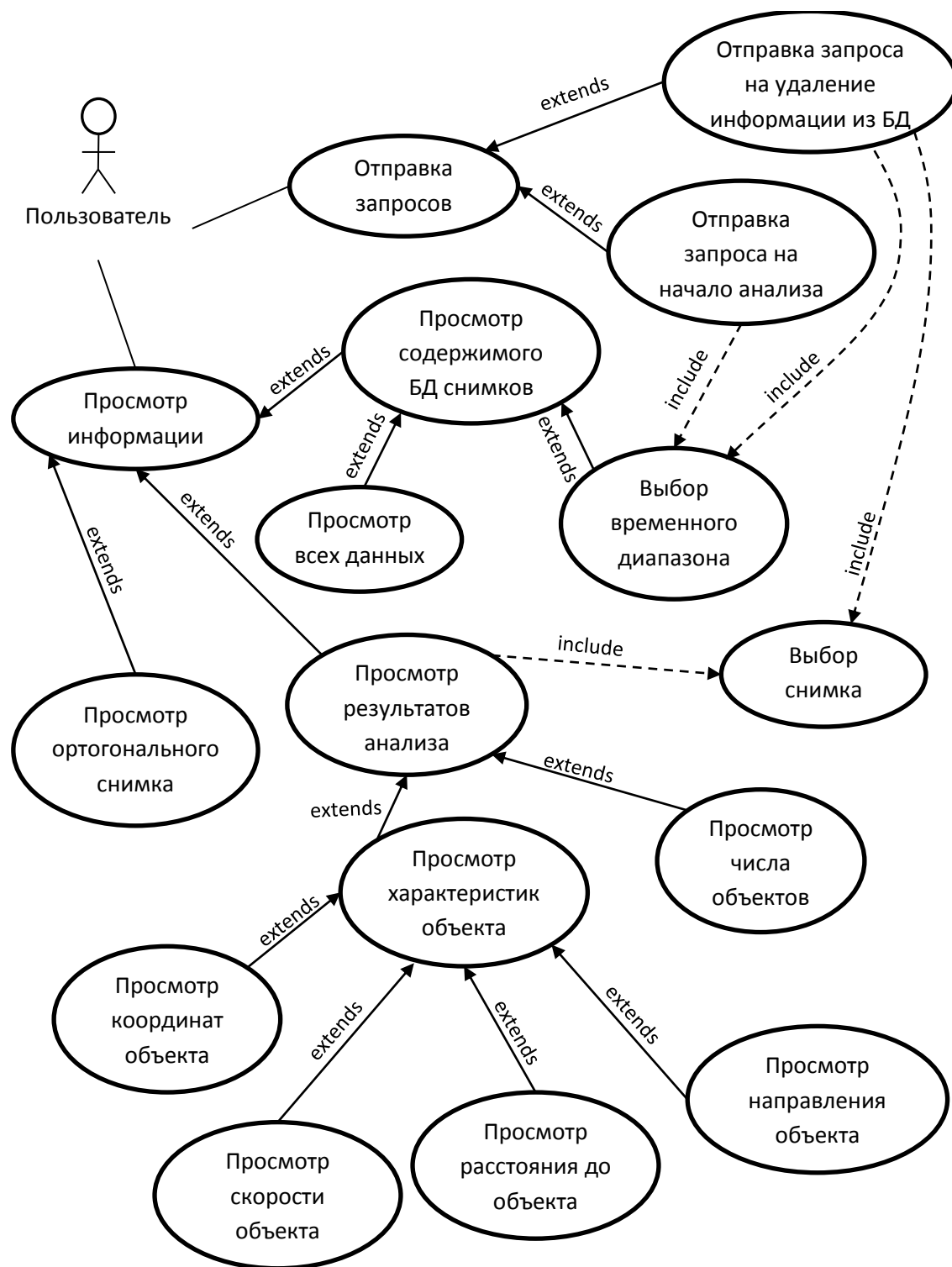


Рисунок 7 – Диаграмма вариантов использования

Пользователь:

– Запрашивает (инициирует):

- Начало анализа снимков
- Выбирает снимок
- Удаляет информацию из БД снимков
- Вводит начальную и конечную даты для запросов и операций
- Просматривает:
 - Результат запроса на удаление данных из БД снимков
 - Число объектов на снимке, выявленное в результате анализа
 - Просмотр географических координат конкретного объекта
 - Просмотр скорости конкретного объекта
 - Просмотр траектории конкретного объекта
 - Просмотр ортогональной проекции изображения

Пример

- Запрос анализа снимков
 - Пользователь переходит в меню анализа снимков
 - Выбирает временной диапазон для анализа
 - Выбирает снимок для просмотра результатов анализа
 - Просматривает число объектов на снимке
 - Просматривает координаты объектов
 - Просматривает расстояние до объектов
 - Просматривает скорость объектов
 - Просматривает направления объектов

3.2.3 Диаграмма взаимодействия

Диаграмма показывает последовательность взаимодействия компонентов системы между собой при участии пользователя.

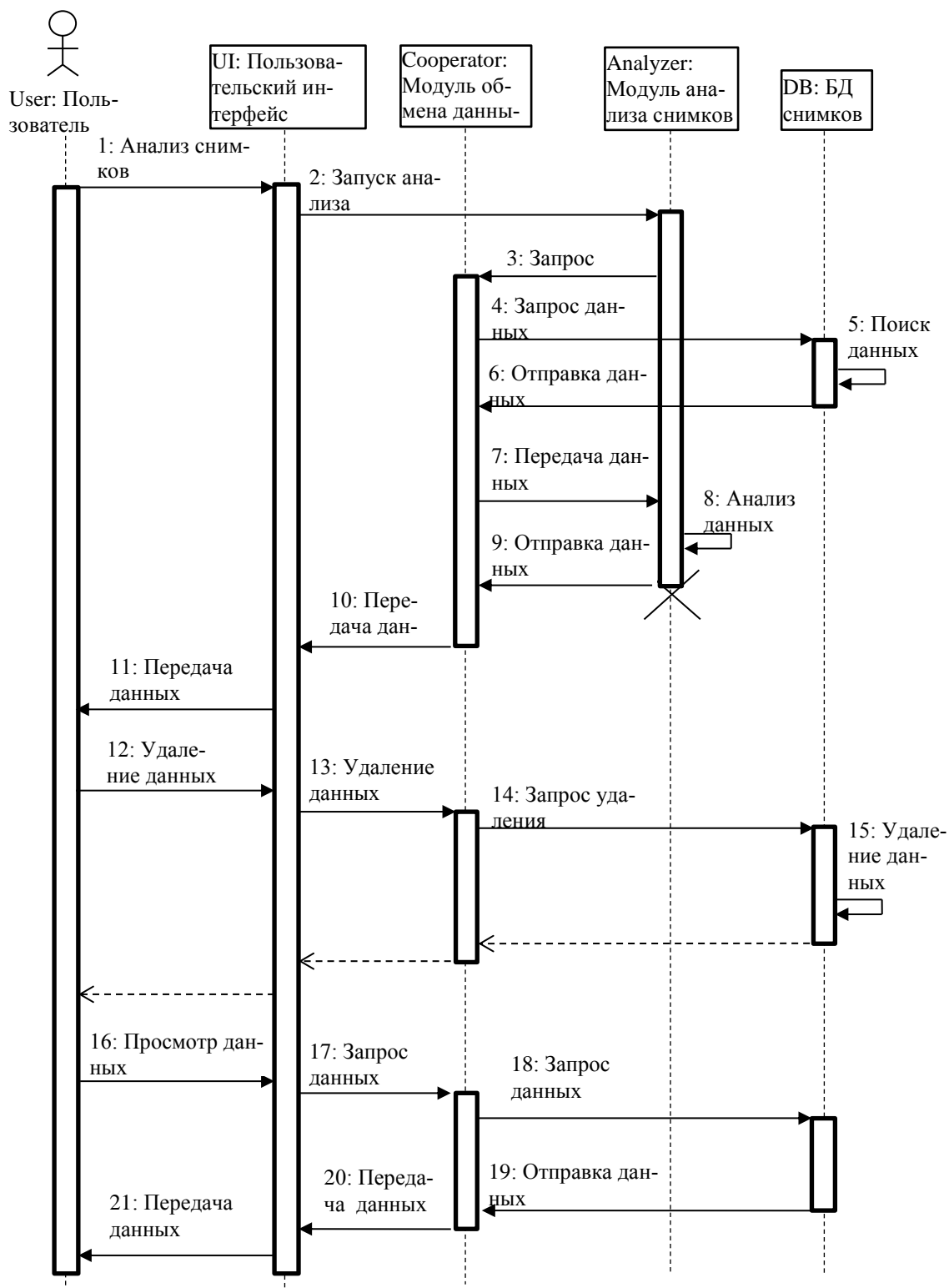


Рисунок 8 – Диаграмма взаимодействия

3.2.4 Диаграмма потоков данных

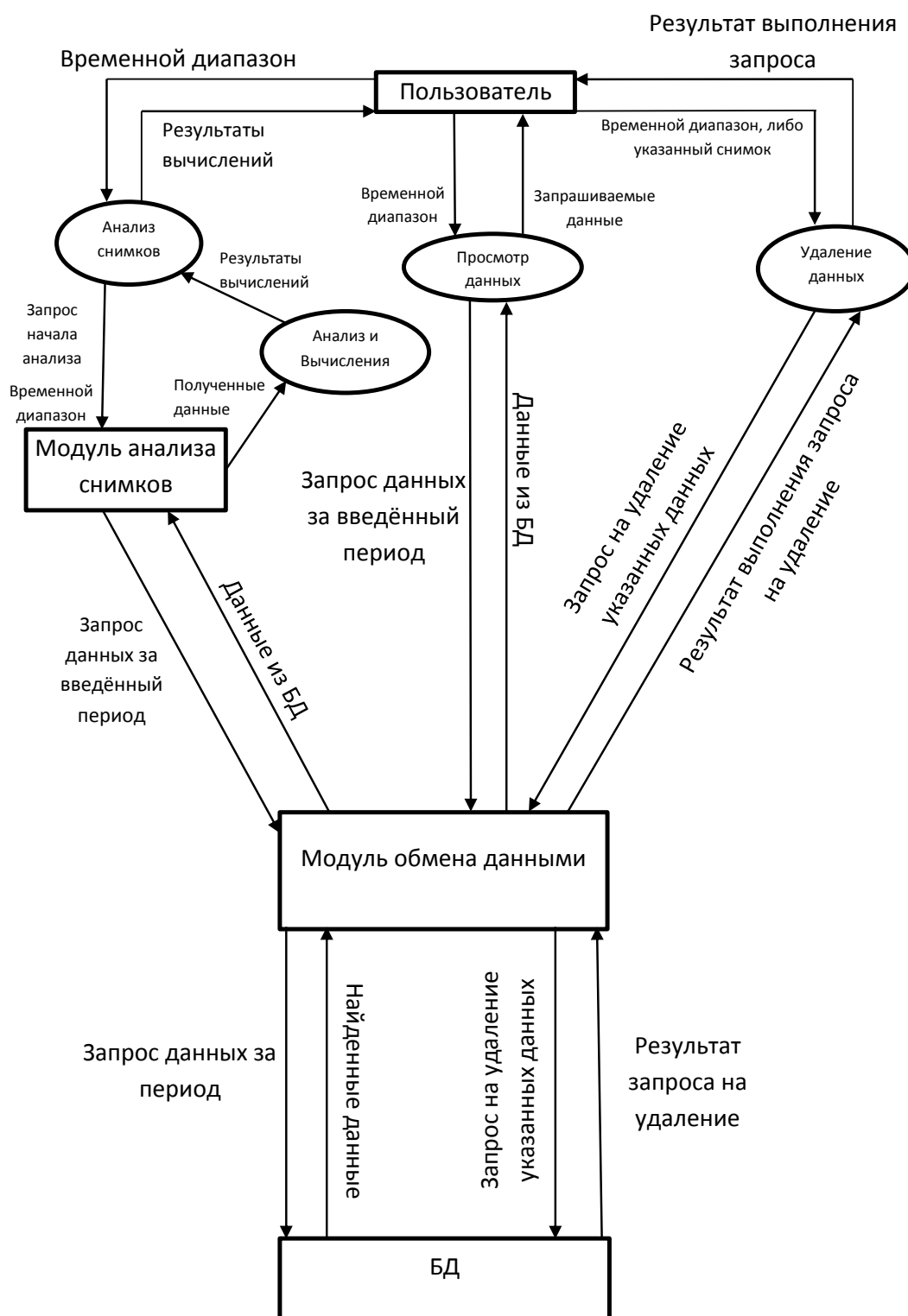


Рисунок 9 – Диаграмма потоков данных

3.2.5 Схема базы данных

Для работы с БД решено использовать MySQL-сервер.

Он будет использоваться для хранения фотографий и результатов их анализа.

База данных представляется следующей схемой:

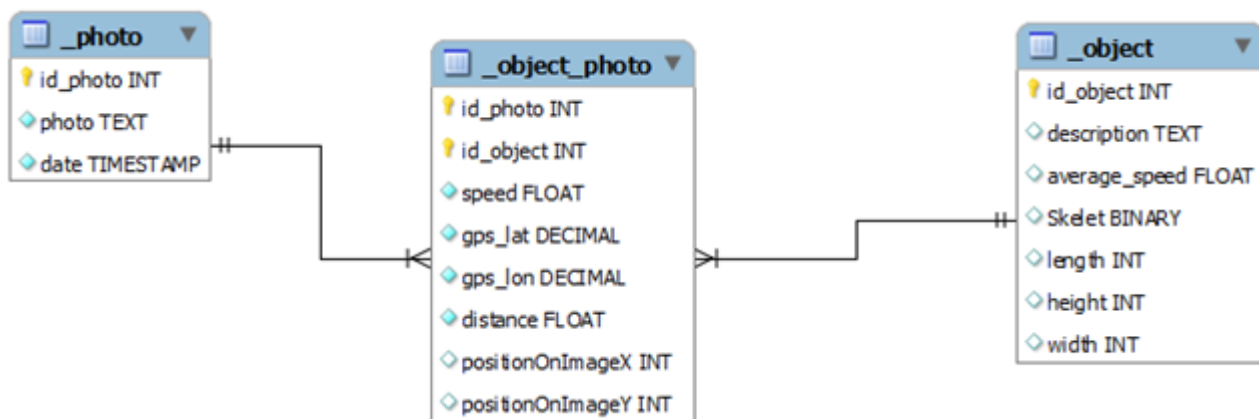


Рисунок 10 – База данных

Таблица _photo:

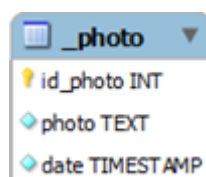
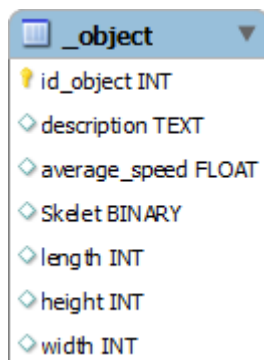


Рисунок 11 – Таблица фотографий

Представляет собой способ хранения фотографии:

- id_photo (INT) – первичный ключ
- photo(TEXT) – путь к фотографии в файловой системе
- date_time (TIMESTAMP) – время создания фотографии

Таблица _object



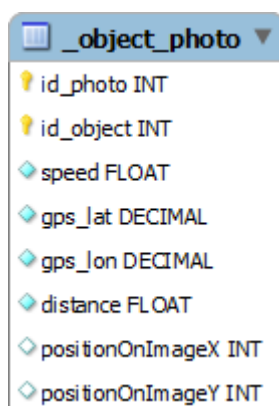
_object	
id_object	INT
description	TEXT
average_speed	FLOAT
Skelet	BINARY
length	INT
height	INT
width	INT

Рисунок 12 – Таблица объектов

Предназначена для хранения характеристик объектов.

- id_object (INT) – первичный ключ для объекта
- description (TEXT) – пользовательское описание объекта
- average_speed (FLOAT) – средняя скорость объекта для всей траектории
- Skelet (BINARY) – скелет объекта, полученный в программе
- length (INT) – длина объекта
- height (INT) – высота объекта
- width (INT) – ширина объекта

Таблица _object_photo



_object_photo	
id_photo	INT
id_object	INT
speed	FLOAT
gps_lat	DECIMAL
gps_lon	DECIMAL
distance	FLOAT
positionOnImageX	INT
positionOnImageY	INT

Рисунок 13 – Таблица связи фотографий с объектами

Связывает фотографию и найденные на ней объекты

- id_photo(INT) – первичный ключ из таблицы _photo

- `id_object` (INT) – первичный ключ из таблицы `_object`
- `speed` (FLOAT) – скорость объекта на текущем снимке
- `gps_lat` (DECIMAL) – широта положения объекта на текущем снимке
- `gps_lot` (DECIMAL) – долгота положения объекта на текущем снимке
- `distance` (FLOAT) – расстояние до объекта
- `positionOnImageX` (DECIMAL) – x-координата пикселя центра объекта

на текущем снимке

- `positionOnImageY` (DECIMAL) – y-координата пикселя центра объекта

на текущем снимке

3.2.6 Проект данных

3.2.6.1 Пользовательский интерфейс

`struct ObjectPhoto` – повторяет строку из базы данных в таблице `_object_photo`

```
{
    unsigned int id_photo, id_object;
    float speed, distance;
    double gps_lat, gps_lon, path_x, path_y;
};
```

3.2.6.2 Модуль анализа снимков

В OpenCV описан тип данных, предназначенный для представления изображения:

```
typedef struct _IplImage {
    int          nSize; // sizeof(IplImage)
    int          ID; // Версия (=0)
    int          nChannels; // Число каналов
    int          alphaChannel; // Альфа-канал
    int          depth; // Глубина в битах
```

```

char        colorModel[4]; // Не используется в OpenCV
char        channelSeq[4]; // Аналогично
int         dataOrder; // Расположение каналов
int         origin; // Начало координат
int         align; // Выравнивание строк изображения (OpenCV использует widthStep)

int         width; // Ширина
int         height; // Высота
struct _IplROI*   roi; // ROI
struct _IplImage* maskROI; // =0
void*       imageId; //
struct _IplTileInfo* tileInfo; //
int         imageSize; // Память выделенная под изображение
char*       imageData; // Данные изображения
int         widthStep; // Число байт в одной строке изображения
int         BorderMode[4]; // Не используется в OpenCV
int         BorderConst[4]; // Аналогично
char*       imageDataOrigin; // Используется для правильного освобождения памяти
} IplImage;

```

3.2.6.3 Построение скелета

```

struct CvPoint {
    int x, y;
}; - встроенная в OpenCV структура. Представляет точку на изображении.
struct coordinates {
    float latitude, longitude;
};
struct Arch {

```

```
vector <CvPoint> points;
```

}; - массив точек, из которых состоит дуга

```
struct Skelet {  
    vector <Arch> arch;
```

}; - массив дуг, составляющих скелет

```
struct TwoPoints {  
    CvPoint vertex; - координаты вершины  
    CvPoint neighbor; - координаты соседней точки для вершины vertex  
}; - структура представляет пару точек – вершина и её сосед. Используется  
при построении скелета объекта.
```

3.2.6.4 Нахождение объектов на снимке

Решено представлять связку объект-фотография с помощью класса:

```
class object_photo {  
private:  
    int photoId, objectId; - идентификаторы записей в базе данных  
    CvPoint positionOnImage; - координата центра пикселя объекта на  
изображении  
public:  
    float distance; - расстояние от камеры до объекта  
    coordinates gps; – координаты объекта для текущего снимка  
    Skelet skelet; - скелет объекта для текущего снимка  
};
```

3.2.6.5 Вычисление мировых координат

```
struct worldCoordinates {  
    float x,y,z; – мировые координаты  
}
```


Глава 4 Реализация и тестирование

При реализации были использованы спецификации, полученные на этапе проектирования.

4.1 Средства реализации

В качестве языка реализации выбран C++, так как C++ программы отличаются высокой производительностью.

Для разработки модуля анализа изображений используется библиотека компьютерного зрения OpenCV 4.4. Она содержит средства для доступа к данным изображений, преобразования их в различные форматы. Включая в себя большое множество различных операций преобразования цифровых снимков библиотека исключает необходимость реализации многих функций, ускоряя разработку. так же преимуществом OpenCV является использование оттестированных и оптимизированных функции обработки цифровых снимков.

В качестве СУБД выбран MySQL 5.6. Причиной выбора стала поддержка MySQL реляционной модели данных. Так как в базе данных не предполагается хранить больших объёмов информации, то MySQL полностью подходит для реализации данного программного средства.

4.2 Тестирование

Было проведено ручное тестирование модуля анализа снимков. Проведены тесты алгоритмов улучшения качества изображений, выделения и построения скелетов объектов.

Для тестирования улучшения качества снимков были выбраны настоящие фотографии. Улучшение качества изображения заметно визуально. Тесты успешно пройдены.

Тестирование выделения скелетов объектов проведено также на реальных снимках акваторий. В зависимости от погодных условий, запечатлённых на фотографиях, достоверность выделения скелетов может уменьшаться до 70%.

Функция построения скелетов объектов была протестирована как на искусственно созданных снимках, так и на настоящих фотографиях, полученных в результате выделения скелетов. Тесты успешно пройдены.

Таким образом проведено тестирование реализованных функций системы. Успешность прохождения тестов 90%.

4.3 Выводы

По причине того, что для реализации использовались OpenCV и MySQL, исходный код может быть перенесён на Unix платформу без больших изменений.

Результатом выполненной работы является программное средство «Анализ панорамных снимков».

Глава 5 Испытания системы «Анализ панорамных снимков»

5.1 Описание предметной области испытаний

Рассмотрим предметную область «Анализ снимков пролива, полученных с помощью стационарных поворотных камер».

В качестве исходных данных профессионал предметной области располагает группами снимков, полученных со стационарной поворотной камеры. Для камеры известны её пространственные и оптические характеристики.

Результаты вычислений заносятся в таблицы на бумажных или электронных бланках. План местности рисуется на бумажных носителях или в электронных документах.

5.2 Автоматизация предметной области

Для проверки работоспособности программного средства было проведено его испытание в реальных условиях.

5.2.1.1 Попытка вычисления параметров объектов на не последовательных снимках

Описание: снимок DC_00005 имеет дату создания на 10 минут позже самой последней даты создания остальных снимков.

Результат: система вывела сообщение о том, что снимок DC_00005 не подходит для последовательности из-за слишком большой даты создания.

Итог: испытание пройдено.

5.2.1.2 Получение характеристик объектов

Описание: введённые снимки являются последовательными. Пользователь запустил процесс получения характеристик объектов.

Результат: система проанализировала снимки, выделила на них объекты и составила список найденных объектов и их характеристик.

Итог: испытание пройдено.

5.2.1.3 Сохранение полученных результатов в базу данных

5.2.1.4 Описание: пользователь ввёл снимки и запустил процесс получения характеристик объектов. Система проанализировала снимки и вывела вычисленные характеристики объектов для этих снимков. Пользователь выбрал сохранение результата.

Результат: для выбранного снимка система создала запись в таблице фотографий базы данных снимков и записала в неё полученный результат.

Итог: испытание пройдено.

5.2.1.5 Построение ортогональной проекции

Описание: пользователь ввёл снимки и запустил процесс получения ортогональной проекции одного из снимков.

Результат: система построила и вывела ортогональную проекцию выбранного снимка.

Итог: испытание пройдено.

5.3 Преимущества автоматизации

С помощью системы «Анализ панорамных снимков» становится возможным автоматизировать выполнение задач анализа снимков. Система упрощает хранение и просмотр полученных результатов анализа, полностью перенимает на себя вычислительные задачи, тем самым значительно сокращая время достижения цели – получения данных о панорамных снимках и их объектах.

Заключение

В результате работы над дипломной работой был произведён обзор литературы на тему «Существующие системы распознавания объектов на панорамных электронных снимках» на основе информации из 34 источников, включая статьи о распараллеливании компьютерных процессов, повышении качества изображений и способах поиска похожих объектов на цифровых снимках. Обзор показал, что существует ряд алгоритмов, подходящих для реализации программной системы, решающей поставленные задачи, но аналогов разработанной системы нет.

Был произведён анализ предметных областей «Определение параметров объектов на снимках» и «построение ортогональной проекции изображений» и их моделирование с помощью языка онтологий.

Реализация программной системы была осуществлена с помощью системы компьютерного зрения OpenCV 2.4.4. Взаимодействие с базой данных было установлено с помощью MySQL Connector/C++. Способы работы со средствами OpenCV и MySQL Connector/C++ были изучены по русско- и англоязычным статьям, а так же с помощью их документации.

База данных была реализована в СУБД MySQL 5. В процессе проектирования БД использовалась программная система MySQL Workbench 5.2.

В ходе проектирования был изучен способ обработки изображений, представленных в стандартном формате библиотеки OpenCV, улучшено владение языком SQL, изучен способ взаимодействия с базой данных СУБД MySQL с помощью кода C++.

Перечень результатов дипломной работы:

- составлен обзор литературы на тему «Определение параметров движения объектов на панорамных снимках» и обоснована необходимость разработки системы «Анализ панорамных снимков»;

- осуществлён анализ предметных областей «Определение параметров объектов на снимках» и «построение ортогональной проекции изображений» и разработаны их формальные модели;
- спроектирована и реализована система «Анализ панорамных снимков»;
- осуществлено тестирование системы «Анализ панорамных снимков»;
- описан эксперимент и указаны преимущества от автоматизации описанной предметной области с помощью системы «Анализ панорамных снимков»;

Перечисленные результаты полностью покрывают поставленные цели дипломной работы. Все задачи, необходимые для достижения поставленных целей, были решены.

Литература

1. Holland K.T., Holman R.A., Lippmann T.C., Stanley J. «Practical use of video imagery in nearshore oceanographic field studies» IEEE Journal Of Oceanic Engineering. 1997. VOL. 22, № 1, P.81–82.
2. J. R. Apel, “Ocean science from space,” EOS Trans. Amer. Geophys. Union, vol. 57, pp. 612–624, 1976.
3. P. G. McCurdy, Manual of Photogrammetry. New York: Pitman, 1944, pp. 1–841.
4. R. Y. Tsai, “A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses,” IEEE J. Robot. Automat., vol. RA-3, pp. 323–344, 1987.
5. Y. D. Park and B. H. Roh “Directshow-based multi-channel surveillance system architecture for simultaneous real-time remote monitoring on the internet,” Collection of Learned Papers of Fall Conferences of the Korea information Science Society, Vol. 30, 2003, pp. 424-426.
6. Remote Detection and Monitoring of a Water Level Using Narrow Band Channel JAEHYOUNG YU AND HERNSOO HAHN //JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 26, 71-82 (2010)
7. Arguello R. J., Sellner H. R., Stuller J. A., The Effect of Channel Errors in the Differential Pulse-Code Modulation Transmission of Sampled Imagery, IEEE Trans. Commun. Tech., COM-19, 6, 926—933 (December 1971).
8. Лурье А. И. Аналитическая механика — М.:Физматлит — 1961 г. — 824 с.
9. С.С. Ефимов, ОБЗОР МЕТОДОВ РАСПАРАЛЛЕЛИВАНИЯ АЛГОРИТМОВ РЕШЕНИЯ НЕКОТОРЫХ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ – Математические структуры и моделирование 2007, вып. 17, с. 72–93

10. Л. Шапиро, Дж. Стокман Компьютерное зрение. Перевод с английского А. А. Богуславского под редакцией С. М. Соколова – М.: БИНОМ. Лаборатория знаний, 2006. - 752 с
11. Huang T. S , Chikhaoui M. T., The Effect of BSC on PCM Picture Quality, IEEE Trans. Inf. Theory, IT-13, 2, 270—273 (April 1967).
12. Young I. T. Mott-Smith J. C, On Weighted PCM, IEEE Trans. Inf. Theory (Correspondence), IT-11, 4, 596—597 (October 1965).
13. Huang T. S., Tretiak O. J., Prasada B., Yaraaguchi Y., Design Considerations in PCM Transmission of Low-Resolution Monochrome Still Pictures, Proc. IEEE, 55, 3, 331—335 (March 1967). [Имеется перевод: Хуан, Третьяк, Прасада, Ямагуши. Соображения по конструированию КИМ-систем для передачи неподвижных одноцветных изображений низкой четкости. — 1ИПЭР тематический вып. «Сокращение избыточности», 1967, т. 55, № 3, с. 97—102.]
14. Pratt W. K., Digital Color Image Coding and Transmission, University of Southern California, Electronic Sciences Laboratory, USCEE Report 403, June 1971.
15. Knight J. M., Maximum Acceptable Bit Error Rates for PCM Analog and Digital TV Systems, Proceedings National Telemetry Conference, 1962, pp. 1-9.
16. Papoulis A., Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1965.
17. Выделение и анализ скелетов объектов на цветных снимках [Электронный ресурс]. – Режим доступа: <http://swsys.ru/index.php?page=article&id=2198> Дата обращения 18.06.2013
18. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: Изд-во Ин-та матем., 1999. 270 с.
19. 33. OpenCV шаг за шагом. Сравнение контуров через суммарные характеристики - моменты / Компьютерное зрение / RoboCraft [Электронный ресурс]. – Режим доступа: <http://robocraft.ru/blog/computervision/867.html> Дата обращения 18.06.2013

20. T. Vincenty, "Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations", Survey Review XXIII.1975. Vol. 93, P. 88-89.
21. Rainsford, I. F. Long geodesies on the ellipsoid. Bull. Geocl., No. 37, 1955.
22. Rapp, R. H. Geometric geodesy notes. The Ohio State Univ., 1969.
23. Charles F. F. Karney Algorithms for geodesics SRI International, 201 Washington Rd, Princeton, NJ 08543-5300, USA
24. R. A. Holman, A. H. Sallenger, Jr., T. C. Lippmann, and J. W. Haines, "The application of video image processing to the study of nearshore processes", Oceanography. 1993. Vol. 6, P. 85, P. 78–85.
25. Назаров А.С. Фотограмметрия: учебное пособие для студентов вузов. Мн.: ТетраСистемс, 2006. 368с.
26. Использование GPU для решения задач компьютерного зрения в библиотеке OpenCV [Электронный ресурс]. – Режим доступа:
https://docs.google.com/viewer?a=v&q=cache:T09dp1hWRy8J:agora.guru.ru/hpc=h/files/Using_GPU_for_computer_vision_in_OpenCV_library.pdf+&hl=ru&gl=ru&pid=bl&srcid=ADGEEShIycclmOxBOLkpgmLz_6WJEquDMzD9XoTZ7Ft8auLGTltb1gvGvCFizlqTNuxJJGE8S28R4n3fV1Hnw021ffaLpw6D0nygpreKtStJFbp3_zW6iVqUGs_GXC3unMzPGP7VKyXO&sig=AHIEtbRBmnd7FXJgBIHlYHjgN6bwoy-3Ww Дата обращения 18.06.2013
27. Развитие вычислений на GPU: преимущества архитектуры Fermi — МИР NVIDIA [Электронный ресурс]. – Режим доступа:
<http://nvworld.ru/articles/calculations-on-gpu-advantages-fermi/> Дата обращения 18.06.2013
28. Гергель В.П., Стронгин Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. Учебное пособие: Изд. 2-е, доп-е. Н.Новгород: изд-во ННГУ, 2003.
29. Миллер Р., Боксер Л. Последовательные и параллельные алгоритмы: Общий подход. М: Бином. Лаборатория знаний, 2006. 406 с.

30. Букатов А. А., Дацюк В. Н., Жегуло А. И. Программирование многопроцессорных вычислительных систем. Ростов-на-Дону: Изд-во ООО «ЦВВР», 2003. 208 с.

31. Постановление Правительства РФ от 28.07.2000 N 568 "Об установлении единых государственных систем координат" - Нормативные правовые акты - Бланки и документы - Управление Росреестра по Владимирской области – Росреестр [Электронный ресурс]. – Режим доступа: http://www.to33.rosreestr.ru/blanks/blanks_norm/1027239/ Дата обращения 18.06.2013

32. Клещев А.С., Артемьева И.Л. Необогатенные системы логических соотношений. Часть 1. // Научно-техническая информация. Сер. 2. 2000. №7. С. 18-28.

33. Клещев А.С., Артемьева И.Л. Необогатенные системы логических соотношений. Часть 2. // Научно-техническая информация. Сер. 2. 2000. №8. С. 8-18.