

ТИТУЛЬНИК

T3

T3

календарный план

## Реферат

РПЗ 111 страниц, рисунков, 12 таблиц, 10 источников.

Ключевые слова: отслеживание объекта, GPS-данные, коррекция погрешностей GPS, объединение данных от разных источников.

Цель квалификационной работы бакалавра – разработка метода для построения и уточнения маршрута объекта по набору GPS-координат.

В квалификационной работе бакалавра проведён обзор средств решения сходных задач с обоснованием необходимости разработки метода, проанализирована предметная область. Предложен метод объединения нескольких последовательностей координат от разных источников и уточнения полученного маршрута. Создан программный продукт, реализующий предложенный метод, и проведен эксперимент, подтверждающий применимость разработанного метода.

## Оглавление

Введение . . . . .	10
1 Аналитический раздел . . . . .	11
1.1 Цель и задачи работы . . . . .	11
1.2 Причины погрешности . . . . .	12
1.3 Существующие решения . . . . .	13
1.3.1 Объединение данных от разных источников .	13
1.3.2 Устранение пиков . . . . .	14
1.3.3 Определение мест остановок . . . . .	15
1.3.4 Сглаживание погрешностей . . . . .	16
1.3.5 Выводы . . . . .	16
1.4 Возможные сферы применения . . . . .	17
1.5 Постановка задачи . . . . .	17
1.5.1 Входные параметры . . . . .	17
1.5.2 Результат . . . . .	18
1.5.3 Функциональная модель задачи . . . . .	18
1.6 Возможные варианты решения . . . . .	19
1.7 Обзор алгоритмов сглаживания . . . . .	20
1.7.1 Медианный фильтр . . . . .	20
1.7.2 Экспоненциальное сглаживание . . . . .	21
1.7.3 Ядерное сглаживание . . . . .	21
1.7.4 Фильтр Калмана . . . . .	22
1.7.5 Выводы . . . . .	23
2 Конструкторский раздел . . . . .	24
2.1 Объединение последовательностей координат . . . .	24
2.2 Устранение пиков . . . . .	25
2.3 Определение остановок . . . . .	27
2.4 Сглаживание погрешностей маршрута . . . . .	29
2.4.1 Построение математической модели с исполь-	
зованием фильтра Калмана . . . . .	29
2.4.2 Применение фильтра Калмана к построенной	
модели . . . . .	32
2.4.3 Согласование единиц измерения . . . . .	33

2.5	Метод уточнения и построения маршрута . . . . .	36
2.6	Структура разрабатываемого ПО . . . . .	38
2.7	Диаграмма классов метода . . . . .	39
3	Технологический раздел . . . . .	40
3.1	Выбор средств разработки . . . . .	40
3.1.1	Выбор целевой платформы . . . . .	40
3.1.2	Выбор языка программирования . . . . .	40
3.1.3	Выбор среды программирования и отладки . . . . .	40
3.1.4	Выбор средства создания интерфейса . . . . .	41
3.1.5	Средство для работы с картой. . . . .	41
3.2	Система контроля версий . . . . .	41
3.3	Физическая структура ПО . . . . .	42
3.4	Требования к вычислительной системе . . . . .	42
3.5	Форматы данных . . . . .	43
3.6	Тестирование программного продукта . . . . .	45
3.7	Установка программного обеспечения . . . . .	45
3.8	Руководство пользователя . . . . .	45
3.8.1	Управление картой . . . . .	45
3.8.2	Выбор маршрута . . . . .	45
3.8.3	Информация о маршруте . . . . .	46
3.8.4	Экспорт результата . . . . .	46
3.8.5	Настройки . . . . .	46
3.9	Пример работы программы . . . . .	47
4	Экспериментальный раздел . . . . .	49
4.1	Постановка эксперимента . . . . .	49
4.1.1	Описание эксперимента . . . . .	49
4.1.2	Размер выборки . . . . .	49
4.1.3	Данные для эксперимента . . . . .	50
4.2	Результат эксперимента . . . . .	51
4.3	Выводы . . . . .	52
5	Организационно-экономический раздел . . . . .	53
5.1	Определение этапов и содержания работ. Расчёт трудоёмкости. . . . .	53

5.2	Расчет трудоемкости . . . . .	55
5.3	Определение численности исполнителей . . . . .	57
5.4	Сетевая модель проекта . . . . .	58
5.5	Календарный график выполнения проекта . . . . .	59
5.6	Анализ структуры затрат проекта . . . . .	62
5.6.1	Расчёт расходов . . . . .	62
5.6.2	Расчет расходов, связанных с обеспечением работ оборудованием . . . . .	64
5.6.3	Расчет расходов, связанных с организацией рабочих мест . . . . .	64
5.6.4	Расчёт накладных расходов . . . . .	65
5.7	Исследование рынка для разрабатываемого ПО . . . . .	68
5.8	Планирование цены и прогнозирование прибыли . . . . .	69
5.9	Выводы . . . . .	70
6	Промышленная экология и безопасность . . . . .	71
6.1	Введение . . . . .	71
6.2	Требования к рабочему месту . . . . .	71
6.2.1	Требования к ПЭВМ . . . . .	71
6.2.2	Требования к помещению . . . . .	75
6.2.3	Требования к микроклимату . . . . .	75
6.2.4	Требования к содержанию аэроионов и вредных химических веществ в воздухе . . . . .	76
6.2.5	Требования к уровням шума . . . . .	77
6.2.6	Требования к освещению . . . . .	77
6.2.7	Требования к визуальным параметрам ВДТ . . . . .	79
6.2.8	Требования к организации и оборудованию рабочих мест . . . . .	79
6.3	Расчёты . . . . .	80
6.3.1	Схема помещения . . . . .	80
6.3.2	Расчёт освещённости помещения . . . . .	81
6.3.3	Типы ламп и светильников. . . . .	82
6.3.4	Выбор системы освещения . . . . .	82
6.3.5	Расположение светильников и их количество . . . . .	82



6.3.6	Расчёт уровней звукового давления . . . . .	86
	Заключение . . . . .	89
	Литература . . . . .	91

## Введение

В современном мире важное значение имеют средства навигации, такие как GPS и ГЛОНАСС. Эти средства активно используются как организациями, так и индивидуальными пользователями для решения широкого спектра задач. Существуют информационные системы, позволяющие сохранять маршрут объекта для последующего просмотра и анализа.

Однако с данной задачей связан ряд проблем. В первую очередь, датчики местоположения имеют погрешность, которая варьируется в зависимости от качества приема и других факторов. Такая погрешность может достигать значений в десятки километров. В результате получаются недостоверные маршруты, что затрудняет их анализ. Кроме того, с отслеживаемым объектом могут быть связаны несколько источников данных. Например, в случае индивидуального пользователя это могут быть смартфон со встроенными средствами навигации и какое-либо специализированное устройство. Данные от нескольких источников различаются, поэтому возникает задача их объединения.

В данной работе рассматривается влияние погрешностей на воспроизводимый маршрут объекта и предлагается метод, позволяющий объединять данные с нескольких устройств и повышать точность маршрута.

# 1. Аналитический раздел

## 1.1. Цель и задачи работы

Целью данной работы является разработка метода, позволяющего воспроизводить маршрут наблюдаемого подвижного объекта по последовательности координат, полученных от одного или нескольких связанных с объектом источников, а также устранять погрешности и повышать точность маршрута.

Предполагается использование метода для обработки полного маршрута объекта, а не в режиме слежения.

На рис. 1.1 приведен пример двух маршрутов объекта, полученных от разных датчиков.

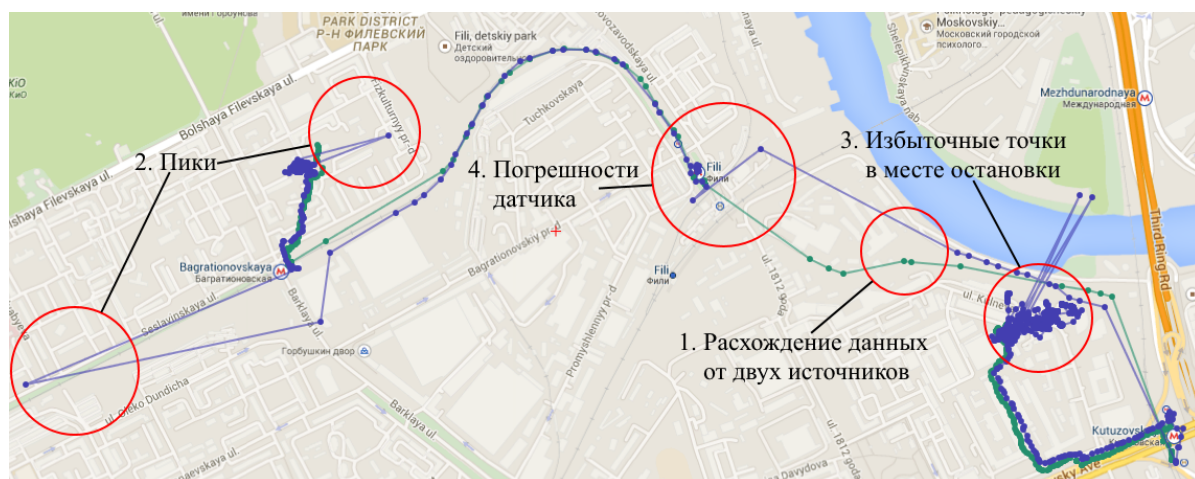


Рисунок 1.1 — Пример маршрутов от двух датчиков, содержащих погрешности

На приведенном рисунке указаны наиболее распространенные проблемы, заметные при анализе маршрута объекта:

— Расхождение данных от разных датчиков. В этом случае невозможно установить, какой из маршрутов является более достоверным. Также возможен случай, при котором на разных участках маршрута более точными являются данные от разных датчиков.

— Присутствуют одиночные точки с высокой степенью погрешности (более 100 м). Для обозначения таких точек в данной работе вводится понятие «пик». Наличие таких точек в маршруте мо-

жет значительно исказить результаты анализа маршрута. Например, недостоверной становится оценка пройденного объектом расстояния.

— Во время длительной стоянки объекта накапливается большое количество точек. Такие точки являются избыточными, так как относятся к одному местоположению. Кроме того, из-за погрешностей датчика точки не совпадают.

— Присутствуют точки с небольшой погрешностью. Данные точки также вносят искажения и снижают достоверность маршрута.

Таким образом, возникают следующие задачи:

- а) Объединение маршрутов, полученных от разных источников
- б) Устранение пиков
- в) Устранение избыточных координат, накопленных во время стоянки объекта
- г) Сглаживание погрешностей датчика

## **1.2. Причины погрешности**

Источником координат может быть широкий спектр устройств:

— Датчики, установленные в мобильные телефоны и планшетные компьютеры. Такие датчики используют несколько источников для определения местоположения: станции мобильной связи, спутники GPS/ГЛОНАСС, данные, полученные из сетей Wi-Fi. Зачастую нельзя получить информацию о источнике местоположения, а в некоторых датчиках результат получается методом комбинирования данных с нескольких источников.

— Специализированные устройства навигации. Источником данных служит GPS/ГЛОНАСС.

— Внешние GPS-датчики, подключаемые к различным устройствам с помощью USB и COM.

Основные факторы, вызывающие погрешность [1]:

— Временная потеря сигнала спутника и использование только базовых станций.

- Количество и взаимное расположение спутников.
- Наличие периодов коррекции спутников.
- Особенности местности, в которой используется датчик.
- Явление отражения сигналов спутника и преграды на их пути.
- Временная погрешность датчика.
- Солнечная интерференция.
- Ошибки округления и вычисления в датчиках.

Несмотря на множество факторов, в данной работе рассматривается только результирующая погрешность как некоторая случайная величина.

### **1.3. Существующие решения**

#### **1.3.1. Объединение данных от разных источников**

Данная задача является разновидностью задачи слияния данных датчиков (sensor fusion) [2]. Данная задача подразумевает использование данных от нескольких независимых источников измерения для получения точности, превышающей точность датчиков при изолированном использовании. Однако методы решения таких задач требуют построение сложной модели наблюдаемого процесса и исследование параметров датчиков, влияющих на их погрешность.

Частное решение данной задачи в рассматриваемой предметной области присутствует в современных смартфонах. Смартфоны используют несколько источников информации о местоположении:

- а) Глобальные системы спутниковой навигации (GPS/ГЛОНАСС).
- б) Данные сетей сотовой связи.
- в) Данные беспроводных сетей Wi-Fi.

Средствами системного ПО происходит выбор наиболее точных доступных данных, которые затем передаются в программы, запросившие определение местоположения.

Данное решение не подходит к рассматриваемой задаче по следующим причинам:

- Объединение информации происходит на уровне системного ПО устройства. Пользователь не имеет возможности повлиять на процесс или предоставить дополнительные данные, которые необходимо учесть.

- Объединение происходит в режиме реального времени, что противоречит условию задачи.

- Известна априорная точность источников данных о местоположении. Таким образом, итоговый результат может являться данными от датчика с максимальной точностью, доступного в текущий момент времени.

Не удалось обнаружить общедоступные программные продукты, выполняющие объединение маршрутов, полученных от разных физических устройств.

### **1.3.2. Устранение пиков**

Данная задача решается во многих программах-навигаторах. Существуют два основных подхода:

- а) Использование данных о предполагаемом маршруте объекта и графа дорог. В данном решении принимается допущение, что объект, вероятнее всего, движется по рекомендуемому программой маршруту. Кроме того, используется информация о дорогах и отбрасываются координаты, где объект не может находиться физически. Недостатки такого решения:

- Требуется дополнительная информация о дорогах. Это требование ограничивает применение данного подхода в случаях неполной информации о дорогах. Кроме того, данные занимают большие объёмы памяти, что затрудняет их автономное использование. Обычно такие данные предоставляются онлайн-сервисами (например, Ян-

декс.Навигатор), либо хранятся на специальном устройстве, используемом только для навигации.

— Объект может находиться не на дороге. Такое решение в указанном случае дает заведомо недостоверный результат. Перечисленные недостатки ограничивают область применения данного решения.

б) Использование информации о точности, получаемой от датчика. Датчики предоставляют данные о предполагаемой погрешности полученной координаты. Удаление координат с высокой погрешностью уменьшает количество пиков. Однако такое решение не способно устранить все пики, так как нередко датчики предоставляют неверные сведения о точности. Таким образом, данный подход является ненадежным.

### **1.3.3. Определение мест остановок**

В смартфонах решается сходная задача уменьшения количества избыточных координат, однако целью решения данной задачи является не повышение точности данных о маршруте, а сокращение потребления энергии. Устройство может не оповещать приложения о поступлении новых данных о местоположении, если разрешен режим энергосбережения, при выполнении следующих условий:

- Новые данные отличаются от ранее полученных на незначительную величину
- Скорость движения объекта, согласно датчику, равна нулю.

Данное решение может снизить количество накапливаемых за время стоянки точек, однако многие системы, собирающие информацию о передвижении наблюдаемых объектов, требуют отчета о местоположении через фиксированные интервалы времени. Таким образом, приложения вынуждены принудительно опрашивать датчик с целью получения новых данных о местоположении и сохранять их. В таких случаях описанное решение не работает.

#### **1.3.4. Сглаживание погрешностей**

Предварительная коррекция данных происходит в процессе работы датчика. Пользователю предоставляются уже скорректированные данные. Однако степень коррекции недостаточна, так как на итоговом маршруте присутствуют погрешности. Дополнительное исправление ошибки происходит в прикладных программах. В программах навигации погрешности корректируются с помощью информации о дороге и предполагаемом маршруте. В данных программах нет принципиального отличия пиков от незначительных погрешностей. Помимо перечисленных ранее недостатков такого решения, присутствует следующий: для параллельных дорог, по которым объект может передвигаться с равной вероятностью (например, проспект и дублёр), программа навигации не может определить местоположение. В результате объект может отображаться не на той дороге, по которой он передвигается, либо может наблюдаться поочередная смена дороги с некоторой периодичностью. В результате при воспроизведении маршрута могут наблюдаться участки, на которых траектория объекта заметно отличается от реальной.

Альтернативным вариантом является обработка всего маршрута при его загрузке в соответствующий сервис. Например, сглаживаются погрешности при сохранении истории о перемещениях в сервисе RunKeeper [3]. Возможно, помимо известных алгоритмов сглаживания сигналов, могут использоваться дополнительные данные, например, от акселерометра смартфона, однако точные алгоритмы неизвестны.

#### **1.3.5. Выводы**

Анализ существующих решений показал, что предложенные к рассмотрению задачи актуальны и решаются. Однако у приведенных решений есть недостатки, ограничивающие их применимость. Необходимо отметить, что точные способы решения не известны. Предположения об используемых данных и методах сделаны исходя из на-



блюдаемых результатов и ограничений. Кроме того, не удалось найти в свободном доступе программное обеспечение, решающее перечисленные задачи в комплексе.

С учетом приведенных ранее доводов, принято решение о разработке собственного метода для решения комплекса указанных задач. В разрабатываемом методе должны учитываться и преодолеваются ограничения существующих решений.

## **1.4. Возможные сферы применения**

Данная задача актуальна для трекинг-сервисов. У таких сервисов есть несколько сценариев использования, для которых решение описанной задачи повысит качество обслуживания:

- Использование различными организациями как средство контроля своих работников. (Особенно актуально для курьерских служб и грузоперевозок). Если необходимо следить на нескольких объектах в ограниченном пространстве, излишние и неточные данные негативно сказываются на качестве предоставляемой информации.

- Использование туристами для сохранения своих маршрутов. В последствии к определенным точкам маршрутов можно добавлять комментарии и фотографии, отображающиеся на карте. В таком варианте использования может возникать несоответствие между фактическим местом, к которому относится комментарий или фотография, и его отображением на карте.

- Использование фитнес-трекеров. Фитнес-трекеры также могут накапливать данные о местоположении пользователя, которые подвержены погрешностям, что затрудняет их анализ (например, оценка пройденного расстояния).

## **1.5. Постановка задачи**

### **1.5.1. Входные параметры**

Исходными данными для задачи является набор маршрутов, связанных с наблюдаемым объектом. В наборе должен присутство-

вать хотя бы один трек. Под маршрутом понимается непустая, упорядоченная по времени последовательность данных о местоположении объекта, полученных от одного датчика.

Одна точка маршрута содержит следующую информацию:

- Широта и долгота
- Время получения
- Направление движения
- Скорость
- Точность

В разрабатываемом методе не учитывается высота объекта над уровнем моря.

### **1.5.2. Результат**

Результатом разрабатываемого метода является маршрут, представляющий собой аппроксимацию реального маршрута объекта. Точки маршрута содержат географические координаты объекта и предполагаемый момент времени, в который объект находился в данной точке.

Полученный трек можно отобразить на карте и экспортировать.

### **1.5.3. Функциональная модель задачи**

На рис. 1.2 приведена диаграмма IDEF0 функциональной модели решаемой задачи.

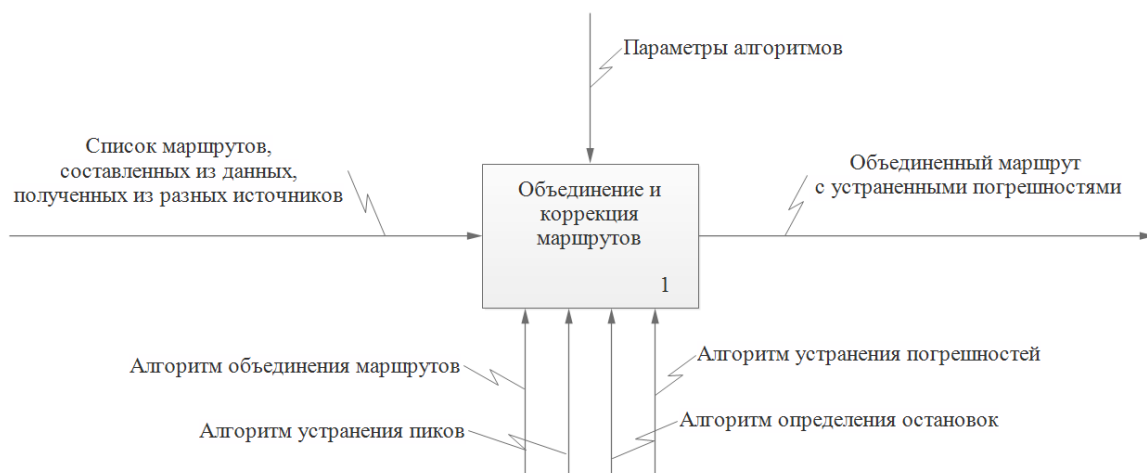


Рисунок 1.2 — Функциональная модель задачи

## 1.6. Возможные варианты решения

Для решения поставленных задач можно предложить следующую последовательность действий:

а) Объединение маршрутов. Выполнение данного этапа позволяет уменьшить число необходимых вычислений, которые в противном случае было бы необходимо проводить для каждого маршрута из исходного списка. Также возможно, что объединение маршрутов и оперирование усредненными по результатам нескольких наблюдений данных повысит точность дальнейших этапов. Для решения данной задачи можно предложить алгоритм, усредняющий значение некоторых групп координат с учетом их точности. Таким образом можно снизить влияние заведомо неточных координат на итоговый результат.

б) Устранение пиков. Данный этап целесообразно проводить до определения остановок, так как пики могут возникать в моменты времени, в которые объект был неподвижен. Наличие таких точек может помешать определению группы точек как остановки и снизить точность алгоритма. Для решения данной задачи можно предложить алгоритм, опирающийся на предполагаемые характеристики объекта. Например, можно анализировать скорость, необходимую для преодоления

расстояния между двумя точками, и сопоставлять эту скорость с максимально возможной.

в) Определение остановок и удаление избыточных координат. Входными данными для данного этапа является маршрут, в котором устранены самые значительные погрешности, что упрощает поиск точек, которые могут относиться к одному положению объекта. Для решения данной задачи можно предложить алгоритм, который разбивает по какому-либо признаку точки маршрута на группы и анализирует распределение точек в группе. Возможно существование критерия, который позволит отличить группы точек, относящиеся к месту остановки объекта.

г) Сглаживание погрешностей. Данным этапом целесообразно завершать обработку маршрута, так как предыдущие этапы могут внести дополнительную погрешность в исходные данные. Кроме того, выполнение данного этапа перед устранением остановок и пиков может внести искажения в исходные данные, что может затруднить выполнение предложенных ранее алгоритмов. На данном этапе маршрут можно рассматривать как двумерный сигнал, что позволяет использование известных фильтров для устранения шумов.

## **1.7. Обзор алгоритмов сглаживания**

### **1.7.1. Медианный фильтр**

Медианный фильтр [4] является нелинейным фильтром с конечной импульсной характеристикой. Данный фильтр широко используется для фильтрации изображений, однако его можно применять и для других типов сигналов. Фильтрация осуществляется следующим образом:

а) Значения отсчетов сигнала внутри окна фильтра сортируются по возрастанию (убыванию).

б) Выходом фильтра является значение, находящееся в середине упорядоченного списка (в случае четного размера окна, выходом является среднее значение двух отсчетов в середине списка).

Медианный фильтр является эффективным средством обработки сигналов, подверженным импульсным помехам. Дополнительным преимуществом является простота реализации. Однако данный фильтр не учитывает тип сигнала, его физическую природу и не позволяет дополнительные известные параметры обрабатываемых данных для повышения точности.

### 1.7.2. Экспоненциальное сглаживание

Экспоненциальное сглаживание [5] — способ обработки последовательности данных, применяемый для получения их сглаженного представления и для прогнозирования. Может применяться как в случае случайного процесса, так и для упорядоченного, но подверженного воздействию шумов. Алгоритм задается формулой 1.1.

$$s_t = \begin{cases} x_t, & t = 0 \\ \alpha x_t + (1 - \alpha)s_{t-1}, & t > 0 \end{cases}, \quad (1.1)$$

где  $x_t$  — значение исходной величины в момент времени  $t$ ,  
 $s_t$  — результат работы алгоритма для момента времени  $t$ ,  
 $\alpha$  — коэффициент сглаживания,  $0 < \alpha < 1$ .

Значение коэффициента сглаживания можно определять для очередной итерации, исходя из дополнительных сведений о точности значений. Применительно к рассматриваемой задаче, значение  $\alpha$  можно вычислять по значениям точности, получаемым от датчика. Однако в такой модели сложно учесть дополнительные известные параметры.

### 1.7.3. Ядерное сглаживание

Ядерное сглаживание [6] — статистический метод оценки истинного значения функции по ее зашумленным измерениям, приме-

няющийся в случае отсутствия информации о законе распределения шумов и значений функции. Данный метод предполагает использование одной из известных функций ядра (kernel functions) для получения оценки наблюдаемого параметра. Возможно вычисление параметров функции ядра по дополнительным данным о процессе, однако данный метод не предполагает использование сведений о природе процесса. Также результаты работы метода чувствительны к выбору функции ядра.

#### 1.7.4. Фильтр Калмана

Фильтр Калмана [7] — рекурсивный фильтр, оценивающий вектор состояния динамической системы, используя ряд неполных и зашумленных измерений. Зачастую размерность вектора состояния объекта превосходит размерность вектора данных наблюдения. Фильтр Калмана позволяет оценивать полное внутреннее состояние объекта.

Фильтр Калмана предназначен для рекурсивного дооценивания вектора состояния априорно известной динамической системы, то есть для расчёта текущего состояния системы необходимо знать текущее измерение, а также предыдущее состояние самого фильтра. Таким образом, фильтр Калмана, подобно другим рекурсивным фильтрам, реализован во временном, а не в частотном представлении, но в отличие от других подобных фильтров, фильтр Калмана оперирует не только оценками состояния, а еще и оценками неопределенности (плотности распределения) вектора состояния, опираясь на формулу Байеса условной вероятности.

Алгоритм работает в два этапа. На этапе прогнозирования фильтр Калмана экстраполирует значения переменных состояния, а также их неопределенности. На втором этапе, по данным измерения (полученного с некоторой погрешностью), результат экстраполяции уточняется.

Применительно к рассматриваемой задаче, фильтр Калмана позволяет учитывать модель движения, а также позволяет разделить

погрешности измерений и погрешности движения. Таким образом, для оценки и уточнения положения объекта можно использовать все доступные от датчика данные, перечисленные в пункте 1.5.1.

### 1.7.5. Выводы

Результаты сравнения перечисленных алгоритмов приведены в таблице 1.1.

Таблица 1.1 — Сравнение алгоритмов сглаживания

Алгоритм	Учет модели движения	Учет погрешностей процесса	Учет погрешностей датчика
Медианный фильтр	-	-	-
Экспоненциальное сглаживание	-	-	+
Ядерное сглаживание	-	-	+
Фильтр Калмана	+	+	+

Так как в данной задаче с помощью фильтра Калмана можно учитывать больше параметров, по сравнению с остальными рассмотренными алгоритмами, для сглаживания погрешностей маршрута решено использовать фильтр Калмана.

## 2. Конструкторский раздел

### 2.1. Объединение последовательностей координат

Входными данными алгоритма, который предлагается использовать для объединения координат от разных источников, является список последовательностей координат. На входные данные накладывается ряд ограничений:

- а) Список последовательностей координат и последовательности координат не могут быть пустыми
- б) Последовательности координат упорядочены по времени получения
- в) С каждой координатой должны быть связаны следующие данные:
  - Широта
  - Долгота
  - Время получения
  - Точность

Результатом работы алгоритма является упорядоченная по времени последовательность координат, в которой представлены усредненные данные, полученные от нескольких источников.

Описание алгоритма:

- а) Задается временное окно для определения группы координат, подлежащих объединению.
- б) Если в группу входит больше одной координаты, то она заменяется на координату, широта и долгота которой вычисляются по формуле (2.1).

$$C = \frac{\sum_{i=1}^n (\sum a - a_i) \times C_i}{\sum_{j=1}^n (\sum a - a_j)}, \quad (2.1)$$

где  $C$  — вектор координат вида  $\begin{bmatrix} x & y \end{bmatrix}^T$ , где  $x$  — долгота точки,  $y$  — широта,

$n$  — количество координат в текущем окне,



$a$  — точность координаты,

$\sum a$  — сумма всех значений точности координат в текущей группе.

Прочие параметры итоговой координаты, такие как направление движения в точке, скорость, точность и время получения являются средним арифметическим соответствующих значений в группе.

На рисунке 2.1 приведен пример объединения двух точек.

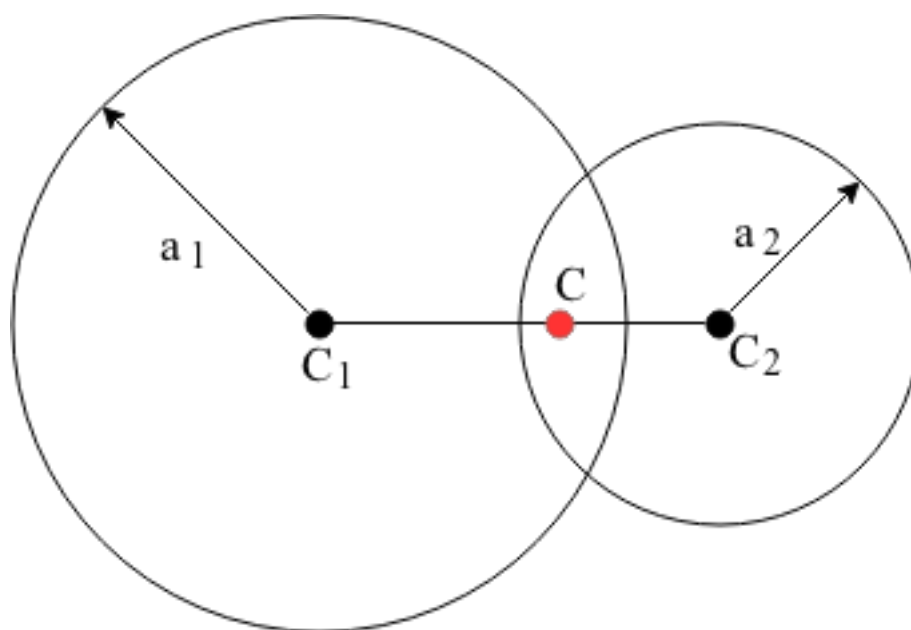


Рисунок 2.1 — Объединение координат

## 2.2. Устранение пиков

Входными данными алгоритма, который предлагается использовать для устранения пиков, является последовательность координат. На входные данные накладывается ряд ограничений:

- а) Последовательность координат упорядочена по времени
- б) С каждой координатой должны быть связаны следующие данные:
  - Широта
  - Долгота
  - Время получения

Результатом работы алгоритма является последовательность координат, из которой удалены точки, соответствующими предлагаемыми критериям пика.

Для обнаружения пиков предлагается использовать два критерия:

а) Оценка скорости объекта.

Данный критерий основан на определении расстояния между парой подряд идущих в последовательности координат и вычислении скорости, необходимой для преодоления данного расстояния. Расстояние определяется по формуле (2.2).

$$H = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2.2)$$

По известному значению расстояния  $H$  определяется теоретическая скорость объекта:

$$v = \frac{H}{T_2 - T_1}, \quad (2.3)$$

где  $T_{1,2}$  — время получения первой и второй координаты соответственно. При выполнении условия (2.4) вторая точка считается пиком и удаляется из исходной последовательности.

$$v > v_{max}, \quad (2.4)$$

где  $v_{max}$  — задаваемая предельно допустимая скорость объекта.

б) Оценка геометрических параметров участка маршрута.

Для данного критерия рассматриваются тройки идущих подряд координат (рис. 2.2). Попарно определяются расстояния между координатам в группе и проверяется условие (2.5).

$$\frac{H_1}{H_{сл}} > k \wedge \frac{H_2}{H_{сл}} > k, \quad (2.5)$$

где  $H_{1,2}$  — расстояния между центральной и крайними точками группы,

$H_{сл}$  — расстояние между крайними точками,

$k$  — задаваемый пороговый коэффициент.

Если указанное условие выполняется, средняя точка считается пиком и удаляется из последовательности.

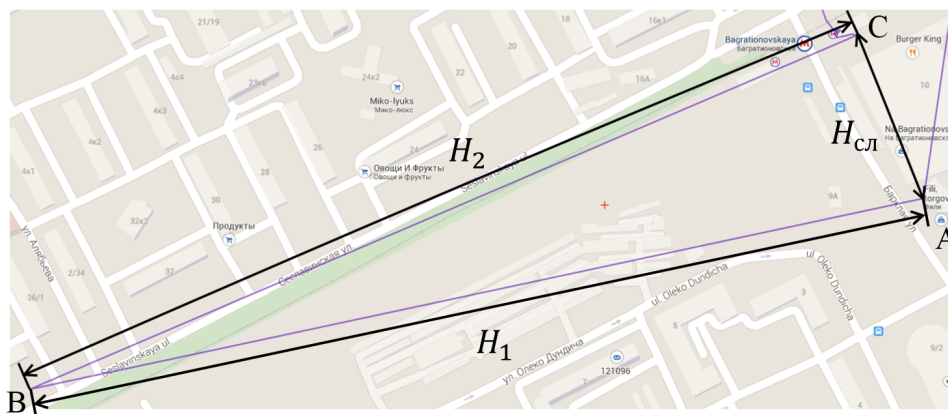


Рисунок 2.2 — Устранение пиков

### 2.3. Определение остановок

Входными данными алгоритма, который предлагается использовать для определения остановок и устранения избыточных координат, является последовательность координат. На входные данные накладывается ряд ограничений:

- а) Последовательность координат упорядочена по времени
- б) С каждой координатой должны быть связаны следующие данные:
  - Широта
  - Долгота
  - Время получения

Результатом работы алгоритма является последовательность координат, в которой группы координат, определенные как остановки, заменены одной координатой, параметры которой определяются как средние значения соответствующих параметров координат группы.

Этапы алгоритма:

- а) Задается временное окно, позволяющее разбить исходную последовательность на группы точек для анализа. Предполагается, что группы будут содержать более одной точки, однако слишком большие периоды окна могут привести к ошибке распознавания из-за уве-

личения размеров групп и попадания точек, относящихся к началу движения объекта.

б) Для каждой группы координат вычисляется координата со средними значениями широты и долготы.

в) Проверяется условие

$$\frac{\sum_{i=1}^n H_i}{n} > H_{min}, \quad (2.6)$$

где  $H_i$  — расстояние от  $i$ -ой координаты точки до средней точки,  $H_{min}$  — задаваемое пороговое значение среднего расстояния,  $n$  — количество координат в группе.

г) Если условие выполняется, группа координат заменяется средним значением.

д) Подряд идущие группы координат, определенные как остановки, заменяются единственным средним значением.

На рисунке 2.3 приведен пример группы координат. Черным цветом обозначены координаты, попавшие в рассматриваемую группу, синим — координаты, отсеченные окном, красным цветом обозначена координата со средними значениями широты и долготы.

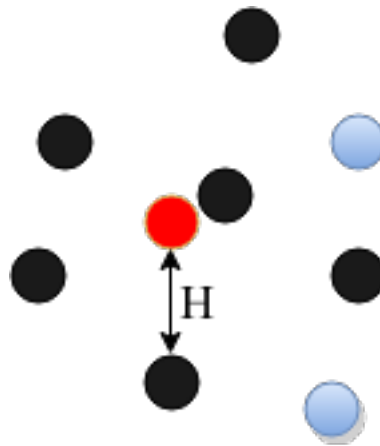


Рисунок 2.3 — Определение остановок

## 2.4. Сглаживание погрешностей маршрута

### 2.4.1. Построение математической модели с использованием фильтра Калмана

Для использования фильтра Калмана необходимо, чтобы исследуемый процесс описывался следующим образом [8]:

$$s_k = A_k s_{k-1} + B_k u_{k-1} + w_k, \quad (2.7)$$

$$z_k = H_k s_k + \nu_k \quad (2.8)$$

В формуле (2.7)  $s_k \in \mathbb{R}^n$ -вектор состояния процесса,  $A$  - матрица размерностью  $n \times n$ , описывающая переход наблюдаемого процесса из состояния  $s_{k-1}$  в состояние  $s_k$ . Вектор  $u \in \mathbb{R}^l$  описывает управляющие воздействия на процесс. Матрица  $B$  размерностью  $n \times l$  отображает вектор управляющих воздействий  $u$  в изменение состояния  $s$ .  $w_k \in \mathbb{R}^n$  является случайной величиной, описывающей погрешности исследуемого процесса, причем  $p(w) \sim N(0, Q)$ , где  $Q$  - ковариационная матрица погрешностей процесса. Формула (2.8) описывает измерения случайного процесса.  $z_k \in \mathbb{R}^m$ - вектор измеряемого состояния процесса, матрица  $H$  размерностью  $m \times n$  отображает состояние процесса  $s_k$  в измерение процесса  $z_k$ .  $\nu_k \in \mathbb{R}^m$  - случайная величина, характеризующая погрешности измерений, причем  $p(\nu) \sim N(0, R)$ , где  $R$  - ковариационная матрица погрешностей измерений.

Так как исследуется процесс движения объекта, уравнение состояния составляется исходя из уравнения движения тела  $\bar{r}_k = \bar{r}_{k-1} + \bar{v}_{k-1} dt_k + \bar{a}_{k-1} \frac{dt_k^2}{2}$ . Кроме того, отсутствует дополнительная информация о процессе движения, поэтому считается, что управляющее воздействие  $Bu$  равно 0. За состояние процесса принят вектор

$$s_k = \begin{bmatrix} x_k \\ y_k \\ v_k^x \\ v_k^y \end{bmatrix}, \text{ где } x_k, y_k - \text{координаты объекта, } v_k^x, v_k^y - \text{проекции скорости}$$

объекта. Таким образом, для рассматриваемого процесса уравнение (2.7) принимает следующий вид:

$$s_k = A_k s_{k-1} + G_k a_k, \quad (2.9)$$

где

$$A_k = \begin{bmatrix} 1 & 0 & dt_k & 0 \\ 0 & 1 & 0 & dt_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.10)$$

$$a_k = \begin{bmatrix} a_k^x \\ a_k^y \end{bmatrix}, \quad (2.11)$$

$$G_k = \begin{bmatrix} \frac{dt_k^2}{2} & 0 \\ 0 & \frac{dt_k^2}{2} \\ dt_k & 0 \\ 0 & dt_k \end{bmatrix} \quad (2.12)$$

В данной модели ускорение объекта рассматривается как случайная погрешность процесса. Принимаются следующие допущения:

а) Ускорения по разным осям являются независимыми случайными величинами.

б)  $p(a_k^{x,y}) \sim N(0, \sigma_a)$ , то есть ускорения распределены нормально с нулевым математическим ожиданием и некоторым среднеквадратичным отклонением  $\sigma_a$ , которое считается известным.

Данные допущения приняты из следующих соображений:

а) Нет оснований полагать, что по значению одной компоненты вектора ускорения можно сделать вывод о значении другой компоненты.

б) Большую часть времени объект движется равномерно. В качестве возможных источников координат рассматриваются пешеходы и автомобили. Ненулевое ускорение обычно связано либо с изменением направления движения, что происходит за небольшой промежуток

времени (от 5 до 15 секунд), либо с остановкой или началом движения, причем разгон или торможение до требуемой скорости также происходят сравнительно быстро (в случае пешеходов практически моментально, автомобилей - в пределах 10-15 секунд). Ускорение, возникающее в других ситуациях (например, обгон или замедление для оценки окружающей обстановки), также отличается от нуля в течение небольших промежутков времени.

Таким образом, в формуле (2.9) член  $G_k a_k$  выполняет роль члена  $w_k$  из формулы (2.7). Для дальнейших вычислений необходимо получить значение матрицы  $Q$ .

$$\begin{aligned} Q_k = cov(w_k) &= \mathbb{E}(w_k w_k^T) - \mathbb{E}(w_k) \mathbb{E}(w_k^T) = \\ &= \mathbb{E}(G_k a_k a_k^T G_k^T) = G_k \mathbb{E}(a_k a_k^T) G_k^T \end{aligned} \quad (2.13)$$

Так как компоненты вектора  $a_k$  (2.11) являются независимыми случайными величинами, то  $\mathbb{E}(a_k a_k^T) = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix}$ . Следовательно, формула (2.13) принимает следующий вид:

$$Q_k = G_k G_k^T \sigma_a^2 \quad (2.14)$$

Вектор измерения  $z_k$  для данной задачи представляется следующим образом:

$$z_k = \begin{bmatrix} \chi_k \\ \gamma_k \\ \vartheta_k^x \\ \vartheta_k^y \end{bmatrix}, \quad (2.15)$$

где  $\chi_k, \gamma_k$  - координаты объекта, полученные от датчика,  $\vartheta_k^x, \vartheta_k^y$  - скорость объекта, полученная от датчика. Матрица  $H$  в формуле (2.8) принимается равной единичной матрице размерностью  $4 \times 4$ , так как в рамках данной задачи считается, что измерение есть линейная комбинация вектора состояния и некоторых случайных погрешностей. Ковариационная матрица погрешности измерений  $R$  считается

заданной. Один из возможных вариантов ее вычисления - использование данных о предполагаемой точности измерения, получаемых от датчика. Тогда  $R$  задается соотношением (2.16).

$$R = \begin{bmatrix} p_x^2 & 0 & 0 & 0 \\ 0 & p_y^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (2.16)$$

где  $p_{x,y}$  — точность определения координаты по горизонтальной и вертикальной оси соответственно.

#### 2.4.2. Применение фильтра Калмана к построенной модели

Для применения фильтра необходимо ввести следующие понятия:

- $\hat{s}_k$ - апостериорная оценка состояния объекта в момент  $k$ , полученная по результатам наблюдений вплоть до момента  $k$  включительно.

- $\hat{\tilde{s}}_k$ - нескорректированная апостериорная оценка состояния объекта в момент времени  $k$ .

- $P_k$ - апостериорная ковариационная матрица ошибок, задающая оценку точности полученной оценки вектора состояния и включающая в себя оценку дисперсий погрешности вычисленного состояния и ковариации, показывающие выявленные взаимосвязи между параметрами состояния системы.

- $\tilde{P}_k$ - нескорректированная апостериорная ковариационная матрица ошибок.

Матрица  $P_0$  задается как нулевая, так как считается, что известно начальное положение объекта.

Одна итерация фильтра Калмана состоит из двух этапов: экстраполяция и коррекция.



а) На этапе экстраполяции вычисляется оценка  $\hat{s}_k^\sim$  по оценке вектора состояния  $\hat{s}_{k-1}$  и ковариационная матрица ошибок  $P_k^\sim$  по следующим формулам:

$$\hat{s}_k^\sim = A_k \hat{s}_{k-1}, \quad (2.17)$$

$$P_k^\sim = A P_{k-1} A^T + Q_k, \quad (2.18)$$

где матрица  $A_k$  известна из формулы (2.10), матрица  $Q_k$  вычисляется по формуле (2.14).

б) На этапе коррекции вычисляется матрица коэффициентов усиления  $K_k$  по следующей формуле:

$$K_k = P_k^\sim H^T (H P_k^\sim H^T + R)^{-1}, \quad (2.19)$$

где  $R, H$  считаются известными.  $K_k$  используется для коррекции оценки состояния объекта  $\hat{s}_k^\sim$  и ковариационной матрицы ошибок  $P_k^\sim$  следующим образом:

$$\hat{s}_k = \hat{s}_k^\sim + K_k (z_k - H \hat{s}_k^\sim), \quad (2.20)$$

$$P_k = (I - K_k H) P_k^\sim, \quad (2.21)$$

где  $I$  - единичная матрица.

Следует заметить, что для использования указанных выше соотношений необходимо, чтобы единицы измерений параметров объекта были согласованы. Однако в исходных данных широта и долгота приводятся в угловых координатах, а скорость — в метрических. Кроме того, ускорение для расчета ошибки процесса также удобнее задавать в метрических единицах.

### 2.4.3. Согласование единиц измерения

Для использования приведенных соотношений необходимо согласовать единицы измерения данных о местоположении. Датчик предоставляет следующие сведения:

- а) Широта и долгота в угловых координатах
- б) Скорость в м/с

- в) Направление движения в угловых градусах
- г) Точность в метрах

Таким образом, необходимо выбрать единицы измерения для проведения вычислений. Так как основной задачей является уточнение широты и долготы, было решено проводить вычисления в угловых координатах. Отсюда возникает задача перевода скорости и точности в угловые единицы. Задача определения координат точки по известным координатам исходной точки, направлению перемещения и горизонтальному расстоянию называется прямой геодезической задачей [9]. Данная задача усложняется тем, что Земля не является идеальным эллипсоидом.

Для перевода скорости и ускорения в угловые единицы решено использовать формулы Винченти [10]. Вводятся следующие обозначения:

- $a$  — длина большой полуоси эллипсоида
- $f$  — коэффициент уплощения
- $b = (1 - f) \times a$  — длина малой полуоси эллипсоида
- $\phi_{1,2}$  — широта точек
- $L_{1,2}$  — долгота точек
- $U_1 = \arctan[(1 - f) \tan \phi_1]$  — параметризованная широта
- $U_2 = \arctan[(1 - f) \tan \phi_2]$  — параметризованная широта
- $L = L_2 - L_1$  — разность долготы двух точек
- $\lambda_1, \lambda_2$  — параметризованная долгота точек
- $\alpha_{1,2}$  — азимуты в точках
- $\alpha$  — азимут на экваторе
- $s$  — расстояние между точками на эллипсоиде
- $\sigma$  — длина дуги между двумя точками на вспомогательной сфере

$$U_1 = \arctan((1 - f) \tan \phi_1) \quad (2.22)$$

$$\sigma_1 = \arctan\left(\frac{\tan U_1}{\cos \alpha_1}\right) \quad (2.23)$$

$$\sin \alpha = \cos U_1 \sin \alpha_1 \quad (2.24)$$

$$\cos^2 \alpha = 1 - \sin^2 \alpha \quad (2.25)$$

$$u_2 = \cos^2 \alpha \frac{a^2 - b^2}{b^2} \quad (2.26)$$

$$A = 1 + \frac{u^2}{16384} \{4096 + u^2[-768 + u^2(320 - 175u^2)]\} \quad (2.27)$$

$$B = \frac{u^2}{1024} \{256 + u^2[-128 + u^2(74 - 47u^2)]\} \quad (2.28)$$

После проведения расчётов по формулам (2.22-2.28) необходимо уточнить значение  $\sigma$  по формулам (2.29-2.31). Начальное значение  $\sigma = \frac{s}{bA}$ . Итерации останавливаются при достижении требуемой точности.

$$2\sigma_m = 2\sigma_1 + \sigma \quad (2.29)$$

$$\begin{aligned} \Delta\sigma = B \sin \sigma \{ & \cos(2\sigma_m) + \frac{1}{4}B[\cos \sigma(-1 + 2 \cos^2(2\sigma_m)) - \\ & - \frac{1}{6}B \cos(2\sigma_m)(-3 + 4 \sin^2 \sigma)(-3 + 4 \cos^2(2\sigma_m))]\} \end{aligned} \quad (2.30)$$

$$\sigma = \frac{s}{bA} + \Delta\sigma \quad (2.31)$$

После вычисления  $\sigma$  определяются результирующие значения по формулам (2.32-2.37).

$$\phi_2 = \arctan\left(\frac{\sin U_1 \cos \sigma + \cos U_1 \sin \sigma \cos \alpha_1}{(1 - f)\sqrt{\sin^2 \alpha + (\sin U_1 \sin \sigma - \cos U_1 \cos \sigma \cos \alpha_1)^2}}\right) \quad (2.32)$$

$$\lambda = \arctan\left(\frac{\sin \sigma \sin \alpha_1}{\cos U_1 \cos \sigma - \sin U_1 \sin \sigma \cos \alpha_1}\right) \quad (2.33)$$

$$C = \frac{f}{16} \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)] \quad (2.34)$$

$$\begin{aligned} L = \lambda - (1 - C)f \sin \alpha \{ & \sigma + C \sin \sigma [\cos(2\sigma_m) + \\ & + C \cos \sigma(-1 + 2 \cos^2(2\sigma_m))]\} \end{aligned} \quad (2.35)$$

$$L_2 = L + L_1 \quad (2.36)$$

$$\alpha_2 = \arctan\left(\frac{\sin \alpha}{-\sin U_1 \sin \sigma + \cos U_1 \cos \sigma \cos \alpha_1}\right) \quad (2.37)$$

Данные формулы способны обеспечить точность до 0.5 мм, что позволяет практически исключить дополнительные погрешности в процессе фильтрации.

Перевод скорости осуществляется следующим образом:

а) Определяются проекции скорости на координатные оси (2.38, 2.39).

$$v_x = v \times \cos \alpha, \quad (2.38)$$

$$v_y = v \times \sin \alpha, \quad (2.39)$$

где  $v$  — скорость объекта,

$\alpha$  — направление движения объекта.

б) По формулам Винченти вычисляются координаты точек, расстояния до которых от точки местоположения объекта определяется значением  $v_x$  по горизонтальной оси и  $v_y$  по вертикальной.

в) За значение скорости в угловых единицах принимается разность соответствующих координат итоговых и исходной точки.

Перевод проекций точности на координатные оси осуществляется аналогичным способом.

## 2.5. Метод уточнения и построения маршрута

Для решения рассматриваемой задачи предлагается метод, использующий приведенные ранее алгоритмы. Функциональная модель метода представлена в виде диаграммы IDEF0 на рис. 2.4.



Рисунок 2.4 — Функциональная модель метода

Входными данными для метода является список последовательностей координат. Результатом работы метода является последовательность координат с устраненными пиками, избыточными точками и сглаженными погрешностями.

Ограничения метода представляют собой объединение ограничений используемых в методе алгоритмов:

- а) Список последовательностей координат и последовательности координат не могут быть пустыми
- б) Последовательности координат упорядочены по времени получения
- в) С каждой координатой должны быть связаны следующие данные:
  - Широта и долгота в угловых координатах
  - Время получения координаты
  - Точность в метрах
  - Скорость в м/с
  - Направление движения в градусах относительно

## 2.6. Структура разрабатываемого ПО

Общая структура разрабатываемого программного обеспечения представлена на рис. 2.5 в виде UML диаграммы пакетов.

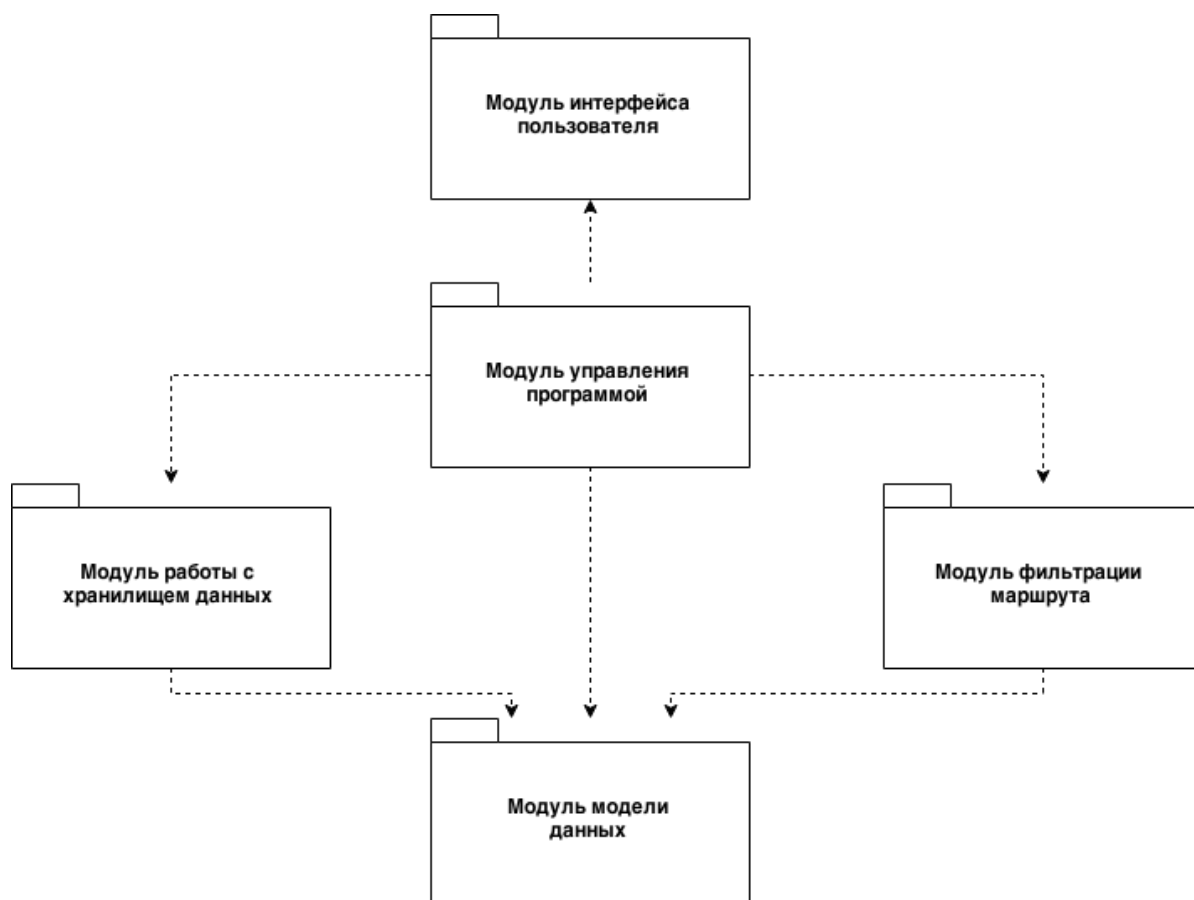


Рисунок 2.5 — Структура ПО

Предложенная структура позволяет независимо реализовывать компоненты программы, решающие разные задачи. Выделение модуля реализации предложенного метода позволяет провести проверку работы метода до завершения работы над всей программой и дает возможность повторного использования разрабатываемого компонента. Выделение модуля работы с хранилищем данных позволяет абстрагироваться от способа представления данных и предоставляет возможность расширения поддерживаемых входных и выходных форматов данных.

## 2.7. Диаграмма классов метода

Для реализации метода предполагается использовать структуру классов, приведенную на рис. 2.6 в виде UML диаграммы классов.

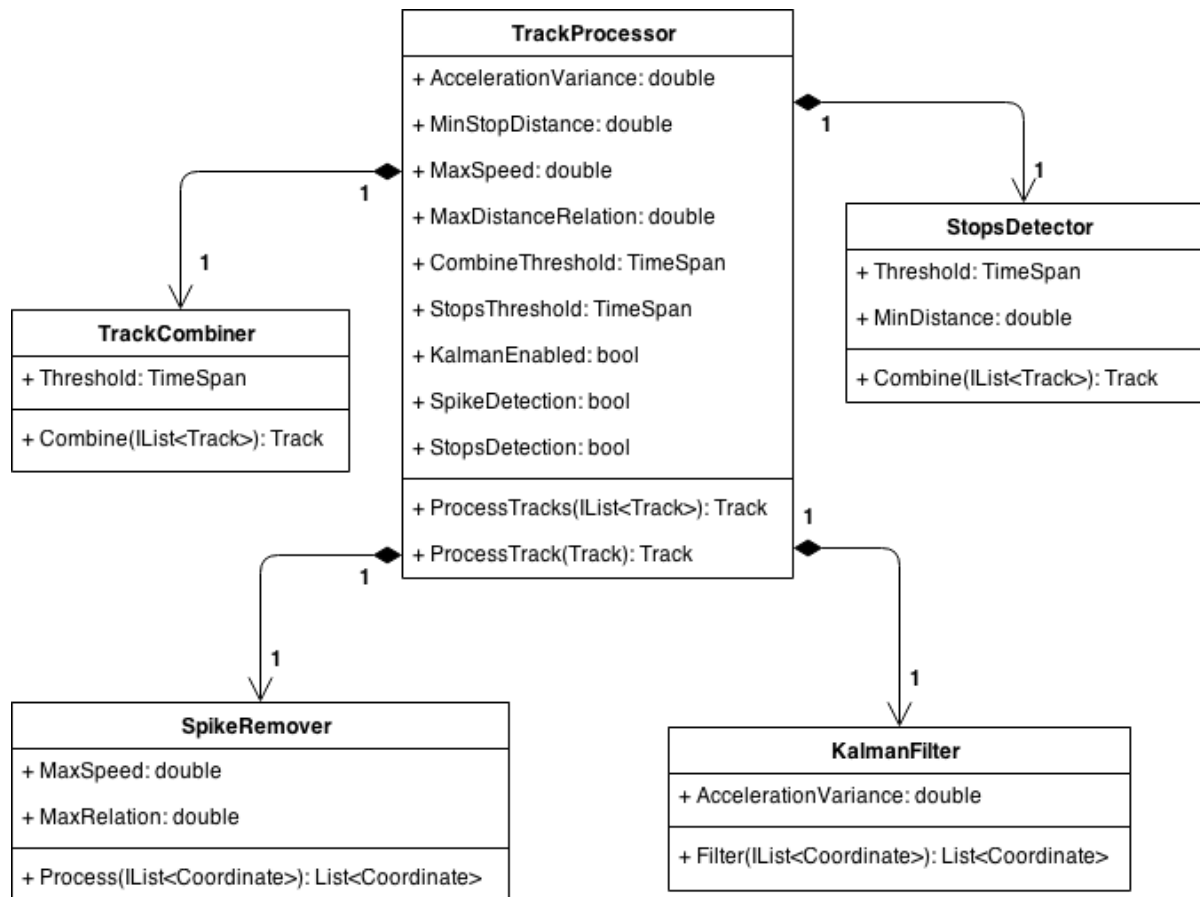


Рисунок 2.6 — Диаграмма классов модуля обработки маршрута

Класс `TrackProcessor` является фасадом [11] модуля обработки маршрута и предоставляет единую точку управления параметрами обработки для остальных модулей.

### **3. Технологический раздел**

#### **3.1. Выбор средств разработки**

##### **3.1.1. Выбор целевой платформы**

Программный продукт разрабатывается для устройств под управлением ОС Windows. Данный выбор обусловлен широкой распространенностью ОС Windows и большим количеством средств разработки для данной платформы, что дает возможность выбирать язык программирования и сопутствующие инструменты, ориентируясь на возможность простого и надежного решения рассматриваемой задачи, а не на ограничения целевой платформы.

##### **3.1.2. Выбор языка программирования**

Для написания программного продукта используется язык C#, так как он сочетает в себе возможности функционального и объектно-ориентированного подходов. С помощью функционального подхода удобно описывать алгоритмы и математические выражения, необходимые для решения поставленной задачи. Использование функциональных средств существенно упрощает описание операций над последовательностями входных данных. В то же время, применение объектно-ориентированных средств позволяет построение приложения с графическим интерфейсом пользователя с учетом известных рекомендаций и правил. Дополнительным преимуществом языка C# является большая база документации MSDN по самому языку и стандартным библиотекам.

##### **3.1.3. Выбор среды программирования и отладки**

Основной средой разработки для языка C# является Microsoft Visual Studio. В данный момент Visual Studio Community Edition бесплатно предоставляется для индивидуальных разработчиков, при этом возможности данной версии не являются урезанными по сравнению с Visual Studio Professional. Visual Studio предоставляет средства



удобной организации модулей, составляющих программу, широкие возможности для отладки приложения, инструментарий для создания приложений с графическим интерфейсом и поддержку автоматизированного тестирования. Так же поддерживается интеграция с системами контроля версий, присутствуют инструменты для статического анализа исходного кода. Приведенные возможности упрощают разработку приложения и способствуют повышению качества исходного кода.

#### **3.1.4. Выбор средства создания интерфейса**

#### **3.1.5. Средство для работы с картой.**

Для наглядного представления входных данных и результатов было решено использовать картографический компонент. Используется библиотека GMap.NET. Выбор библиотеки обусловлен следующими причинами:

- Данная библиотека является продуктом с открытым исходным кодом и предоставляется по лицензии MIT, что позволяет использовать ее без ограничений.
- Поддерживается Windows Presentation Foundation — рекомендуемое средство построения пользовательского интерфейса на платформе .NET.
- Поддерживаются как карты из онлайн-источников, так и карты из локального хранилища.
- Предоставляется возможность отображения на карте маркеров и линий с подписями.

В программе используются растровые карты Google Maps.

### **3.2. Система контроля версий**

В процессе разработки программы использовалась система контроля версий Git. Система контроля версий позволяет вносить в проект атомарные изменения, направленные на решения каких-либо

задач. В случае обнаружения ошибок или изменения требований, внесенные изменения можно отменить. Кроме того, с помощью системы контроля версий решается вопрос резервного копирования.

Особенности Git:

- данная система контроля версий является децентрализованной, что позволяет иметь несколько независимых резервных копий проекта
- поддерживается хостингом репозитория GitHub
- поддерживается средой разработки Visual Studio
- предоставляет широкие возможности для управления изменениями проекта и просмотра истории изменений

### **3.3. Физическая структура ПО**

Разработанное ПО разделено на несколько управляемых динамических библиотек и один исполняемый файл:

- Domain.dll — библиотека, содержащая определения типов данных предметной области, а также средства чтения и записи файлов с данными
- Filter.dll — библиотека, содержащая реализацию алгоритмов, используемых разработанным методом
- Analysis.dll — библиотека, содержащая средства, используемые для экспериментальной проверки применимости разработанного метода
- TrackFilter.exe — исполняемый файл, в котором определен интерфейс пользователя и логика взаимодействия с указанными выше библиотеками

### **3.4. Требования к вычислительной системе**

Для реализации ПО использовались средства .NET Framework 4.5. Данная версия поставляется вместе с операционными системами семейства Windows, начиная с версии 7.

### 3.5. Форматы данных

Для хранения входных и выходных данных используются XML-документы, соответствующие приведенной XSD-схеме:

```
<xs:schema attributeFormDefault="unqualified"
            elementFormDefault="qualified"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Tracks">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Coordinates"
                    maxOccurs="unbounded"
                    minOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Coordinate"
                          maxOccurs="unbounded"
                          minOccurs="2">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element type="xs:float"
                                name="Latitude"/>
                    <xs:element type="xs:float"
                                name="Longitude"/>
                    <xs:element type="xs:float"
                                name="Speed"/>
                    <xs:element type="xs:byte"
                                name="Accuracy"/>
                    <xs:element type="xs:short"
                                name="Azimuth"/>
                    <xs:element type="xs:string"
                                name="Time"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Способ представления в виде XML-документов был выбран из-за следующих достоинств

- Легкость редактирования
- Возможность формирования документов на разных платформах
- Каждый файл соответствует полному маршруту, с которым может работать программа
- Простота проверки корректности представления данных

Указанные особенности позволили упростить сбор и подготовку тестовых данных. Документы формировались на различных устройствах (таких как смартфон, планшетный компьютер и ноутбук) под управлением различных ОС. Использование XML-документов позволило собирать информацию с разных устройств независимо и не реализовывать клиент-серверную системы для подготовки данных. Кроме того, для дальнейшего использования подготовленной информации нет необходимости преобразовывать ее к иному виду, например, записывать в БД, доступную программе. Также для разработанной программы не предусмотрены сложные сценарии доступа к данным, требующие сортировку и отсеивание по каким-либо критериям. Таким образом, использование для хранения данных СУБД в данном случае не оправдано.

### **3.6. Тестирование программного продукта**

Тестирование осуществлялось на наборах данных, полученных с помощью разных источников. Для сбора использовались смартфоны, планшетные компьютеры и внешние GPS-датчики. В ходе тестирования проверялась корректность считывания входных данных, корректность экспорта выходных данных и отсутствие заведомо неверных результатов при обработке маршрутов. Так как критерии корректности слабо формализованы, тестирование проводилось вручную методом «черного ящика».

### **3.7. Установка программного обеспечения**

Так как у разрабатываемого ПО нет зависимостей от внешних служб и программ, установленных в операционной системе, а также отсутствует необходимость в модификации системных и пользовательских настроек, было принято решение не разрабатывать инсталлятор. Установка производится копированием папки с программой в нужную директорию. Исполняемым файлом программы является TrackFilter.exe.

### **3.8. Руководство пользователя**

#### **3.8.1. Управление картой**

Управление картой осуществляется с помощью мыши. Для масштабирования используется колесо мыши. Для изменения участка карты необходимо перемещать курсор с нажатой правой кнопкой мыши.

#### **3.8.2. Выбор маршрута**

Для выбора маршрута используется пункт меню File->Open Track. В открывшемся диалоговом окне следует указать путь к xml-файлу, содержащему данные маршрута. Пример выбора маршрута приведен на рис. 3.1.

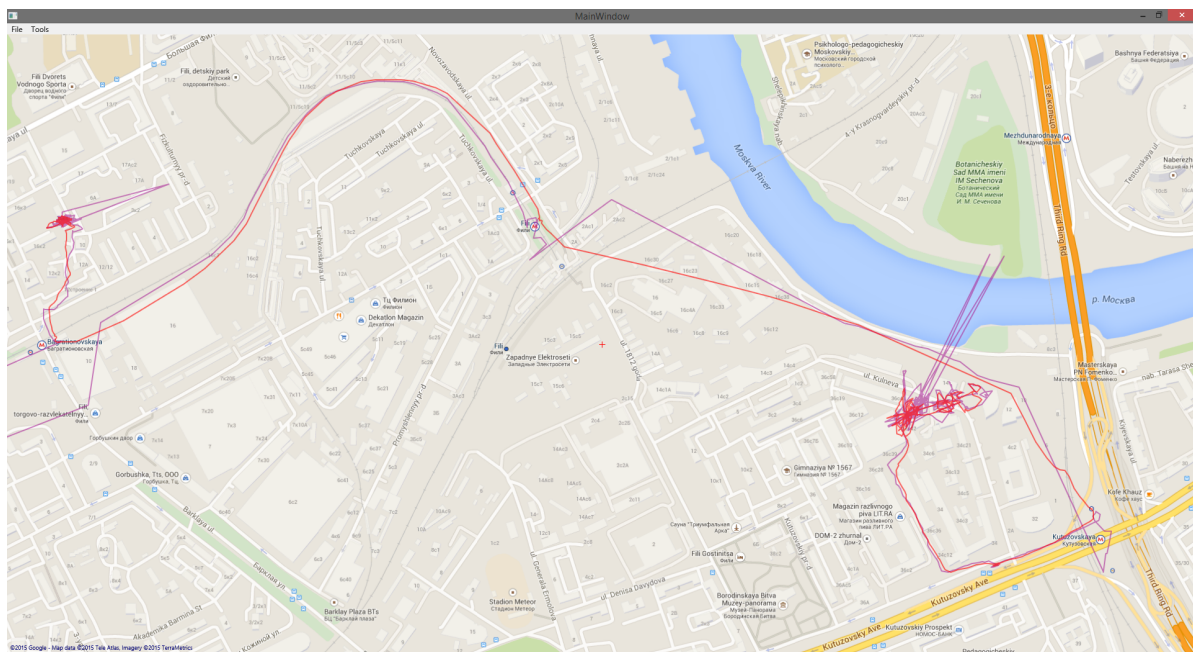


Рисунок 3.1 — Выбор маршрута

Для удобства пользователя файлы, формат которых не соответствует ожидаемому, не отображается.

### 3.8.3. Информация о маршруте

Для вызова всплывающей подсказки, содержащей название маршрута и количество точек в маршруте, необходимо расположить курсор мыши над любым видимым участком маршрута.

### 3.8.4. Экспорт результата

Экспорт результата становится доступен после того, как был выбран и обработан маршрут. Экспорт осуществляется через пункт меню File -> Save Result. В открывшемся диалогом окне следует выбрать путь для сохранения файла с результатом и ввести название.

### 3.8.5. Настройки

Окно настроек программы (рис. 3.2) открывается с помощью пункта меню Tools->Settings. В данном окне осуществляется управления параметрами метода и отображения данных на карте.

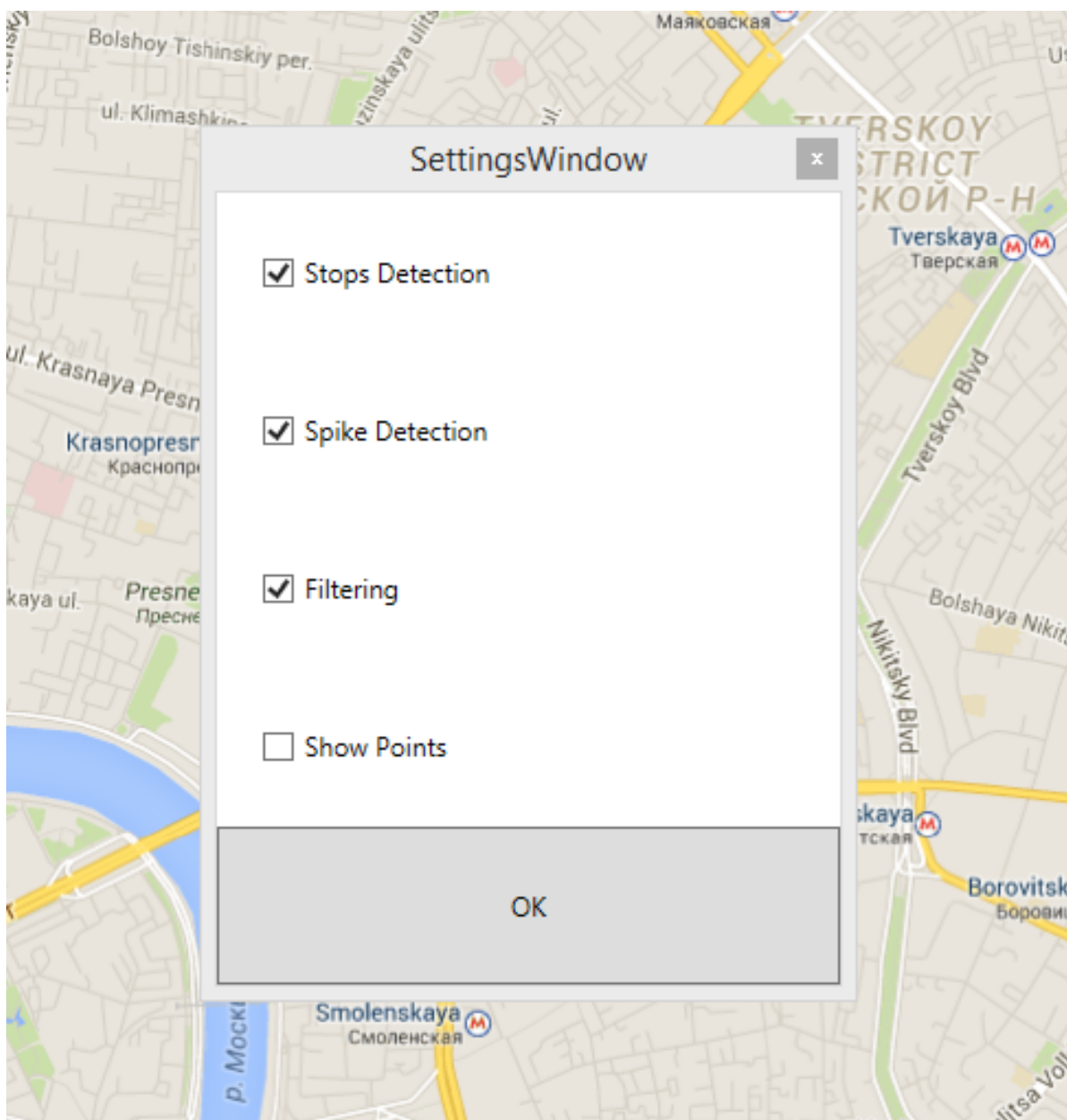


Рисунок 3.2 — Окно настроек программы

### 3.9. Пример работы программы

Пример работы программы на тестовом маршруте приведен на рис. 3.3.

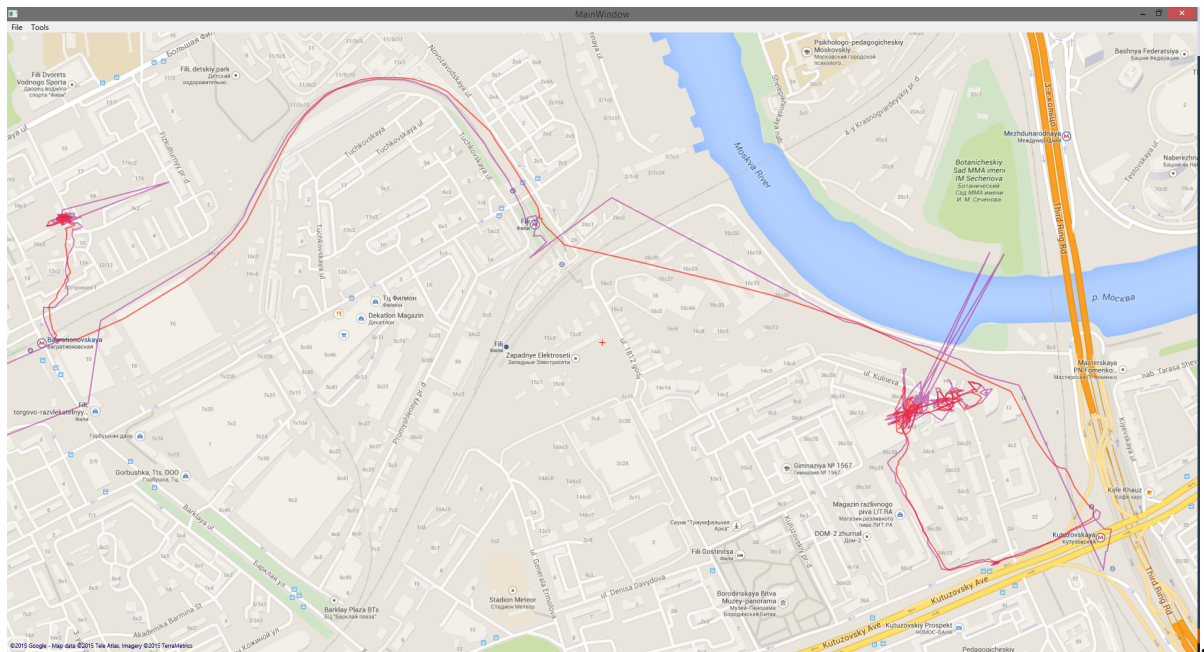


Рисунок 3.3 — Пример работы программы

Фиолетовым цветом обозначен исходный маршрут, красным - ИТОГОВЫЙ.



## 4. Экспериментальный раздел

### 4.1. Постановка эксперимента

#### 4.1.1. Описание эксперимента

Эксперимент проводится для подтверждения применимости предложенного метода к рассматриваемой задаче.

В ходе эксперимента рассматривается отклонение исходного маршрута и маршрута, полученного в результате работы предложенного метода, от эталонного маршрута.

#### 4.1.2. Размер выборки

Перед началом исследования необходимо определить размер выборки. Следует отметить, что элементом выборки в данном случае является один кадр. Размер выборки обусловлен точностью определения параметров метода и необходимой сложностью исследования. Для оценки размера выборки можно воспользоваться законом больших чисел, записанном в форме 2-го неравенства Чебышева (4.1).

$$P\{|X - MX| \geq \epsilon\} \leq \frac{\sigma^2}{N\epsilon^2}, \quad (4.1)$$

где  $N$  – размер выборки,  $X$  – случайная величина,  $\epsilon$  – требуемая точность,  $\sigma^2$  – дисперсия случайной величины,  $MX$  – математическое ожидание случайной величины. Критерий качества лежит в диапазоне от 0 до 1, поэтому дисперсия качества оцениваемого параметра не превосходит 0.25. Для получения оценки объёма выборки можно принять  $\sigma^2 = 0.25$ . Задав достоверность метода  $P = 0,95$ , а требуемую точность  $\epsilon = 0.05$ , можно получить значение необходимого объёма выборки (4.2) – (4.4).

$$N \geq \frac{\sigma^2}{P\{|X - MX| \geq \epsilon\}\epsilon^2} \quad (4.2)$$

$$N \geq \frac{0.25}{(1 - 0.95) \times 0.5^2} \quad (4.3)$$

$$N \geq 2000 \quad (4.4)$$

Следовательно, с ошибкой не более 5% и вероятностью 95% необходимый размер выборки равен 2000.

#### 4.1.3. Данные для эксперимента

Для эксперимента были подготовлены входные 4 классов :

- а) Пеший маршрут + метро
- б) Автомобильный маршрут
- в) Маршрут наземного общественного транспорта
- г) Велосипедный маршрут

Перечисленные классы покрывают основные способы передвижения в черте города.

Количество координат в каждом из перечисленных маршрутов превышает 2000, что удовлетворяет установленному объему выборки.

Эталонные маршруты для сравнения были составлены вручную с использованием инструментов Яндекс.Карт.

На рис. 4.1 приведен пример данных для эксперимента. Зеленым цветом изображен исходный маршрут, красным – итоговый, синим – эталонный.

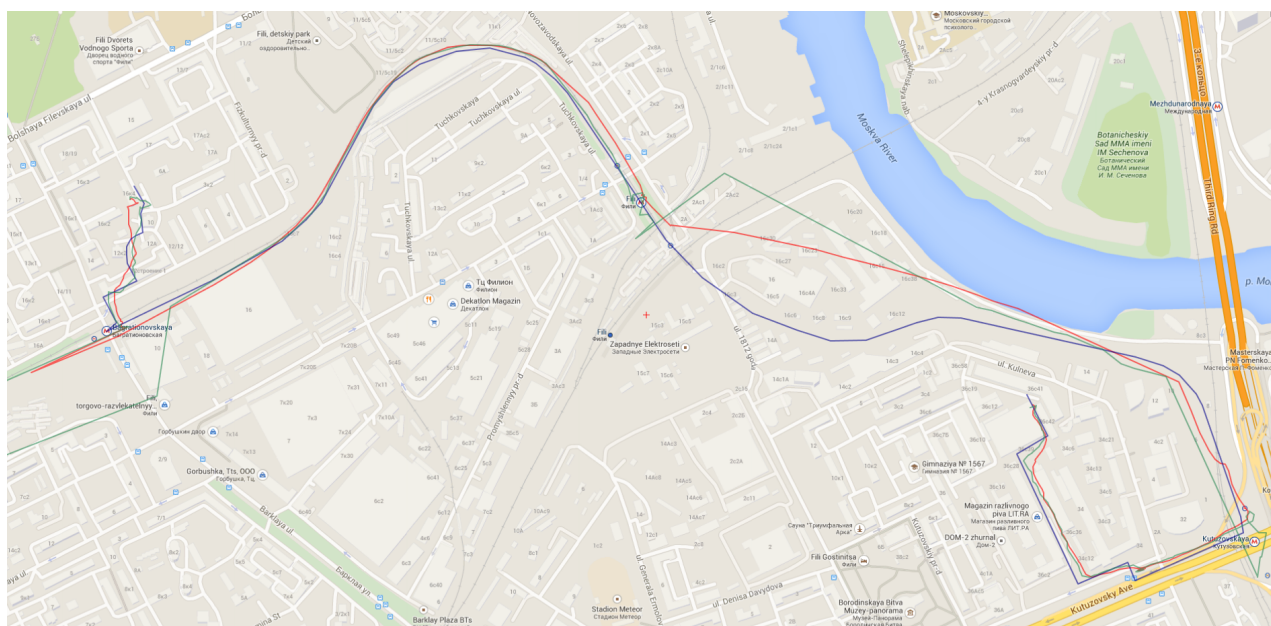


Рисунок 4.1 — Данные для эксперимента

## 4.2. Результат эксперимента

На рис. 4.2 приведена гистограмма, построенная по результатам эксперимента. На гистограмме цветными столбиками показано математическое ожидание отклонения маршрутов от эталонного. Черными столбиками отражается его среднеквадратичное отклонение.



Рисунок 4.2 — Гистограмма с результатами эксперимента

По приведенным результатам можно сделать вывод, что для трех классов предложенный метод снижает отклонение маршрута и уменьшает дисперсию отклонения. Отсутствие существенного улучшения в случае автомобильного маршрута объясняется более высокой точностью исходных данных, по сравнению с остальными рассмотренными случаями. Однако из приведенных в таблице 4.1 численных результатов можно сделать вывод, что наблюдается повышение точности и в случае автомобильного маршрута.

Таблица 4.1 — Результаты эксперимента

Маршрут	Отклонение исходного маршрута, м	Отклонение итогового маршрута, м	Коэффициент снижения отклонения	Коэффициент уменьшения интервала ошибки
Пешие участки + метро	$22,7263 \pm 205,307$	$11,4884 \pm 12,6923$	1,9782	16,1757
Наземный общественный транспорт	$30,0419 \pm 147,8227$	$12,9868 \pm 44,1246$	2,31326	3,35012
Автомобиль	$10,6518 \pm 10,6853$	$10,0099 \pm 10,0372$	1,06413	1,06457
Велосипед	$17,9114 \pm 115,615$	$11,2262 \pm 70,577$	1,5955	1,63814

### 4.3. Выводы

В результате эксперимента была подтверждена применимость предложенного метода. Рассмотренные классы входных данных покрывают основные случаи возможных источников входных данных для данной задачи.

Наилучшие результаты метод показывает для данных с высокой степенью погрешности, что объясняется устранением пиков, однако наблюдается повышение точности и для других ситуаций.

## 5. Организационно-экономический раздел

### 5.1. Определение этапов и содержания работ. Расчёт трудоемкости.

Планирование длительности этапов и содержания работ осуществляется в соответствии с ГОСТ 19.102-77 [12].

На стадии разработки технического задания определены следующие стадии и этапы работ, отраженные в таблице:

Таблица 5.1 — Этапы выполнения проекта

Стадия разработки	Этап работ	Содержание работ
1. Техническое задание	Обоснование необходимости разработки программы	Постановка задачи, сбор исходных материалов
	Научно-исследовательские работы	Предварительный выбор методов решения задач
	Разработка и утверждение технического задания	Определение требований к программе; определение стадий, этапов и сроков разработки программы и документации к ней.
2. Эскизный проект	Разработка эскизного проекта	Предварительная разработка структуры входных и выходных данных; уточнение методов решения задачи

3. Технический проект	Разработка технического проекта; утверждение технического проекта	Уточнение структуры входных и выходных данных; разработка алгоритма решения задачи; определение формы представления входных и выходных данных; разработка структуры программы; разработка пояснительной записки; согласование и утверждение технического проекта.
4. Рабочий проект	Разработка программы	Разработка и реализация ядра системы; разработка и реализация пользовательского уровня; тестирование и отладка программы.

5. Внедрение	Подготовка и передача программы	Установка и настройка; опытная эксплуатация; обучение пользователей.
--------------	---------------------------------	--

## 5.2. Расчет трудоемкости

Вероятные трудозатраты на выполнение данного проекта определяются с помощью экспертных оценок. Для этой цели опрошено пять экспертов-разработчиков, которые дали следующую оценку трудовых затрат на данную разработку. Результаты приведены в таблице 5.2.

Таблица 5.2 — Результат экспертных оценок

Эксперт 1	700 час
Эксперт 2	810 час
Эксперт 3	690 час
Эксперт 4	720 час
Эксперт 5	750 час

Ожидаемая продолжительность вычисляется по формуле (5.1).

$$t_{\text{ож}} = \frac{3 \cdot T_{\min} + 2 \cdot T_{\max}}{5} = \frac{3 \cdot 690 + 2 \cdot 810}{5} = 738 \text{ часов} \quad (5.1)$$

Затраты труда на разработку ПО выражаются формулой (5.2).

$$Q_{\text{prog}} = t_1 + t_2 + t_3, \quad (5.2)$$

где  $t_1$  — время на разработку алгоритмов и проектирование системы;

$t_2$  — время на написание программы;

$t_3$  — время на проведение тестирования, исправлений и написание документации;

Коэффициент затрат на алгоритмизацию и проектирование определяется равным  $n_1 = 0.7$ , так как данный этап наиболее важный. Затраты труда на проведения тестирования, внесения исправлений и подготовки сопроводительной документации определяются суммой затрат на выполнение каждой работы этапа тестирования по формуле (5.3).

$$t_3 = t_T + t_{И} + t_{Д}, \quad (5.3)$$

где  $t_T$  — затраты труда на проведение тестирования;

$t_{И}$  — затраты труда на внесение исправлений;

$t_{Д}$  — затраты труда на написание документации;

Значение  $t_3$  определяется посредством ввода соответствующих коэффициентов к значениям затрат труда на непосредственно программирование (формула 5.4).

$$t_3 = t_2 \cdot n_i \quad (5.4)$$

Коэффициент затрат на проведение тестирования отражает отношение затрат труда на тестирование программы по отношению к затратам труда на ее разработку. Так как отсутствуют высокие требования к надежности разрабатываемого ПО, коэффициент  $n_T$  принимается равным 0,29.

Коэффициент затрат на исправление программы  $n_{И}$  принимается равным 0,3, как наиболее используемый. Коэффициент затрат на написание документации  $n_{Д}$ , в связи с простотой пользования разрабатываемой программой, принимается равным 0,21.

Затраты на выполнение этапа тестирования выражаются формулой (5.5).

$$t_3 = t_2(n_T + n_{И} + n_{Д}) = t_2(0,29 + 0,3 + 0,21) \quad (5.5)$$

Связь  $Q_{prog}$  и  $t_2$  выражается формулой (5.6).

$$Q_{prog} = t_2(0,7 + 0,8 + 1) = 2,5t_2 \quad (5.6)$$

Затраты труда на этапы разработки приведены в таблице 5.3.



Таблица 5.3 — Затраты труда

Этап	Формула	Затраты
$t_1$	$t_1 = 0,7 \cdot t_2$	207 часов
$t_2$	$t_2 = Q_{prog}/2,5$	295 часов
$t_T$	$t_T = 0,29 \cdot t_2$	77 часов
$t_{II}$	$t_{II} = 0,3 \cdot t_2$	89 часов
$t_D$	$t_D = 0,21 \cdot t_2$	62 часа

Заказчик установил, что опытный срок эксплуатации для программного комплекса 10 чел/дней. При 8-ми часовом рабочем дне для одного программиста затраты на внедрение будут составлять 80 часов.

Общее значение трудозатрат определяется по формуле (5.7).

$$Q_p = t_1 + t_2 + t_3 + t_4 = 207 + 295 + (77 + 89 + 62) + 80 = 810 \quad (5.7)$$

### 5.3. Определение численности исполнителей

Средняя численность исполнителей определяется по формуле (5.8):

$$N = \frac{Q_p}{F}, \quad (5.8)$$

где  $F$  — фонд рабочего времени, определяющийся по формуле (5.9).

$$F = \sum_{i=1}^T F_{M_i} \quad (5.9)$$

$F_{M_i}$  — это фонд времени в текущем месяце и вычисляется для каждого месяца с учетом выходных и праздничных дней. На реализацию проекта отведено 3 месяца рабочего времени при односменной работе с продолжительностью рабочего дня 8 часов. Значения  $F_{M_i}$  для каждого месяца приведены в таблице 5.4.

Согласно формуле 5.9,  $F = 168 + 176 + 144 = 488$  часов.

Средняя численность исполнителей вычисляется по формуле (5.10):

Таблица 5.4 — Месячный фонд времени

Месяц	Фонд времени
Март	$(31 - 7 - 3) \cdot 8 = 168$ часов
Апрель	$(30 - 8) \cdot 8 = 176$ часов
Май	$(31 - 7 - 6) \cdot 8 = 144$ часов

$$N = \frac{810}{488} = 1,66 \quad (5.10)$$

Следовательно, необходима команда разработчиков в количестве не менее двух человек. В выполнении проекта будут заняты математик и два программиста. Математик необходим для составления математической модели и алгоритмов решаемой задачи, однако он не может участвовать непосредственно в процессе кодирования. По этой причине необходимы два программиста для обеспечения надлежащего контроля качества программного кода и параллельной разработки различных компонентов программы.

#### 5.4. Сетевая модель проекта

Для определения временных затрат и трудоемкости разработки программного обеспечения систем используется метод сетевого планирования. Метод сетевого планирования позволяет установить единой схемой связь между всеми работами в виде наглядного и удобного для восприятия изображения (сетевого графика), представляющего собой информационно-динамическую модель, позволяющую определить продолжительность и трудоёмкость, как отдельных этапов, так и всего комплекса работ в целом.

Составление сетевой модели включает в себя оценку степени детализации комплекса работ и определения логической связи между отдельными работами.

С этой целью составляется перечень всех основных работ, приведенный на рис. 5.1. В перечне указываются кодовые номера работ,

наименования, длительности и номера их предшественников с типом связи (окончание-начало, начало-начало, окончание-окончание).

	Режим задачи	Название задачи	Длительность	Начало	Окончание	Предшественн	Названия ресурсов
0		Метод уточнения и построения маршрута по набору GPS-координат	561 ч	Пн 02.03.15	Пн 08.06.15		
1		Начало проекта	0 ч	Пн 02.03.15	Пн 02.03.15		
2		Анализ и проектирование	115 ч	Пн 02.03.15	Пт 20.03.15		
3		Анализ предметной области	35 ч	Пн 02.03.15	Пт 06.03.15		Математик; Программист 1; Прог
4		Составление математической модели задачи	40 ч	Пн 09.03.15	Пт 13.03.15	3	Математик
5		Разработка алгоритма	35 ч	Пн 16.03.15	Пт 20.03.15	4	Математик; Программист 1
6		Разработка архитектуры ПО	45 ч	Пн 09.03.15	Пн 16.03.15	3	Программист 1; Программист 2
7		Проектирование интерфейса	35 ч	Вт 10.03.15	Пн 16.03.15	3	Программист 2
8		Кодирование	233 ч	Пт 20.03.15	Чт 30.04.15	2	
9		Реализация созданного алгоритма	55 ч	Пт 20.03.15	Вт 31.03.15	2	Математик; Программист 1
10		Реализация компонента загрузки данных	40 ч	Пн 30.03.15	Пн 06.04.15	2	Программист 2
11		Реализация компонента работы с картой	60 ч	Вт 31.03.15	Чт 09.04.15	2	Программист 1
12		Реализация пользовательского интерфейса	50 ч	Пт 20.03.15	Пн 30.03.15	7	Программист 2
13		Интеграция разработанных компонентов	118 ч	Чт 09.04.15	Чт 30.04.15	9;10;11;12	Программист 1; Программист 2
14		Тестирование и отладка	128 ч	Чт 30.04.15	Пт 22.05.15	8	
15		Сбор тестовых данных	6 ч	Чт 30.04.15	Пт 01.05.15	9	Математик; Программист 1; Прог
16		Тестирование на собранных данных	116 ч	Пт 01.05.15	Пт 22.05.15	8;15	Математик; Программист 1; Прог
17		Написание документации	54 ч	Чт 30.04.15	Пн 11.05.15	8	Математик; Программист 1; Прог
18		Внедрение	85 ч	Пт 22.05.15	Пн 08.06.15	14;17	
19		Установка и настройка	3 ч	Пт 22.05.15	Пт 22.05.15	14	Программист 1; Программист 2
20		Обучение пользователей	2 ч	Пт 22.05.15	Пн 25.05.15	19	Математик; Программист 1; Прог
21		Опытная эксплуатация	80 ч	Пн 25.05.15	Пн 08.06.15	20	Программист 1
22		Окончание проекта	0 ч	Пн 08.06.15	Пн 08.06.15	18	

Рисунок 5.1 — Сетевая модель проекта

Результат в днях:  $561 \text{ ч} / 8 = 71 \text{ день}$ .

## 5.5. Календарный график выполнения проекта

По сетевой модели может быть построена диаграмма Ганта, иллюстрирующая проведение последовательности работ, отражённых в сетевой модели (рис. 5.2). Красным выделен критический путь проекта длительностью в 69 дней. Дальнейшая оптимизация критического пути невозможна из-за малого количества разработчиков и жёстко установленной последовательности выполнения работ.

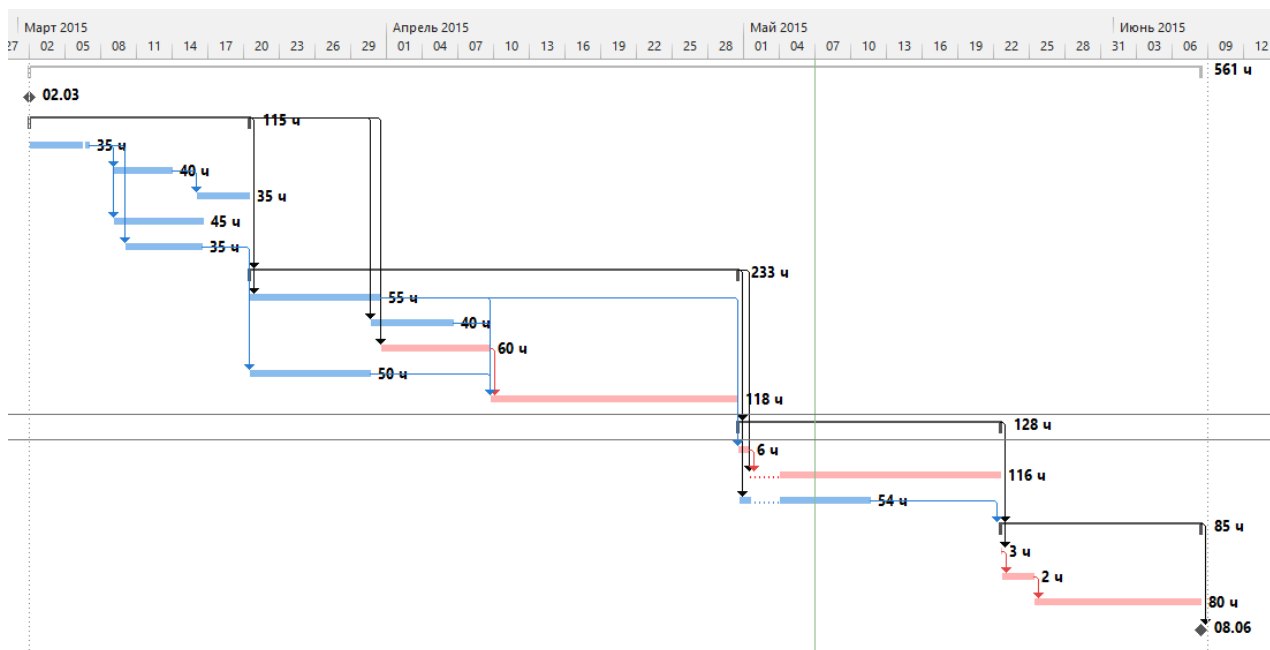


Рисунок 5.2 — Диаграмма Ганта с отслеживанием

Из рис. 5.2, 5.1 видно, что работы по созданию ПО завершаются 22.05.15 (окончание тестирования). Таким образом, непосредственно разработка ПО укладывается в отведенный срок 3 месяца. Однако для внедрения требуется дополнительное время, увеличивающее сроки завершения проекта. В таблице 5.5 отражена занятость каждого из работников в каждой задаче проекта.

Таблица 5.5 — Использование ресурсов

Работник / название задачи	Трудозатраты, ч	Затраты, р.
Программист 1	365	159687,50
Анализ предметной области	20	8750,00
Разработка алгоритма	8	3500,00
Разработка архитектуры ПО	45	19687,50
Реализация созданного алгоритма	50	21875,00
Реализация компонента работы с картой	60	26250,00

Интеграция разработанных компонентов	70	30625,00
Сбор тестовых данных	0	0,00
Тестирование на собранных данных	19	8312,50
Написание документации	8	3500,00
Установка и настройка	3	1312,00
Обучение пользователей	2	875,00
Опытная эксплуатация	80	35000,00
Программист 2	308	134750,00
Анализ предметной области	33	14437,50
Разработка архитектуры ПО	8	3500,00
Проектирование интерфейса	35	15312,00
Реализация компонента загрузки данных	40	17500,00
Реализация пользовательского интерфейса	50	21875,00
Интеграция разработанных компонентов	70	30625,00
Сбор тестовых данных	0	0,00
Тестирование на собранных данных	59	25812,50
Написание документации	8	3500,00
Установка и настройка	3	1312,50
Обучение пользователей	2	875,00
Математик	229	107343,75

Анализ предметной области	19	8906,25
Составление математической модели задачи	40	18750,00
Разработка алгоритма	35	16406,25
Реализация созданного алгоритма	11	5156,25
Сбор тестовых данных	6	2812,50
Тестирование на собранных данных	70	32812,50
Написание документации	46	21562,50
Обучение пользователей	2	937,50

## 5.6. Анализ структуры затрат проекта

Затраты на выполнение проекта состоят из затрат на заработную плату исполнителям, затрат на закупку или аренду оборудования, затрат на организацию рабочих мест и затрат на накладные расходы (формула 5.11).

$$K = C_{\text{ЗАРП}} + C_{\text{ОБ}} + C_{\text{ОРГ}} + C_{\text{НАКЛ}}, \quad (5.11)$$

где  $C_{\text{ЗАРП}}$  — заработная плата исполнителей;

$C_{\text{ОБ}}$  — затраты на обеспечение необходимым оборудованием;

$C_{\text{ОРГ}}$  — затраты на организацию рабочих мест;

$C_{\text{НАКЛ}}$  — накладные расходы.

### 5.6.1. Расчёт расходов

Затраты на выплату исполнителям заработной платы определяется соотношением (5.12):

$$C_{\text{ЗАРП}} = C_{\text{З.ОСН}} + C_{\text{З.ДОП}} + C_{\text{З.ОТЧ}} \quad (5.12)$$

где  $C_{3.осн}$  — основная заработная плата;

$C_{3.доп}$  — дополнительная заработная плата;

$C_{3.отч}$  — отчисление с заработной платы.

Расчет основной заработной платы при дневной оплате труда исполнителей следует проводить на основе данных по окладам и графику занятости исполнителей согласно формуле (5.13).

$$C_{зарп} = T_{зан} \cdot O_{дн}, \quad (5.13)$$

где  $T_{зан}$  — число дней, отработанных исполнителем проекта;

$O_{дн}$  — дневной оклад исполнителя.

При 8-и часовом рабочем дне  $O_{дн}$  рассчитывается по соотношению (5.14):

$$O_{дн} = \frac{O_{мес}}{F_M}, \quad (5.14)$$

где  $O_{мес}$  — месячный оклад;

$F_M$  — месячный фонд рабочего времени.

По данным кадровых агентств и Интернет-ресурса <https://rabota.yandex.ru>, средний месячный оклад специалистов требуемой квалификации следующий:

— оклад программиста — 76000 руб.

— специалист — 62000 руб.

Указанная зарплата не является «чистой» и из неё ещё будет вычтена сумма подоходного налога. Дополнительная заработная плата не предусматривается в связи с тем, что проект короткий и исполнители нанимаются на срок менее полугода.

MS Project позволяет автоматизировать расчёт затрат на ресурсы: как трудовые, так и материальные (табл. 5.7). Расходы на заработную плату составляют 401780,00р.

Отчисления на заработную плату в виде выплаты единого социального налога в 30% составляют 120534,00р.

### **5.6.2. Расчет расходов, связанных с обеспечением работ оборудованием**

Для работ, связанных с выполнением проекта необходимы три компьютера с процессором не менее производительным, чем Intel Core i5, минимум 8 Гб оперативной памяти и SSD. Возможна закупка трёх моноблоков Lenovo IdeaCentre B40-30 по 49000 рублей (согласно данным Яндекс.Маркет [13], это минимальная стоимость моноблока, удовлетворяющего приведенным требованиям) (табл. 4.6). Аренда в данной ситуации не является решением, так как выбор техники сильно ограничен. Кроме того, стоимость аренды в день слишком высока. Например, согласно сервису IT-аутсорсинга "Колизей Технологий", за трехмесячную аренду компьютера, не полностью удовлетворяющего указанным требованиям, необходимо заплатить 40410р + 13470р за доставку и установку на рабочем месте. Таким образом, покупка указанных выше моноблоков является более разумным решением.

### **5.6.3. Расчет расходов, связанных с организацией рабочих мест**

В соответствии с СанПин [14], расстояние между рабочими столами с видеомониторами должно быть не менее 2 м., а между боковыми поверхностями видеомониторов - не менее 1,2 м. Площадь на одно рабочее место с терминалом или ПК должна составлять не менее 6 кв.м., а объем - не менее 20 куб.м. Площадь, предусмотренная для размещения одного принтера, соответствует 0,5 площади рабочего места исполнителя. Расположение рабочих мест в подвальных помещениях не допускается. Помещения должны быть оборудованы системами отопления, кондиционирования воздуха или эффективной приточно-вытяжной вентиляцией. Дополнительно требуется удобное расположение офисного помещения. Таким требованиям соответствуют офисы класса С и выше. Для организации рабочих мест на три компьютера потребуется рабочая площадь 18 кв.м. Стоимость годов-



вой аренды помещений требуемого класса в различных районах г. Москвы [15] приведена в табл. 5.6:

Таблица 5.6 — Стоимость аренды помещений

Район	Площадь, м кв.	Стоимость аренды (за м кв. в год), р.
м. Речной вокзал	20	12000
м. Беговая	20	12700
м. Тульская	26	15000
м. Электrozаводская	20	11000
м. Семеновская	20	10000
м. Текстильщики	19	9000

Аренда самого дешевого офиса требуемого класса обходится в 9000 рублей за квадратный метр. Затраты на аренду помещения на этапе разработки вычисляются по формуле (5.15):

$$C_{\text{ОРГ}} = \frac{C_{\text{КВ.М}}}{12} \cdot S \cdot T_{\text{АР}}, \quad (5.15)$$

где  $C_{\text{КВ.М}}$  — стоимость аренды одного квадратного метра за год;

$S$  — арендуемая площадь офиса;

$T_{\text{АР}}$  — срок аренды.

В офисе нет необходимости на этапе внедрения. Учитывая, что тестирование заканчивается в конце мая (22.05.2015), офис необходимо арендовать на 3 месяца, тогда  $C_{\text{ОРГ}} = \frac{9000}{12} \cdot 19 \cdot 3 = 42750$  р.

#### 5.6.4. Расчёт накладных расходов

Накладные расходы вычисляют в расчете 60%-100% от расходов на основную заработную плату и в данном проекте планируются 60% (формула 5.16):

$$C_{\text{Р.НАКЛ.}} = C_{\text{Р.З.НАКЛ.}} \cdot 0.6 \quad (5.16)$$

Накладные расходы на разработку программы, рассчитанные по формуле (5.16), составляют:  $C_{p.накл} = 401780,00 \cdot 0.6 = 241068$  р.

Таблица 5.7 — Ресурсы проекта

Название ресурса	Тип	Стандартная ставка	Начисление	Затраты
Программист 1	Трудовой	70000,00 р./мес.	Пропорц.	159687,50 р.
Программист 2	Трудовой	70000,00 р./мес.	Пропорц.	134750,00 р.
Математик	Трудовой	75000,00 р./мес.	Пропорц.	107343,75 р.
Моноблок Lenovo IdeaCentre B40-30	Материальный	49000,00 р./шт.	В начале	147000,00 р.
Накладные расходы	Материальный	42750,00 р.	В начале	241068,00 р.
Отчисления на заработную плату	Материальный	241068,00 р.	В начале	120 534,00 р.
Офис	Материальный	120534,00 р.	В начале	42750,00 р.

Суммарные затраты составляют 953133,25р. и представлены в табл. 5.8.

Только на разработку без внедрения необходимо 912820,75р. Общие затраты на разработку проекта приведены в табл. 5.8.

Таблица 5.8 — Затраты проекта

Название задачи	Названия ресурсов	Затраты, р.
<b>Метод уточнения и построения маршрута по набору GPS-координат</b>		<b>953133,25</b>
Начало проекта	Моноблок Lenovo IdeaCentre B40-30[3 шт]; Накладные расходы[1]; Отчисления на заработную плату[1]; Офис[1]	551352,00

<b>Анализ и проектирование</b>		109250,00
Анализ предметной области	Математик;Программист 1;Программист 2	32093,75
Составление математической модели задачи	Математик	18 750,00
Разработка алгоритма	Математик;Программист 1	19906,25
Разработка архитектуры ПО	Программист 1;Программист 2	23187,50
Проектирование интерфейса	Программист 2	15312,50
<b>Кодирование</b>		153906,25
Реализация созданного алгоритма	Математик;Программист 1	27031,25
Реализация компонента загрузки данных	Программист 2	17500,00
Реализация компонента работы с картой	Программист 1	26250,00
Реализация пользовательского интерфейса	Программист 2	21875,00
Интеграция разработанных компонентов	Программист 1;Программист 2	61250,00
<b>Тестирование и отладка</b>		69750,00
Сбор тестовых данных	Математик;Программист 1;Программист 2	2812,50
Тестирование на собранных данных	Математик;Программист 1;Программист 2	66937,50
<b>Написание документации</b>	Математик;Программист 1;Программист 2	28562,50
<b>Внедрение</b>		40312,50
Установка и настройка	Программист 1;Программист 2	2625,00
Обучение пользователей	Математик;Программист 1;Программист 2	2687,50
Опытная эксплуатация	Программист 1	35000,00

Структура затрат на разработку проиллюстрирована на рис. 5.3.

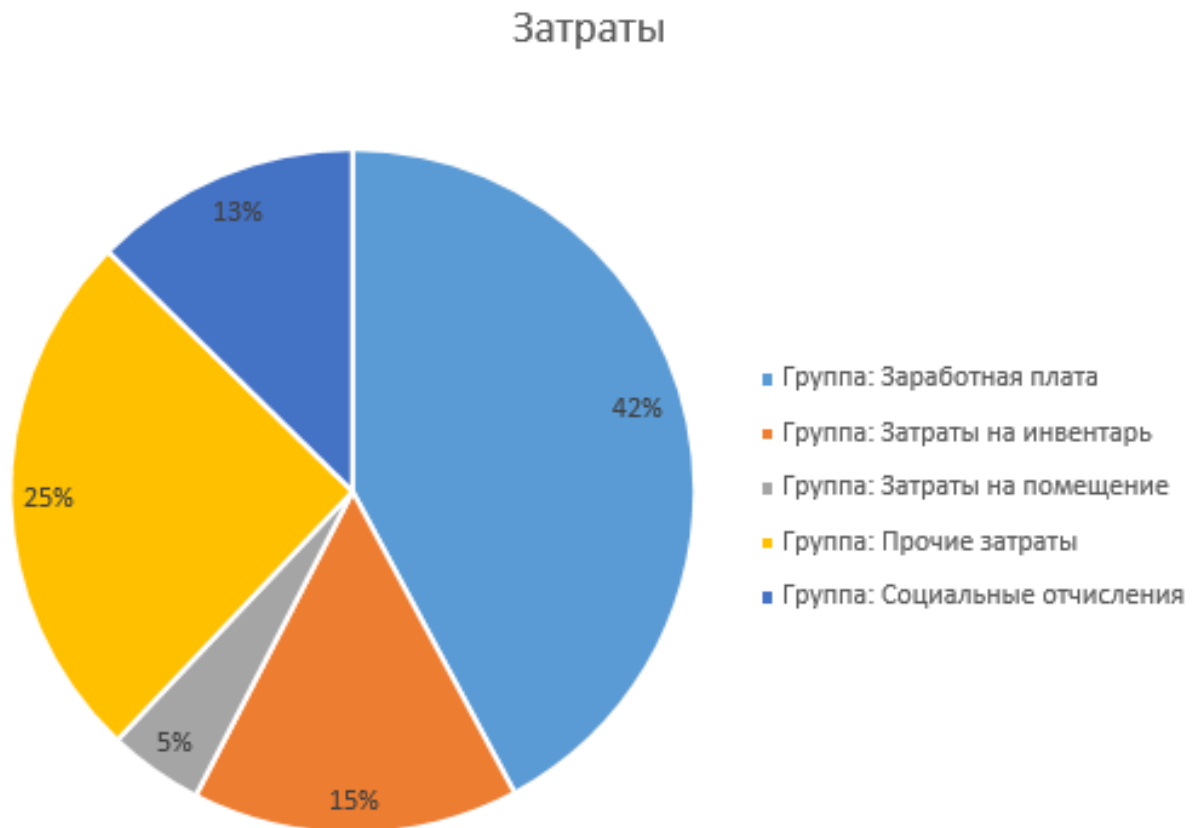


Рисунок 5.3 — Структура затрат на разработку

### 5.7. Исследование рынка для разрабатываемого ПО

В настоящее время сервисы, предоставляющие услуги GPS-трекинга не проводят фильтрацию данных от датчиков и не учитывают их погрешность. Разрабатываемый программный продукт направлен на решение конкретной указанной проблемы и предназначен для встраивания в существующую информационную систему для увеличения ее конкурентоспособности. Маловероятно, что данная программа заинтересует компании, предоставляющие услуги в данной сфере из-за отсутствия согласованности по используемым технологиям. Таким образом, число потенциальных покупателей на годовом интервале времени будет не более 1, т.е.  $N_P^O = 1$ .

## 5.8. Планирование цены и прогнозирование прибыли

На основе данных о затратах на разработку и внедрение, результатах прогнозирования объема продаж, определим стоимость одного комплекта программного обеспечения. Стоимость поставляемого на рынок ПО определяется частью стоимости разработки ПО, затрат на внедрение и прибыли фирмы-разработчика. Стоимость ПО можно рассчитать, используя соотношение (5.17):

$$K_{\text{ПО}} = (\Delta K + K_{\text{ВН}}) \cdot (1 + D_{\text{ПРИБ}}), \quad (5.17)$$

где  $\Delta K$  — часть стоимости разработки, приходящаяся на одну копию программы;

$K_{\text{ВН}}$  — стоимость внедрения программы;

$D_{\text{ПРИБ}}$  — процент прибыли, заложенный в стоимость.

Стоимость внедрения остается постоянной для каждой установки ПО, а частичная стоимость разработки, приходящаяся на каждый комплект ПО, определяется исходя из данных о планируемом объеме установок по формуле (5.18):

$$\Delta K = \frac{K}{N_P^O} \cdot (1 + H_{\text{СТ}}), \quad (5.18)$$

где  $K$  — стоимость проекта;

$N_P^O$  — планируемое число копий ПО;

$H_{\text{СТ}}$  — ставка банковского процента по долгосрочным кредитам.

Если, в качестве ставки процента по долгосрочным кредитам использовать 24,99% годовых и использовать рассчитанные значения затрат на разработку и планируемое число установок, то  $\Delta K = 912820,75/1 \cdot (1 + 0.2499) = 1140934,66\text{р.}$  В таком случае,  $K_{\text{ПО}} = (1140934,66 + 40312,50) \cdot (1 + 0.15) = 1358434,23\text{р.}$

Сумма 912820,75р., необходимая для разработки ПО, берётся в кредит в «ВТБ 24» под ставку 24,99% годовых на срок 6 месяцев. Основные финансовые параметры приведены в таблице 5.9.

Таблица 5.9 — Фрагмент таблицы общего баланса

Период расчёта	Баланс начальный	Сумма продаж	Сумма погашения кредита	Факт. совокупная прибыль	Факт. чистая прибыль	Баланс конечный
3–6.2015	-912820,75	1358434,23	1140934,66	217500,23	178350,19	406464,10

## 5.9. Выводы

В результате выполнения организационно-экономических расчетов установлены следующие показатели данного проекта:

— Общие затраты труда на реализацию проекта, которые составили 810 часов.

— Исходя из временных требований к реализации проекта, численность исполнителей составила 3 человека: математик и 2 программиста.

— Составлена сетевая модель с контролем затрат на каждом этапе и построена диаграмма Ганта — временной график выполнения проекта.

— Рассчитаны денежные затраты на реализацию проекта, которые составили в общей сумме 953133,25р. Источник денежных средств – кредит в «ВТБ 24» на сумму 912820,75р, под 24,99% годовых на срок 6 месяцев.

— Стоимость продукта составила 1358434,23р при условии продажи одного экземпляра в год.

— Ориентировочный срок окупаемости продукта – 4 месяца с момента начала разработки.

На основании вышеизложенного можно сделать вывод об экономической целесообразности проведения работ и внедрения в производство данной разработки.

## **6. Промышленная экология и безопасность**

### **6.1. Введение**

Безопасность жизнедеятельности (БЖД) — наука о комфортном и травмобезопасном взаимодействии человека со средой обитания.

Опасность — это явления, процессы или объекты, которые способны в определенных условиях наносить вред здоровью человека непосредственно или косвенно.

Для решения задач БЖД в рабочих условиях предназначены санитарные правила и нормы. Санитарные правила и нормы (СанПиН) — нормативные акты, устанавливающие критерии безопасности и безвредности для человека факторов окружающей его среды и требования к обеспечению благоприятных условий его жизнедеятельности. Согласно Закону РФ о санитарно-эпидемиологическом благополучии населения (1991), санитарные правила обязательны для соблюдения всеми учреждениями, предприятиями и общественными организациями, независимо от их ведомственного подчинения и форм собственности, а также всеми гражданами. Санитарным нарушением считается посягающее на права граждан и интересы общества умышленное или неосторожное действие (или бездействие), связанное с несоблюдением санитарных правил. Разработка программного обеспечения требует длительного взаимодействия с вычислительными системами.

Нормы для работы с ПЭВМ определены в СанПиН 2.2.2/2.4.1340-03 [14].

### **6.2. Требования к рабочему месту**

#### **6.2.1. Требования к ПЭВМ**

ПЭВМ должны соответствовать требованиям действующих санитарных правил и каждый их тип подлежит санитарно-эпидемиологической экспертизе с оценкой в испытательных лабораториях, аккредитованных в установленном порядке. Перечень кон-

тролируемых параметров вредных и опасных факторов представлен в таблице 6.1.

Таблица 6.1 — Контролируемые гигиенические параметры

№	Вид продукции	Код ОКП	Контролируемые гигиенические параметры
1	Машины вычислительные электронные цифровые, машины вычислительные электронные цифровые персональные (включая портативные ЭВМ)	40 1300, 40 1350, 40 1370	Уровни электромагнитных полей (ЭМП), акустического шума, концентрация вредных веществ в воздухе, визуальные показатели ВДТ, мягкое рентгеновское излучение*
2	Устройства периферийные: принтеры, сканеры, модемы, сетевые устройства, блоки бесперебойного питания и т.д.	40 3000	Уровни ЭМП, акустического шума, концентрация вредных веществ в воздухе



3	Устройства отображения информации (видеодисплейные терминалы)	40 3200	Уровни ЭМП, визуальные показатели, концентрация вредных веществ в воздухе, мягкое рентгеновское излучение*
4	Автоматы игровые с использованием ПЭВМ	96 8575	Уровни ЭМП, акустического шума, концентрация вредных веществ в воздухе, визуальные показатели ВДТ, мягкое рентгеновское излучение*
* Контроль мягкого рентгеновского излучения осуществляется только для видеодисплейных терминалов с использованием электронно-лучевых трубок.			

Допустимые уровни звукового давления и уровней звука, создаваемого ПЭВМ, не должны превышать значений, представленных в таблице 6.2.

Таблица 6.2 — Допустимые значения уровней звукового давления в октавных полосах частот и уровня звука, создаваемого ПЭВМ

Уровни звукового давления в октавных полосах со среднегеометрическими частотами									Уровни звука в дБА
31,5 Гц	63 Гц	125 Гц	250 Гц	500 Гц	1000 Гц	2000 Гц	4000 Гц	8000 Гц	
86 дБ	71 дБ	61 дБ	54 дБ	49 дБ	45 дБ	42 дБ	40 дБ	38 дБ	50

Допустимые визуальные параметры устройств отображения информации представлены в таблице 6.3

Таблица 6.3 — Допустимые визуальные параметры устройств отображения информации

№	Параметры	Допустимые значения
1	Яркость белого поля	Не менее 35 кд/м <sup>2</sup>
2	Неравномерность яркости рабочего поля	Не более ±20%
3	Контрастность (для монохромного режима)	Не менее 3:1
4	Временная нестабильность изображения (непреднамеренное изменение во времени яркости изображения на экране дисплея)	Не должна фиксироваться
5	Пространственная нестабильность изображения (непреднамеренные изменения положения фрагментов изображения на экране)	Не более $2 \cdot 10^{-4}L$ , где L - проектное расстояние наблюдения, мм

Концентрации вредных веществ, выделяемых ПЭВМ в воздух помещений, не должны превышать предельно допустимых концентраций (ПДК), установленных для атмосферного воздуха. Конструкция ПЭВМ должна обеспечивать возможность поворота корпуса в горизонтальной и вертикальной плоскости с фиксацией в заданном положении для обеспечения фронтального наблюдения экрана ВДТ. Дизайн ПЭВМ должен предусматривать окраску корпуса в спокойные мягкие тона с диффузным рассеиванием света. Корпус ПЭВМ, клавиатура и другие блоки и устройства ПЭВМ должны иметь матовую поверхность с коэффициентом отражения 0,4 - 0,6 и не иметь

блестящих деталей, способных создавать блики. Конструкция ВДТ должна предусматривать регулирование яркости и контрастности.

### **6.2.2. Требования к помещению**

Помещения для эксплуатации ПЭВМ должны иметь естественное и искусственное освещение. Эксплуатация ПЭВМ в помещениях без естественного освещения допускается только при соответствующем обосновании и наличии положительного санитарно-эпидемиологического заключения, выданного в установленном порядке. На одно рабочее место с ПЭВМ должно отводиться не менее 6 м<sup>2</sup> для ВДТ на базе электронно-лучевой трубки и 4,5 м<sup>2</sup> для ВДТ на базе плоских дискретных экранов. Внутренняя отделка помещений должна быть выполнена с использованием диффузно-отражающих материалов со следующими коэффициентами отражения:

- 0,7-0,8 для потолка;
- 0,5-0,6 для стен;
- 0,3-0,5 для пола.

Помещение должно быть оборудовано защитным заземлением (занулением) в соответствии с техническими требованиями.

### **6.2.3. Требования к микроклимату**

В помещении должны обеспечиваться оптимальные условия для теплообмена тела человека с окружающей средой. Так как вычислительная техника является выделяет значительное количество тепла, помещение должно проветриваться и/или кондиционироваться. Оптимальные параметры микроклимата определяются (таблица 6.4) в СанПиН 2.2.4.548-96 [16] и ССБТ ГОСТ 12.1.005-88 [17].

Таблица 6.4 — Оптимальные параметры микроклимата

Период года	Категория работ по уровням энергозатрат, Вт	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	Ia (до 139)	22,0 - 24,0	21,0 - 25,0	60 - 40	0,1
	Iб (140-174)	21,0 - 23,0	20,0 - 24,0	60 - 40	0,1
Теплый	Ia (до 139)	23,0 - 25,0	22,0 - 26,0	60 - 40	0,1
	Iб (140-174)	22,0 - 24,0	21,0 - 25,0	60 - 40	0,1

В помещениях должна проводиться ежедневная влажная уборка для избегания запыления.

#### 6.2.4. Требования к содержанию аэроионов и вредных химических веществ в воздухе

Уровни положительных и отрицательных аэроионов в воздухе помещения должны соответствовать нормам, указанным в СанПиН 2.2.4.1294-03 [18] (таблица 6.5).

Таблица 6.5 — Уровни аэроионов

Нормируемые показатели	Концентрация аэроионов, $\rho$ (ион/см <sup>3</sup> )		Коэффициент униполярности У
	Положительной полярности	Отрицательной полярности	
Минимально допустимые	400	600	0,4
Максимально допустимые	50000	50000	1,0

Содержанию химических веществ в помещении не должно превышать предельно допустимых концентраций в соответствии с действующими гигиеническими нормативами [19]. Концентрации вредных веществ, выделяемых ПЭВМ в воздух помещений, не должны превышать предельно допустимых концентраций, установленных для атмосферного воздуха.

### 6.2.5. Требования к уровням шума

Уровни шума в помещении не должны превышать предельно допустимых значений, установленных ГОСТ 12.1.003.83 [17] (таблица 6.6).

Таблица 6.6 — Нормативные значения уровней шума

Рабочее место	Уровень звукового давления в дБ в октавных полосах со среднегеометрическими частотами, Гц								Эквивалентный уровень звука, дБА
	63	125	250	500	1000	2000	4000	8000	
Помещения КБ, расчетчиков, программистов вычислительных машин	71	61	54	49	45	42	40	38	50

Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ. Уровни звукового давления и уровни звука, создаваемые ПЭВМ не должны превышать нормы, указанные в таблице 6.2.

### 6.2.6. Требования к освещению

Рабочие столы в помещении следует размещать так, чтобы естественный свет падал на ВДТ преимущественно слева. Искусственное освещение в помещении должно представлять собой систему общего равномерного освещения. Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300 - 500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк. Следует ограничивать прямую блескость от источников освещения, при этом яркость светящихся поверхностей (окна, светильники и др.), находящихся в поле зрения, должна быть не более 200 кд/м<sup>2</sup>. Следует ограничивать отраженную блескость на рабочих поверхностях

(экран, стол, клавиатура и др.) за счет правильного выбора типов светильников и расположения рабочих мест по отношению к источникам естественного и искусственного освещения, при этом яркость бликов на экране ПЭВМ не должна превышать 40 кд/м<sup>2</sup> и яркость потолка не должна превышать 200 кд/м<sup>2</sup>. Показатель ослепленности для источников общего искусственного освещения должен быть не более 20. Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м<sup>2</sup>, защитный угол светильников должен быть не менее 40 градусов. Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов. Следует ограничивать неравномерность распределения яркости в поле зрения пользователя ПЭВМ, при этом соотношение яркости между рабочими поверхностями не должно превышать 3:1 - 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1. Для освещения помещений с ПЭВМ следует применять светильники с зеркальными параболическими решетками, укомплектованными электронными пуско-регулирующими аппаратами (ЭПРА). Допускается использование многоламповых светильников с электромагнитными пуско-регулирующими аппаратами (ЭПРА), состоящими из равного числа опережающих и отстающих ветвей. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенные. Применение светильников без рассеивателей и экранирующих решеток не допускается. Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору. Коэффициент запаса (Кз) для осветительных установок общего освеще-

щения должен приниматься равным 1,4. Коэффициент пульсации не должен превышать 5%.

#### **6.2.7. Требования к визуальным параметрам ВДТ**

Предельно допустимые значения визуальных параметров ВДТ, контролируемые на рабочих местах, представлены в таблице 6.3.

#### **6.2.8. Требования к организации и оборудованию рабочих мест**

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм. Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной - не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм. Конструкция рабочего стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400 - 550 мм и углам наклона вперед до 15 град, и назад до 5 град.;
- высоту опорной поверхности спинки 300  $\pm$  20 мм, ширину - не менее 380 мм и радиус кривизны горизонтальной плоскости - 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах  $\pm$  30 градусов;
- регулировку расстояния спинки от переднего края сиденья в пределах 260 - 400 мм;

— стационарные или съемные подлокотники длиной не менее 250 мм и шириной - 50 - 70 мм;

— регулировку подлокотников по высоте над сиденьем в пределах 230  $\pm$  30 мм и внутреннего расстояния между подлокотниками в пределах 350 - 500 мм.

Рабочее место пользователя ПЭВМ следует оборудовать подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20°. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм. Клавиатуру следует располагать на поверхности стола на расстоянии 100 - 300 мм от края, обращенного к пользователю или на специальной, регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

### **6.3. Расчёты**

#### **6.3.1. Схема помещения**

Помещение представляет собой прямоугольную комнату размером 6,5м на 4,5м. Высота помещения – 2,9 м. В помещении расположены три рабочих места. В соответствии с требованиями к рабочим местам пользователя ПЭВМ, под каждое рабочее место отводится 4,5 м<sup>2</sup>, рабочие места расположены так, чтобы естественное освещение поступало слева, как указано на рис. 6.1.



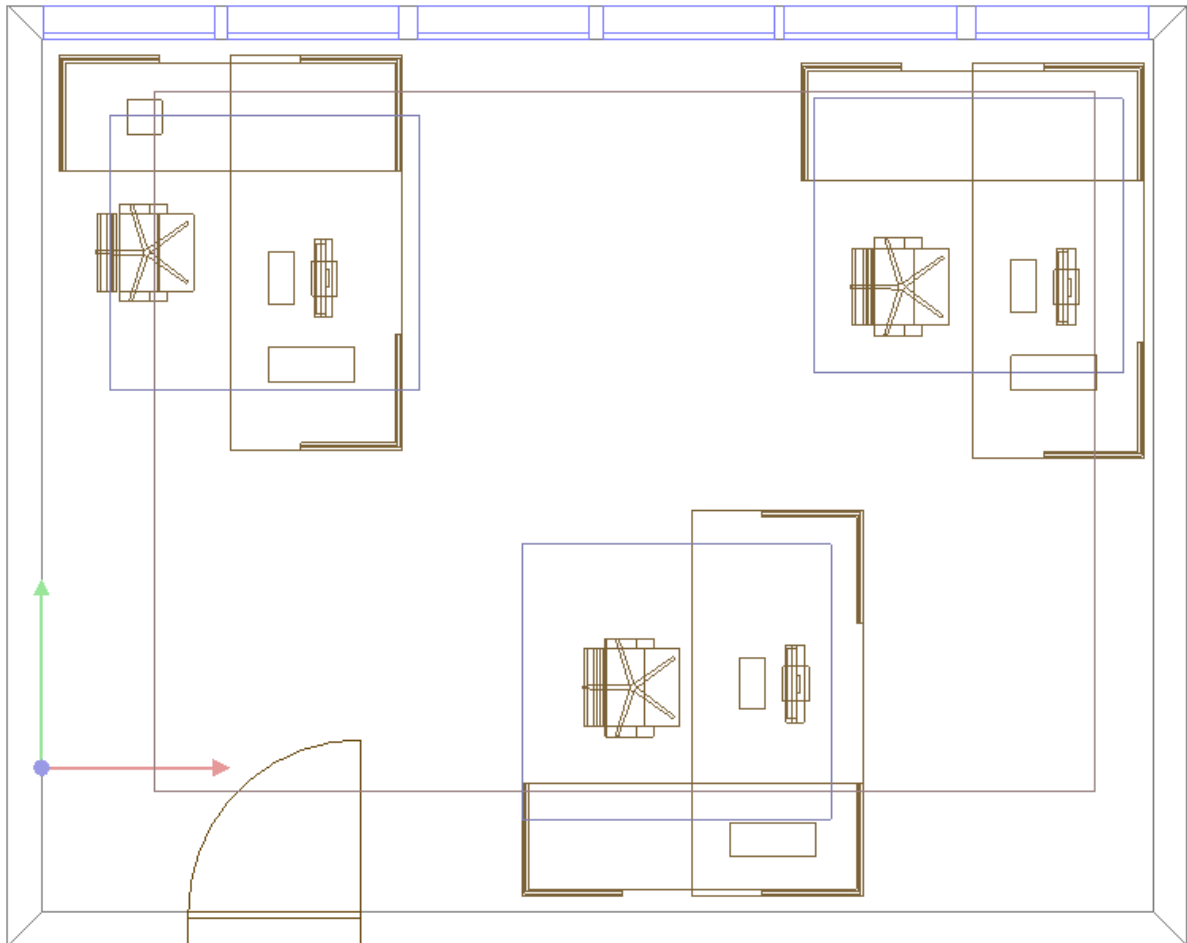


Рисунок 6.1 — Схема рабочего помещения

### 6.3.2. Расчёт освещённости помещения

В зависимости от цели расчета при проектировании искусственного освещения приходится решать следующий ряд вопросов:

- а) Выбрать или определить типы ламп и светильников.
- б) Выбрать систему освещения.
- в) Выбрать расположение светильников и определить их количество. Светильники, расположенные симметрично вдоль или поперек помещения, в шахматном порядке, рядами, ромбовидно, обеспечивают равномерное по площади освещение. Локализованное неравномерное размещение светильников производят с учетом местонахождения ПЭВМ, оборудования и т.д.
- г) Определить нормируемую освещенность рабочего места.

### 6.3.3. Типы ламп и светильников.

Согласно СанПиН [14], система светильники должны быть равномерно распределены по помещению. Целесообразно использовать люминесцентные лампы, поскольку они имеют ряд преимуществ перед лампами накаливания, среди которых повышенная светоотдача и КПД. Были выбраны лампы со следующими параметрами:

- Световой поток: 4896 lm
- Мощность светильников: 42.0 W

### 6.3.4. Выбор системы освещения

Наиболее экономичной является система комбинированного освещения, так как она создает наиболее равномерное светораспределение. При комбинированном освещении доля общего освещения в нем не должна быть меньше 10%.

### 6.3.5. Расположение светильников и их количество

Расчёт производится методом светового потока. Для определения количества светильников по формуле (6.1) определяется световой поток, падающий на поверхность.

$$F = \frac{E \cdot K \cdot S \cdot Z}{n}, \quad (6.1)$$

где  $F$  — рассчитываемый световой поток, Лм;

$E$  — нормированная минимальная освещенность, Лк. Для работы с ПЭВМ  $E = 350$  Лк;

$S$  — площадь освещаемого помещения ( $S = 23,04$  м<sup>2</sup>);

$Z$  — отношение средней освещенности к минимальной (обычно принимается равным 1,1-1,2, пусть  $Z = 1,1$ );

$K$  — коэффициент запаса, учитывающий уменьшение светового потока лампы в результате загрязнения светильников в процессе эксплуатации (его значение зависит от типа помещения и характера проводимых в нем работ, в случае работы с ПЭВМ  $K = 1,4$ );

$n$  — коэффициент использования, (выражается отношением свето-

вого потока, падающего на расчетную поверхность, к суммарному потоку всех ламп и исчисляется в долях единицы; зависит от характеристик светильника, размеров помещения, окраски стен и потолка, характеризующих коэффициентами отражения от стен (РС) и потолка (РП). В соответствии с СанПиН [14], эти коэффициенты равны РС = 50% и РП = 70%.

Значение  $n$  определяется по таблице коэффициентов использования светильников. Для определения  $n$  нужно вычислить индекс помещения по формуле (6.2):

$$I = \frac{S}{h \cdot (A + B)}, \quad (6.2)$$

где  $S$  — площадь помещения,  $S = 23,04 \text{ м}^2$ ;

$h$  — расчетная высота подвеса,  $h = 2,9 \text{ м}$ ;

$A$  — ширина помещения,  $A = 3,6 \text{ м}$ ;

$B$  — длина помещения,  $B = 6,4 \text{ м}$ .

Следовательно,  $I = 0,79$ , а  $n = 0,42$ . Тогда световой поток равен  $F = 29568 \text{ Лм}$ . Световой поток люминесцентных ламп, выбранных для помещения, равен  $F = 4896 \text{ Лк}$ . Необходимое количество ламп определяется по формуле  $N = F / F_{\text{л}}$ , где  $N$  — определяемое число ламп;  $F$  — световой поток,  $F = 29568 \text{ Лм}$ ;  $F_{\text{л}}$  — световой поток лампы,  $F_{\text{л}} = 4896 \text{ Лм}$ . Следовательно,  $N = 6$ .

В программе DIALux были выполнены построение схемы помещения и расчет освещенности. Расположение светильников в помещении показано на рисунке 6.2.

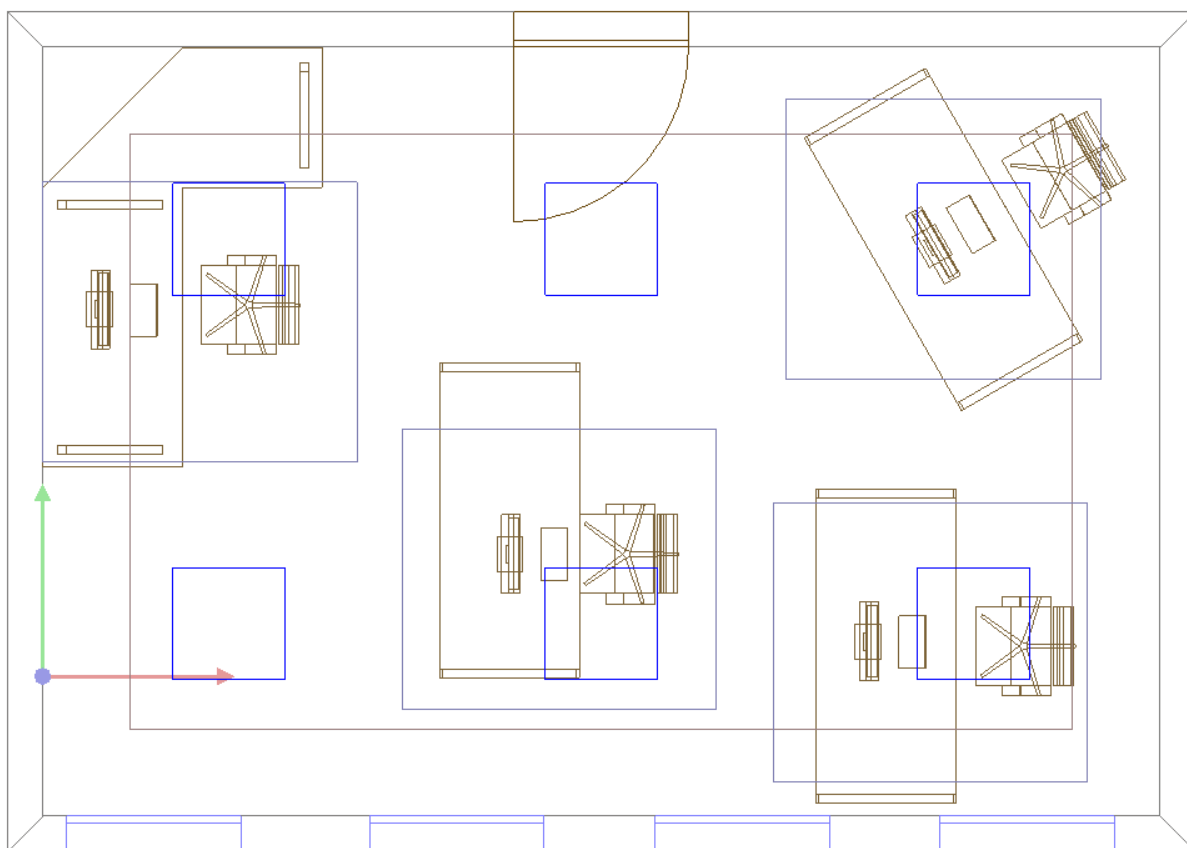


Рисунок 6.2 — Расположение светильников в помещении

Для помещения был произведён расчёт двух способов освещения: искусственного (рис. 6.3) и естественного (рис. 6.1).

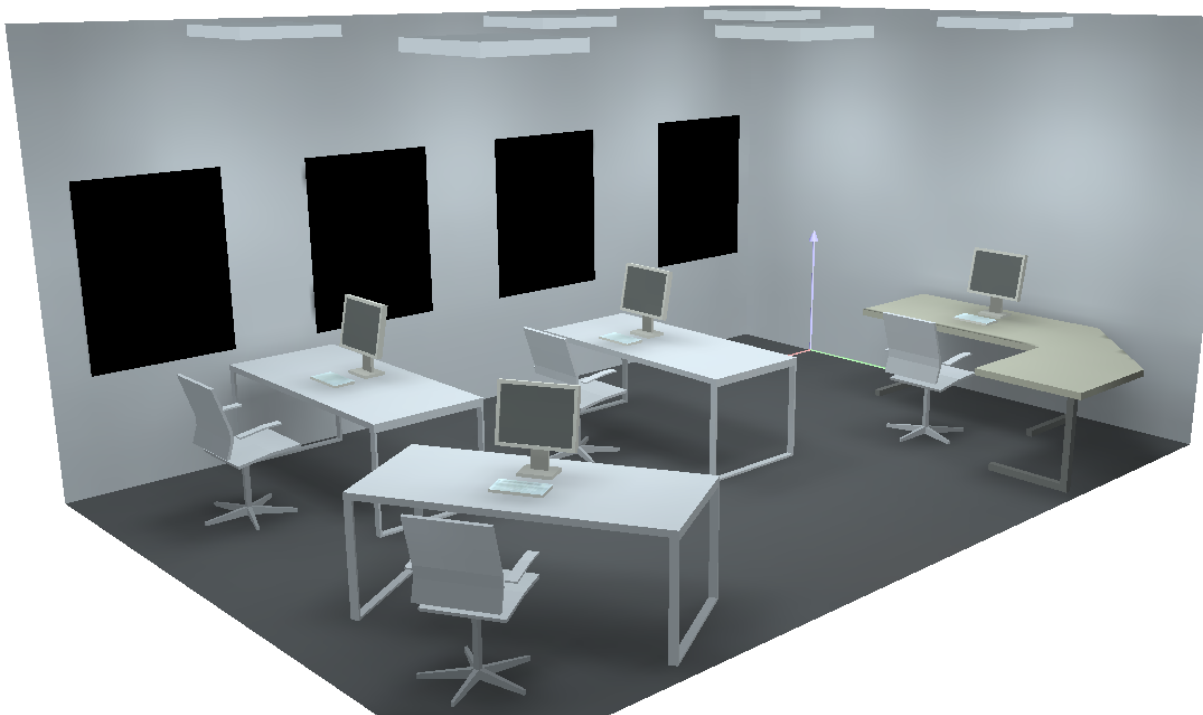


Рисунок 6.3 — Моделирование искусственного освещения в помещении

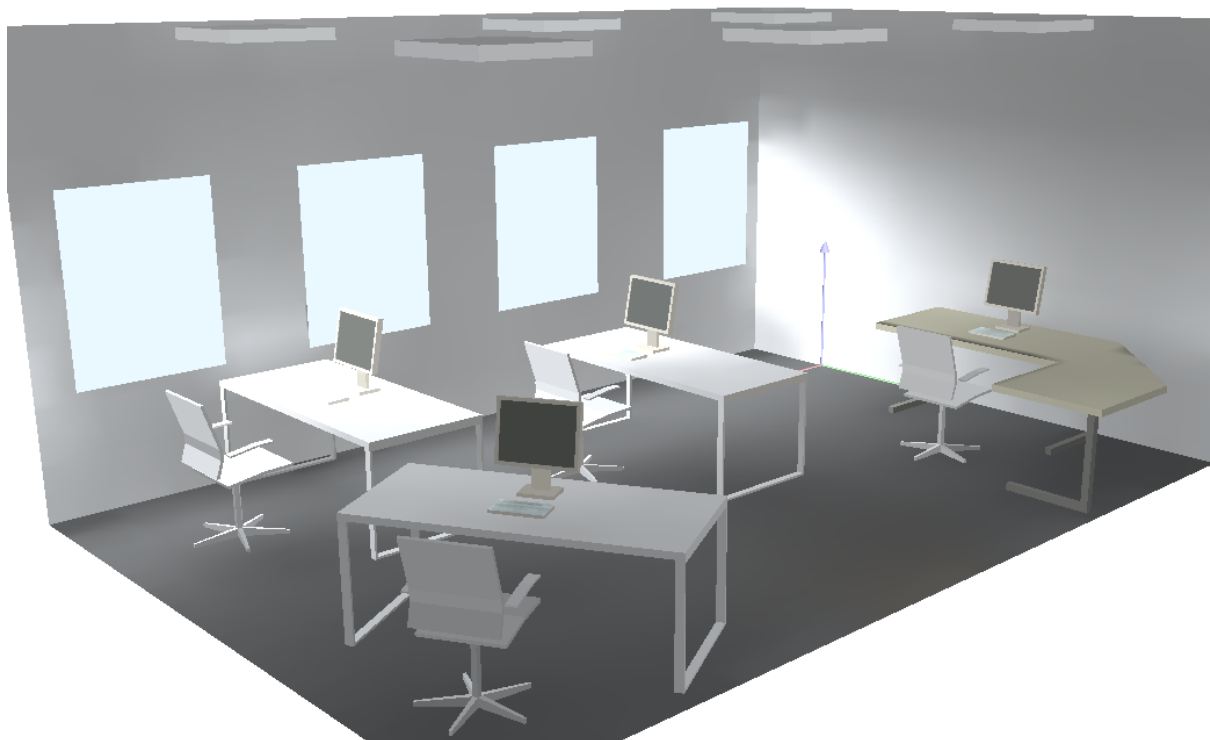


Рисунок 6.4 — Моделирование естественного освещения в помещении

Параметры светильников, использовавшихся для расчётов, приведены на рисунке 6.5.

3FFILIPPI 12508 03F LED 4x9W 640x640  
 № изделия: 12508  
 Световой поток (Светильник): 4896 lm  
 Световой поток (Лампы): 4896 lm  
 Мощность светильников: 42.0 W  
 Классификация светильников по CIE: 94  
 CIE Flux Code: 59 86 96 94 100  
 Комплектация: 4 x 9W 4xLED (Поправочный коэффициент 1.000).

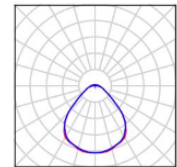
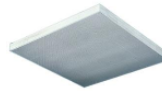


Рисунок 6.5 — Параметры светильников

Кривые освещенности для искусственного освещения приведены на рисунке 6.6.

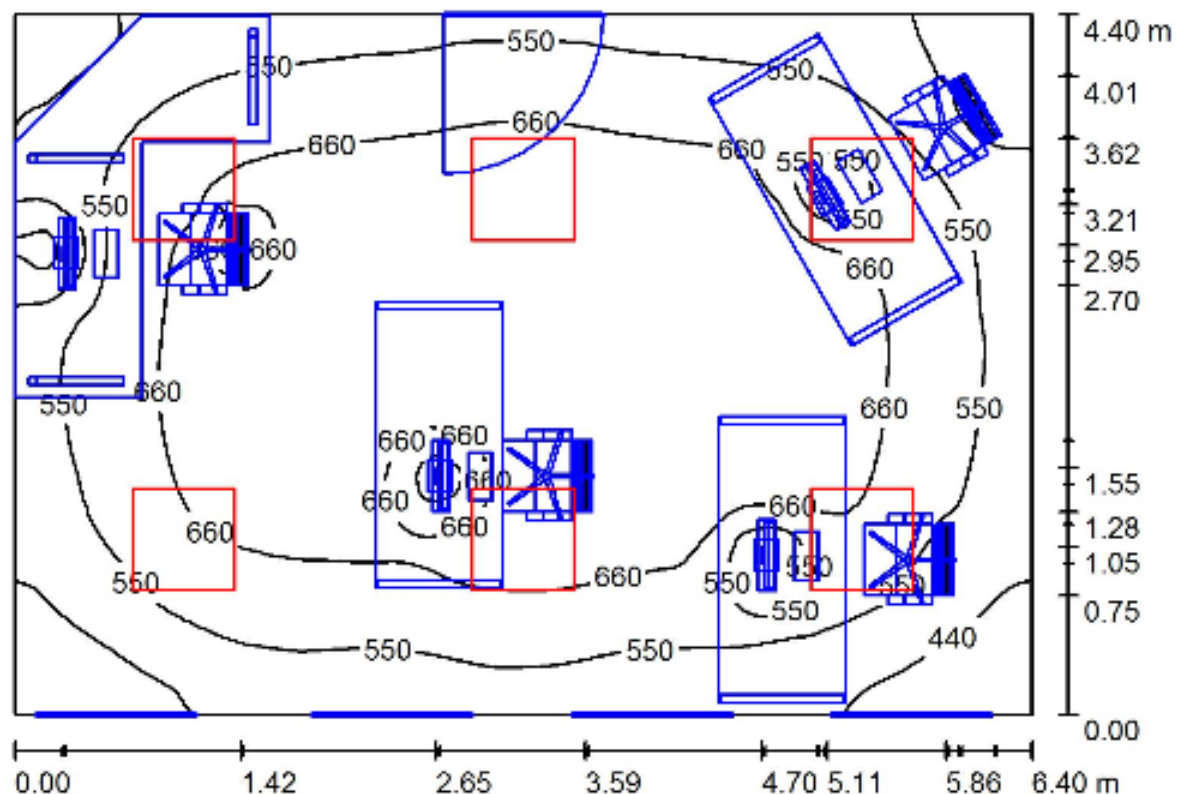


Рисунок 6.6 — Кривые освещенности

### 6.3.6. Расчёт уровней звукового давления

Исходные данные: источник шума находится у стену по центру на расстоянии 4 метра от расчётной точки. Уровни звуковой мощности источника шума приведены в таблице 6.7.

Таблица 6.7 — Уровни звуковой мощности источника шума

F	63	125	250	500	1000	2000	4000	8000
L	78,5630949	76,74032	78,62112	85,03365	87,00615	85,65765	81,2725	79,91561

На рисунке 6.7 приведена схема расположения источника шума.

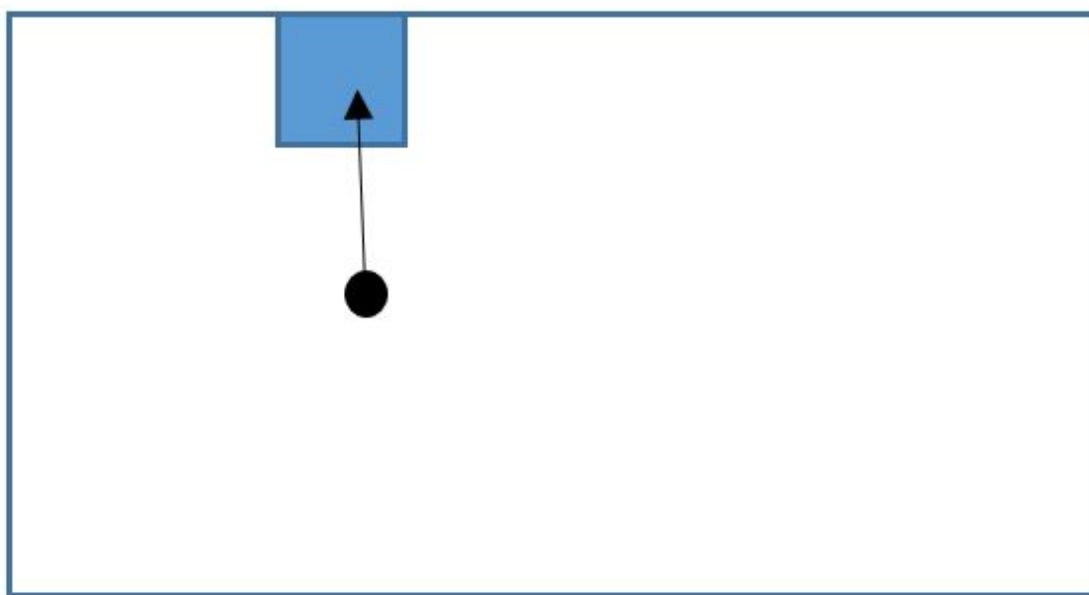


Рисунок 6.7 — Схема расположения источника шума

Расчёт проводился по следующим формулам:

$$L_p = L(f) + 10 \cdot \lg\left(\frac{\phi}{K \cdot S} + \frac{4}{B}\right), \quad (6.3)$$

$$B = B_{1000} \cdot \mu, \quad (6.4)$$

$$L_{\Sigma} = 10 \cdot \lg\left(10^{\frac{L_1}{10}}\right), \quad (6.5)$$

где  $\phi = 1, K = 1$ . Площадь  $S$  вычисляется по формуле  $S = \pi \cdot r^2$ , так как источник находится на полу у стены.

По СНиП II-12-77 был выбран следующий параметр  $B$ : с небольшим количеством людей (металлообрабатывающие цехи, вентиляционные камеры, генераторные, машинные залы, испытательные стенды) –  $V/20$ . Таким образом,  $B = 10 \cdot 10 \cdot 5/20 = 25$ .

Нормативные значения звукового давления выбираются в соответствии с СН [20].

В таблице 6.8 приведены результаты вычислений и нормативное значение. Норма – предельный уровень звукового давления. Соответствующий график изображен на рисунке 6.8.

Таблица 6.8 — Уровни звукового давления

F	63	125	250	500	1000	2000	4000	8000
L	78,5630949	76,74032	78,62112	85,03365	87,00615	85,65765	81,2725	79,91561
Норма	71	61	54	49	45	42	40	38

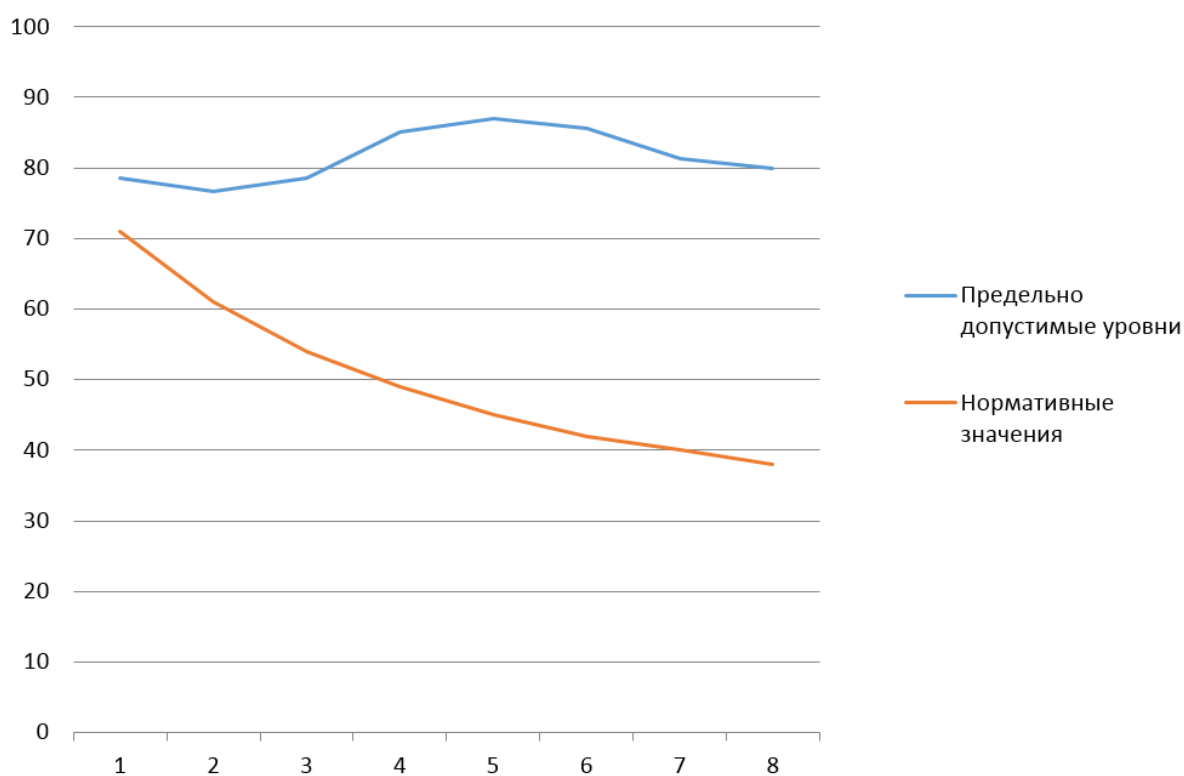


Рисунок 6.8 — Кривые освещенности

Уровень шума в помещении значительно превышает санитарные нормы во всём рассматриваемом частотном спектре. Отсюда следует необходимость защитных мероприятий. Следует использовать защитные экраны для ограждения источника шума.



## Заключение

В рамках выполнения квалификационной работы бакалавра:

- а) проведён анализ предметной области
- б) поставлены задачи, направленные на повышение точности маршрута
- в) предложены решения задач объединения маршрутов, определения остановок и удаления одиночных погрешностей, приводящих к возникновению пиков на маршруте
- г) выбран алгоритм для сглаживания погрешностей, составлена математическая модель исследуемой задачи для применения выбранного алгоритма
- д) на основе выбранных алгоритмов предложен метод для решения исследуемой задачи
- е) спроектировано и разработано ПО, реализующее предложенный метод
- ж) проведен эксперимент для подтверждения применимости метода к исследуемой задаче

В процессе выполнения квалификационной работы была написана и сдана в печать статья «Применение фильтра Калмана для обработки последовательности GPS-координат» (Молодёжный научно-технический вестник) под авторством Листеренко Р.Р. и Бекасова Д.Е..

Результаты проведенного эксперимента подтвердили применимость предложенного метода к решению поставленной задачи. Применение метода позволило повысить точность маршрута для всех рассмотренных случаев.

В процессе тестирования и эксплуатации разработанного ПО выявлены следующие недостатки:

- Удаляются не все избыточные точки, полученные в период неподвижности объекта

- Недостоверные сведения о точности датчика ухудшают качество работы метода

Кроме исправления вышеперечисленных недостатков, можно выделить следующие пути развития разработанной системы:

- Разработка способа автоматического определения параметров алгоритма
- Реализация режима реального времени работы метода

## Литература

1. *Yadav, J.* Error handling in GPS data processing / J. Yadav, R. Giri, L. Meena // *Mausam*. — 2011. — Vol. 62, no. 1. — Pp. 97–102.
2. *Xiong, N.* Multi-sensor management for information fusion: Issues and approaches / N. Xiong, P. Svensson // *Information Fusion*. — 2002. — Vol. 3, no. 2. — Pp. 163–186.
3. RunKeeper. URL: <http://runkeeper.com/> (Дата обращения: 29.05.2015).
4. *Sinha, P. K.* An improved median filter / P. K. Sinha, Q. H. Hong // *IEEE Transactions on Medical Imaging*. — 1990. — Vol. 9, no. 3. — Pp. 345–346.
5. *Chen, S.* A survey of smoothing techniques for ME models / S. Chen, R. Rosenfeld // *IEEE Transactions on Speech and Audio Processing*. — 2000. — Vol. 8, no. 1. — Pp. 37–50.
6. *Ramsay, J. O.* Kernel smoothing approaches to nonparametric item characteristic curve estimation / J. O. Ramsay // *Psychometrika*. — 1991. — Vol. 56, no. 4. — Pp. 611–630.
7. *Kalman, R. E.* A New Approach to Linear Filtering and Prediction Problems / R. E. Kalman // *Transactions of the ASME—Journal of Basic Engineering*. — 1960. — Vol. 82, no. Series D. — Pp. 35–45.
8. *Welch, G.* An Introduction to the Kalman Filter: Tech. Rep. TR-95-041 / G. Welch, G. Bishop: Department of Computer Science University of North Carolina at Chapel Hill, 2006. — 16 p.
9. *Hradilek, L.* Closed formulas for the direct and reverse geodetic problems / L. Hradilek // *Bulletin G od sique*. — 1976. — Vol. 50, no. 4. — Pp. 301–306.
10. *Vincenty, T.* Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations / T. Vincenty // *Survey Review*. — 1975. — apr. — Vol. 23, no. 176. — Pp. 88–93.
11. Design Patterns: Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides. — Addison-Wesley. — P. 395.

12. ГОСТ 19.102-77. Стадии разработки.
13. Яндекс.Маркет. URL: <https://market.yandex.ru/> (Дата обращения: 15.05.2015).
14. СанПиН 2.2.2/2.4.1340-03. Гигиенические требования к персональным электронно-вычислительным машинам и организации работы.
15. Из рук в руки. URL: <http://irr.ru> (Дата обращения: 15.05.2015).
16. СанПиН 2.2.4.548-96. Гигиенические требования к микроклимату производственных помещений.
17. ССБТ ГОСТ 12.1.005-88. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
18. СанПиН 2.2.4.1294-03. Гигиенические требования к аэроионному составу воздуха производственных и общественных помещений.
19. ГН 2.2.5.1313-03. Предельно допустимые концентрации вредных веществ в воздухе работы метода.
20. СН 2.2.4/2.1.8.562-96. Творческая деятельность, руководящая работа с повышенными требованиями, научная деятельность, конструирование и проектирование, программирование, преподавание и обучение, врачебная деятельность.