

Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)

На правах рукописи

Головков Александр Александрович

**Алгоритм фильтрации геоданных для мобильных
устройств в реальном времени**

Специальность 05.13.01 – Системный анализ, управление и обработка
информации (в технических системах)

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель
д.т.н., профессор Г.С. Иванова



Москва, 2018

Оглавление

	Стр.
Введение.....	5
Глава 1. Анализ существующих моделей и методов обработки геоданных, постановка задачи.....	12
1.1. Архитектура геоинформационных систем.....	12
1.2. Источники геоданных и их характеристики.....	15
1.2.1. Основные источники.....	16
1.2.2. Дополнительные источники.....	20
1.2.3. Сравнение характеристик основных и дополнительных источников	21
1.3. Существующие методы обработки геоданных.....	23
1.4. Формальная постановка задачи.....	30
Выводы по главе 1.....	32
Глава 2. Разработка метода обработки потока геолокационных данных.....	34
2.1. Модель входных и выходных данных.....	35
2.2. Метод фильтрации геоданных.....	40
2.2.1. Фильтр по точности.....	41
2.2.2. Фильтр сетевых координат.....	48
2.2.3. Фильтр по времени.....	50
2.2.4. Фильтр по скорости.....	53
2.2.5. Фильтр по ускорению.....	59
2.2.6. Фильтр стоянок.....	61
2.2.7. Сглаживающий фильтр.....	66
2.2.8. Угловой фильтр.....	68
2.2.9. Диагностический фильтр.....	72
Выводы по главе 2.....	77
Глава 3. Алгоритм обработки потока геоданных в реальном времени.....	84
3.1. Уточнение модели данных и метода обработки геокоординат.....	84
3.1.1. Получение геоданных и препроцессинг.....	84

3.1.2.	Фильтрация типов источников и спутников	86
3.1.3.	Фильтрация ложных координат	87
3.1.4.	Использование дополнительной информации из мобильного приложения	87
3.1.5.	Кеш координат и состояний	89
3.1.6.	Метод фильтрации геоданных в реальном времени	91
3.2.	Разработка алгоритмов препроцессинга и фильтрации	93
3.2.1.	Алгоритмы препроцессинга данных	94
3.2.2.	Алгоритм фильтрации по типам источников	97
3.2.3.	Алгоритм фильтрации по типам спутников	98
3.2.4.	Алгоритм фильтрации ложных координат	100
3.2.5.	Алгоритм фильтрации по точности	101
3.2.6.	Алгоритм фильтрации сетевых координат	102
3.2.7.	Алгоритм фильтрации по времени	104
3.2.8.	Алгоритм фильтрации по скорости	105
3.2.9.	Алгоритм фильтрации по ускорению	108
3.2.10.	Алгоритм фильтрации стоянок	109
3.2.11.	Алгоритм сглаживающей фильтрации	114
3.2.12.	Алгоритм угловой фильтрации	116
3.2.13.	Алгоритм диагностической фильтрации	119
3.2.14.	Алгоритм работы кеша координат и состояний	121
3.3.	Оценка быстродействия фильтрации	125
	Выводы по главе 3	130
Глава 4.	Разработка методов оценки эффективности фильтрации и экспериментальные исследования	133
4.1	Оценка эффективности алгоритма фильтрации потока геоданных	133
4.1.1	Оценка пройденного расстояния	136
4.1.2	Оценка плотности точек в треке	137
4.1.3	Оценка времени обработки	137
4.1.4	Оценка быстродействия фильтрации	138

4.1.5	Оценка точности фильтрации	139
4.1.6	Оценка ошибки фильтрации	146
4.2	Результаты экспериментов	154
	Выводы по главе 4	167
	Общие выводы и заключение	169
	Список сокращений и условных обозначений	171
	Список литературы	173

Введение

Актуальность темы диссертации. Значительный рост рынка мобильных устройств, производительности портативной аппаратной базы и качества программного обеспечения с начала 21 века привнес существенные изменения в бизнес-процессы различных компаний. В сферах логистики, финансов, FMCG, обслуживания и производства начали активно использоваться мобильные технологии для увеличения ключевых показателей эффективности: снижения себестоимости, повышения лояльности клиентов, сокращения времени планирования, увеличения производительности труда и др. Внедрение таких решений позволило расширить возможности сотрудников, как правило работающих вне офиса.

Одним из перспективных направлений является разработка комплексов мониторинга деятельности мобильных сотрудников, основанных на геоинформационных системах (ГИС). Такие комплексы наиболее востребованы в областях автоматизации транспортной логистики и технического обслуживания и ремонта оборудования (ТОиР). Их ключевые возможности заключаются в автоматическом формировании оптимальных маршрутов с учетом различных критериев, с учетом минимизации транспортных издержек и внеплановых простоев. Практическая реализация комплексов тесно связана со сбором, обработкой, хранением и визуализацией геолокационных данных, поступающих с мобильных устройств пользователей, что в целом является сложной технической задачей.

В рамках проектирования ГИС инженеры сталкиваются с необходимостью получения корректных геокоординат пользователей из набора необработанных данных от разных источников мобильных устройств. Учитывая необходимость снижения нагрузки на сервер, уменьшения количества передаваемых данных и получения геокоординат в реальном времени, в современных ГИС основная обработка геоданных обычно выполняется мобильными устройствами (см. Рисунок 1).

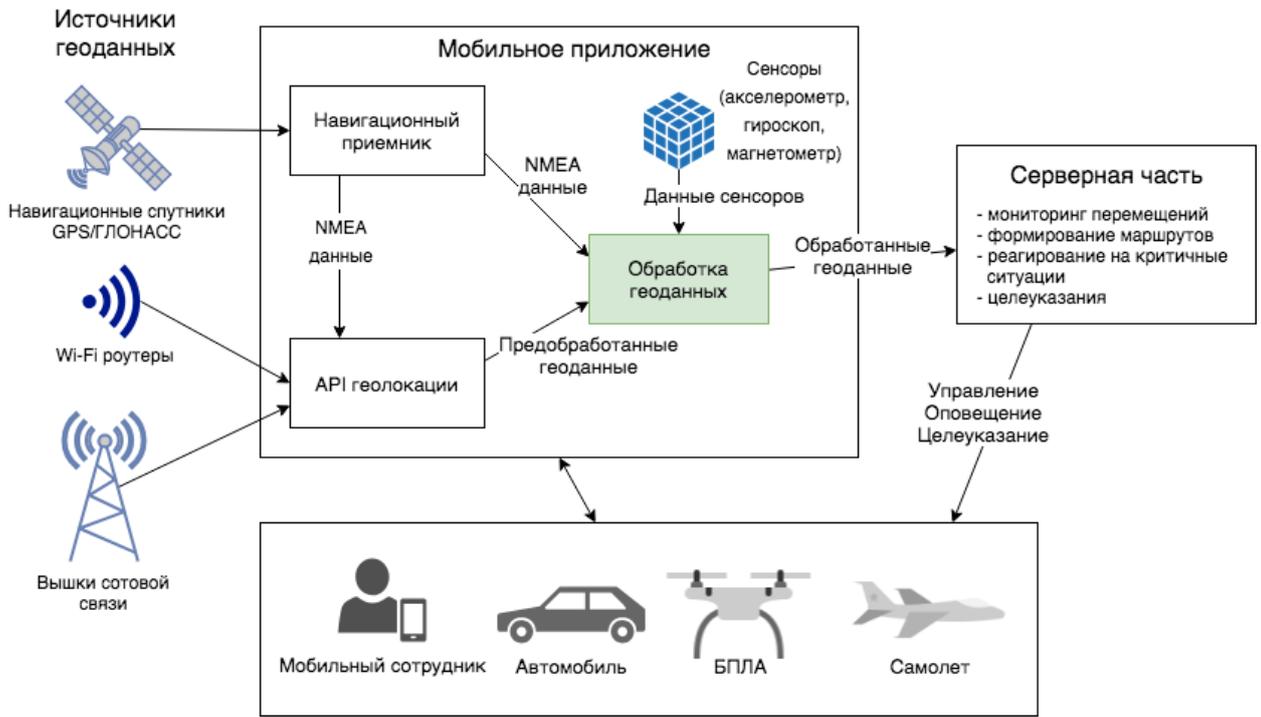


Рисунок 1. Архитектура типовой ГИС

Специфика геоинформационных технологий, алгоритмов, подходов, моделей и методов, а также особенности программного и аппаратного обеспечения определяют комплексный характер решения задачи обработки геоданных [47]. Практика использования стандартных геолокационных интерфейсов и существующих методов обработки геоданных в мобильных устройствах показала, что результат их работы не полностью удовлетворяют требованиям современных ГИС. К ГИС как правило предъявляются высокие требования по точности геопозиционирования для принятия решений по формированию маршрутов, оперативному реагированию на различные критичные ситуации. Однако в лучших существующих системах отклонение пройденного расстояния от реального составляет десятки километров, что недостаточно для обеспечения эффективности полевых работ в реальном времени.

В работе предлагается разработать алгоритм фильтрации для увеличения точности геопозиционирования за счет использования каскадной схемы обработки геоданных, учитывающей не только данные стандартных API (англ. Application Programming Interface) геолокации мобильных устройств, но и предобработанные NMEA (англ. National Marine Electronics Association) данные, а также

информацию со встроенных в устройство сенсоров. Реализация алгоритма обработки тесно связана со следующими особенностями.

Во-первых, выбор лучшего источника данных из доступных представляет собой неразрешимую задачу по причине невозможности предсказания активности источника и точности выдаваемых им координат в будущем. Следовательно, для увеличения полноты информации необходимо использовать все доступные источники, что ставит задачу нормализации и типизирования данных и учета внешних и внутренних факторов, влияющих на работу геопозиционирования.

Во-вторых, многие виды работ предполагают получение геоданных в режиме реального времени. Чтобы увеличить автономное время работы переносимых устройств, к алгоритмам обработки геоинформации предъявляются высокие требования по быстродействию, вычислительной и емкостной сложности. Реальное время также предполагает получение информации о местоположении настолько быстро, насколько это возможно. В связи с этим подсистема обработки геоданных должна обладать малым временем отклика.

Во-третьих, точность определения координат существенно зависит от окружения. В помещениях и вблизи высотных зданий по причине отражений, преломлений и затуханий радиосигналов многие источники работают с большой погрешностью. Ошибки геопозиционирования в городской местности, пригороде, лесной зоне и других областях также имеют свою специфику.

В-четвертых, определение стоянок сотрудников, координат с нулевой или близкой к нулю скоростью, представляет собой нетривиальную задачу из-за «блуждания» геокоординат в окрестности точек стоянок.

Обозначенные проблемы составляют далеко не все множество задач, с которыми сталкиваются исследователи и инженеры при проектировании и реализации геоинформационных систем. Проблемами разработки методов фильтрации геоданных занимались многие ученые. Среди них можно назвать Sahai A., Mann W., Enge P., Harle R., Сонькина Д.М., Павловского В.Е., Ворошилина Е.П., Алешечкина А.М., Прохорцова А.В., Пудловского В.Б., Садковского Б.П., Садковскую Н.Е., Салычева О.С., Романова А.Ю., Гаврилова А.В. и многих

других. Однако в основном они занимались частными случаями геопозиционирования, когда в расчет брался только один источник данных и не занимались вопросами разработки методов фильтрации геоданных на мобильных устройствах в реальном времени. В лучших алгоритмах фильтрации отклонение пройденного расстояния от реального составляет десятки километров (например, 27.9 ± 23.6 км, у Fused технологии геолокации в операционной системе Android). Многие существующие алгоритмы фильтрации основаны на привязке к картам дорог, однако в условиях отсутствия карт на каком-либо участке местности, навигации внутри помещений или движения пешком такой подход неприменим. В связи с этим в рамках перечисленных выше задач проблема фильтрации геоданных в мобильных устройствах в реальном времени и построения результирующего трека с высокой степенью соответствия реальному маршруту движения остается актуальной на настоящий момент.

Объектом исследования настоящей работы является система обработки информации для управления сравнительно медленно движущимися объектами (мобильными сотрудниками, автомобилями, беспилотными летательными аппаратами и др.) в реальном времени.

Предмет исследования – трек, строящийся геокоординатам с мобильных устройств, размещенных на движущихся объектах.

Целью диссертации является разработка алгоритма фильтрации геоданных для увеличения степени соответствия трека реальному маршруту движения.

Задачи исследований. Для достижения поставленной цели было необходимо решить следующие задачи:

- 1) анализ архитектур геоинформационных систем (ГИС), процесса обработки геоданных, стратегии разделения ответственности за обработку информации, источников геоинформации в мобильных устройствах;
- 2) анализ треков, строящихся в реальном времени по геокоординатам от различных источников, выявление причин и характера отклонений треков от реальных маршрутов движения;

- 3) анализ существующих моделей и методов улучшения качества треков, выявление их преимуществ и недостатков;
- 4) разработка модели описания входных и выходных данных на основе детального анализа геоинформации от разных источников;
- 5) разработка метода фильтрации потока геоданных, исследование различных типов «выбросов» геокоординат и получение расчетных соотношений для их исключения;
- 6) разработка алгоритма последовательной обработки потока геоданных, ориентированного на мобильные операционные системы, с целью исключения «выбросов», уменьшения плотности точек и сглаживания получаемого трека; разработки набора параметров и характеристик оценки эффективности и качества фильтрации, а также расчетных соотношений для их определения;
- 7) разработка подсистемы фильтрации на основе предложенного алгоритма, критериев оценки его эффективности и качества, экспериментальная проверка полученных результатов.

Методы исследования. В работе использованы методы декомпозиции, формализации, определения общей цели в рамках постановки задачи увеличения степени соответствия трека реальному маршруту движения, статистический и сравнительный анализ геоданных, в том числе регрессионный и кластерный анализ; методы анализа эффективности алгоритмов, оценки достоверности классификаторов и качества работы модели.

Научная новизна работы заключается в следующем:

- 1) разработана формальная постановка задачи увеличения степени соответствия трека реальному маршруту движения, что позволило систематизировать исходные данные и выбрать пути решения;
- 2) классифицированы случаи отклонений трека от реального маршрута движения для различных источников, определены признаки отклонений, что позволило сформулировать расчетные соотношения для фильтрации геокоординат;

- 3) построена модель входных и выходных геоданных, ориентированная на мобильные операционные системы, что позволило учитывать разнородные источники геоинформации, по мере возможности дополняя отсутствующие данные на основе имеющихся;
- 4) разработаны методы фильтрации потока геокоординат, обеспечивающие высокую степень точности и низкую плотность точек трека;
- 5) разработан алгоритм фильтрации, который позволяет получать треки в реальном времени с достоверностью 99.164 %, точностью 99.836 % и полнотой 97.704 % на мобильных устройствах;
- 6) разработан набор критериев оценки качества фильтрации и расчетные соотношения для их определения.

Практическая значимость. Разработанный алгоритм обработки положен в основу подсистемы фильтрации геоданных в реальном времени, ориентированной на мобильные устройства; достоверность результирующих треков была увеличена на 1.39 % в сравнении с существующими рассмотренными системами, что позволило уменьшить среднее отклонение пройденного расстояния от реального примерно в 30 раз.

Достоверность результатов работы подтверждается адекватностью используемых моделей и разработанных методов, корректностью математических преобразований и статистически достоверными результатами проведенных экспериментов.

Внедрение результатов работы. Результаты работы в виде моделей, методов, методик, алгоритмов, аналитических и программных средств были использованы в ряде разработок ГК CDC, в частности в коммерческом решении ОПТИМУМ ГИС, которое успешно эксплуатируется сотрудниками ГК СНС. За счет отслеживания координат сотрудников в реальном времени решение позволяет оперативно осуществлять мониторинг перемещений курьеров в сферах логистики. Подсистема фильтрации также используется в коммерческом продукте ОПТИМУМ ММС, ключевой особенностью которого является быстрое формирование маршрутов сервисных инженеров и ремонтных бригад, оперативная

координация их действий и реагирование на критические ситуации в областях технического обслуживания и ремонта различных технических объектов.

Апробация работы. Основные положения и результаты диссертационной работы представлялись и докладывались на:

1. III Международной научно-практической конференции «Научные исследования в области технических наук» (Саратов, 2018);
2. Международной научной конференции «Приоритетные направления развития науки, технологий и техники» (Рим, 2018);
3. Международном научном конгрессе «Наука и инженерное образование» SEE-2016 (Москва, 2016).

Публикации. Основные научные результаты диссертационной работы отражены в 6 статьях, 4 из которых – в научных журналах из перечня ВАК РФ.

Личный вклад. Все результаты и положения, составляющие основное содержание диссертации, разработаны и получены лично автором или при его непосредственном участии.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, и содержит 189 страниц текста, 74 рисунка и 13 таблиц.

Глава 1. Анализ существующих моделей и методов обработки геоданных, постановка задачи

1.1. Архитектура геоинформационных систем

С точки зрения работы с геоданными типовую геоинформационную систему (ГИС), предназначенную для мониторинга деятельности мобильных сотрудников в рамках комплексов автоматизации транспортной логистики и технического обслуживания и ремонта оборудования (ТОиР), можно представить в виде совокупности двух взаимодействующих частей: серверной и мобильной.

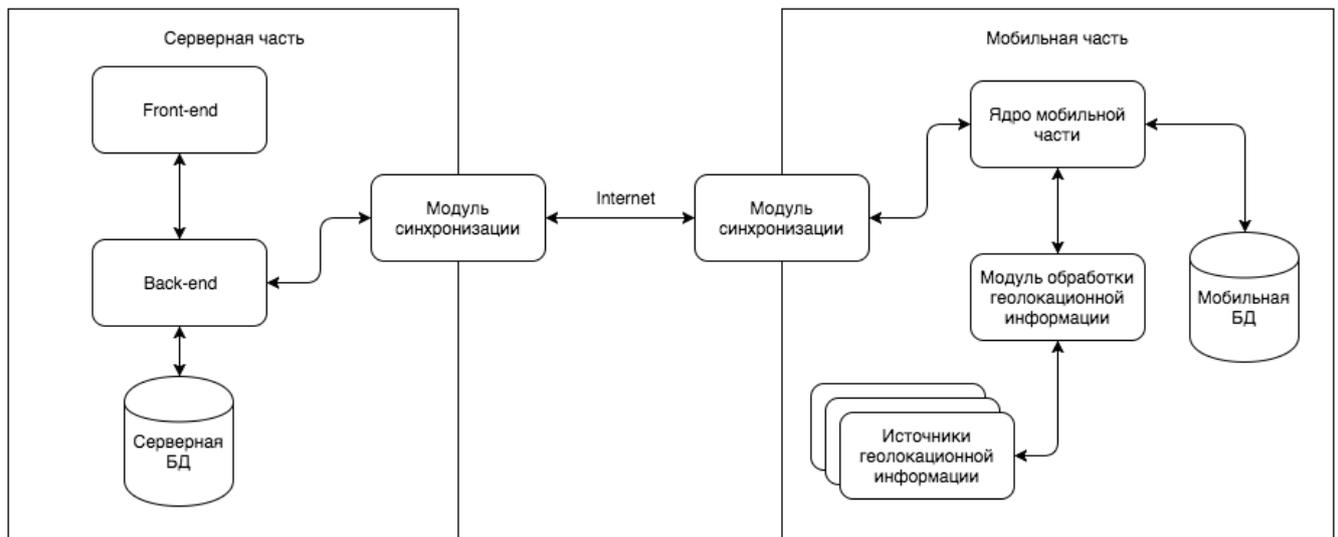


Рисунок 1.1. Архитектура типовой ГИС

На рисунке серверная часть представлена в виде одной централизованной системы. Однако ГИС могут также иметь распределенную архитектуру, когда существует несколько серверных частей, отвечающих за разные функции, и различных модулей, в том числе дублирующих друг друга для обеспечения отказоустойчивости, внутри каждой из частей. Однако общий принцип работы остается неизменным.

Количество мобильных частей системы равно количеству пользователей.

Помимо основных блоков ГИС включают дополнительные компоненты, обеспечивающие безопасность и целостность данных, интеграцию с другими корпоративными системами: модули шифрования, репликации, внешние и

внутренние API и др. Данные подсистемы практически не связаны с задачей сбора и обработки геоданных, поэтому в настоящей работе рассматриваться не будут.

Рассмотрим процесс обработки геолокационной информации в основных модулях системы. Данные из различных источников в мобильной части поступают в модуль обработки, где происходит фильтрация данных. Модуль может работать с различными источниками в зависимости от настроек системы. Ядро мобильной части сохраняет обработанные данные в мобильную базу данных и совместно с модулем синхронизации обеспечивает обмен с серверной частью. Со стороны сервера модуль Back-end как правило производит постобработку геоинформации, определяя значимые характеристики траектории движения сотрудника: отклонение от заданного маршрута, аномалии, простои и др. [1, 6, 8, 16, 17, 18, 27, 51, 52, 55, 70-72]. Вся результирующая информация попадает в серверную базу данных для хранения [57, 58, 62, 66-69]. Компонент Front-end обычно представляет собой «тонкий» или «толстый» клиент всей ГИС.

В архитектуре можно выделить два модуля, которые непосредственно обрабатывают геолокационную информацию: модуль обработки в мобильной части [75-78] и Back-end в серверной [73]. При проектировании ГИС возникает задача разграничения обязанностей данных модулей. Вычисления различных интегральных характеристик движения, определение аномалий и отклонений, расчет средних значений параметров местоположения обычно производится на серверной стороне. Эта информация в общем случае необходима только в этой части системы. Существует два основных направления дальнейшей стратегии по разделению функций обработки геоданных:

- приоритет серверной части: применение алгоритмов фильтрации, классификации и кластеризации для получения адекватных высокоточных геоданных непосредственно на сервере, при этом мобильные приложения передают всю необработанную или частично обработанную информацию;

- приоритет мобильной части: обработка информации на мобильных устройствах с последующей передачей в серверную часть только итоговых результатов.

Первый способ, менее сложный с точки зрения реализации, позволяет получить больше данных в одном месте системы, что в некоторых случаях существенно улучшает качество анализа интегральных характеристик трека и синтеза целевых показателей маршрутов пользователей [1]. Недостатком является высокая нагрузка на Back-end и каналы связи: сервер – мобильные приложения [2]. Отметим также возникновение существенной задержки (минуты и часы) между поступлением необработанной геоинформации от источников и получением результата на мобильной части из-за промежуточной обработки сервером.

Второй способ является предпочтительным: в этом случае существенно снижается нагрузка на серверную часть и уменьшается объем передаваемых данных, что в случае плохого соединения с сетью или полного отсутствия связи является ключевым фактором. Данный подход предполагает четкое разделение границ ответственности за обработку геоинформации между серверной и мобильной подсистемами. Мобильная часть должна выполнить корректный анализ необработанных данных минимизируя вычислительные затраты и длительность обработки, от которой напрямую зависит время работы мобильного устройства. И отправить результат на сервер, при этом не ограничивая его в возможностях постобработки информации для отображения различных характеристик трека во Front-end модуле. Такая стратегия позволяет реализовать потоковую обработку геоинформации в реальном времени на мобильном устройстве, что дает возможность получить высокоточные данные местоположения настолько быстро, насколько это возможно [63, 64]. Это несомненно важно для ГИС, которые предполагают немедленное реагирование на различные ситуации на основе данных местоположения. Реализация способа предполагает комплексный подход к проектированию потоковой системы обработки геоинформации.

1.2. Источники геоданных и их характеристики

В рамках решения специфических бизнес-задач исследователи и разработчики часто сталкиваются с необходимостью модификации алгоритмов обработки геоданных. Высокие требования к конечным мобильным приложениям обязывают использовать все доступные возможности для обработки геолокационной информации, комбинировать низкоуровневое и высокоуровневое API разнородных источников данных. Задача усложняется необходимостью типизации и сравнения данных. Также стоит отметить, что проприетарные мобильные ОС, в которых внутренняя реализация обработки полностью закрыта от разработчика, существенно усложняют освоение специфики предметной области.

Рассмотрим доступные источники геолокационных данных в современных мобильных устройствах [46]. Без ограничения общности далее будем ориентироваться на устройства под управлением ОС Android. Практически все приведенные источники, их характеристики и специфика работы аналогичны во всех мобильных операционных системах.

Разделим все источники данных на основные и дополнительные. Основные – те, которые непосредственно определяют местоположение устройства, выходными данными этих источников являются широта и долгота. Они всегда входят в состав аппаратно-программной части мобильных устройств, при этом достаточно наличия одного, чтобы осуществлять геолокацию. Дополнительные источники могут предоставить только косвенную информацию о геопозиции, каким-либо образом характеризую физическое положение устройства: пространственные сенсоры и др. К дополнительным также отнесем источники, работающие только в определенной среде или требующие отдельных внешних устройств. Далее в работе приоритет будет отдан основным источникам, как наиболее используемым, достоверным и работающим независимо от окружения.

1.2.1. Основные источники

Спутниковые навигационные приемники

Данные источника представляют собой геоинформацию, полученную с помощью геостационарных спутников GPS/GLONASS/Galileo/GNSS. В общем случае существует два способа получения данных: непосредственно через GPS-модуль в формате необработанных символьных сообщений NMEA различного типа [3] и с помощью API операционной системы, которая осуществляет предварительную фильтрацию.

Проанализируем основные проблемы, которые могут возникнуть при обработке информации от данного источника.

1. Выбросы координат при навигации внутри зданий

За счет переотражения, преломления и затухания радиосигналов внутри помещений, изменения количества и типа доступных спутников при перемещении от одной стороны здания к другой, необработанные данные представляют собой разбросанные геокоординаты около одной реальной точки местоположения пользователя в здании (см. Рисунок 1.2).

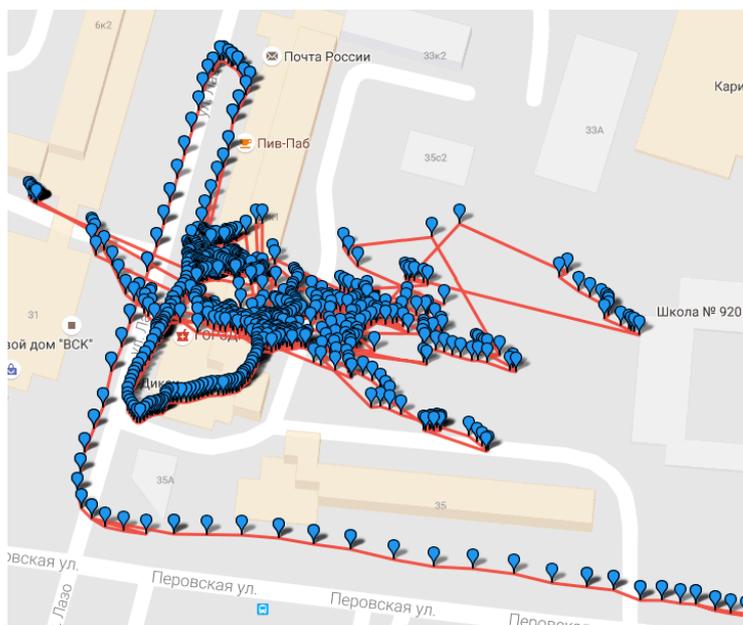


Рисунок 1.2. Пример выбросов координат при навигации внутри здания

Выбросы координат имеют непостоянный характер, характеристики точности таких координат могут быть довольно высокими, однако реальное

положение точек может не соответствовать полученной широте и долготе. Фильтрация таких точек представляет собой сложную задачу. Попытки решения проблемы рассмотрены в [1, 2, 7].

2. «Блуждание» координат при нулевой скорости

При нулевой или близкой к нулю скорости пользователя широта и долгота в NMEA сообщении может незначительно смещаться относительно реальной точки стоянки (см. Рисунок 1.3).



Рисунок 1.3. «Блуждание» координат при нулевой скорости

Такое поведение в общем случае связано с особенностями работы спутниковой навигации. Проблема похожа на рассмотренную в предыдущем пункте, отличительной особенностью является то, что здесь параметры погрешности рассчитываются довольно точно, что позволяет выявлять и обрабатывать такие случаи [1, 2, 65].

3. Отклонение трека вблизи высотных зданий

Отражение радиосигнала вблизи высоких объектов приводит к отклонению результирующего трека (см. Рисунок 1.4). Исходя из анализа NMEA сообщений точность таких координат по может быть высока, однако трек на карте может явно не совпадать с реальной траекторией движения пользователя. Для решения этой проблемы можно выполнить анализ общего направления движения и скорости и осуществить привязку к ближайшей дороге на карте. Однако в случае неполной карты или значительного отклонения координат этот метод принесет только бóльшую неточность в трек.

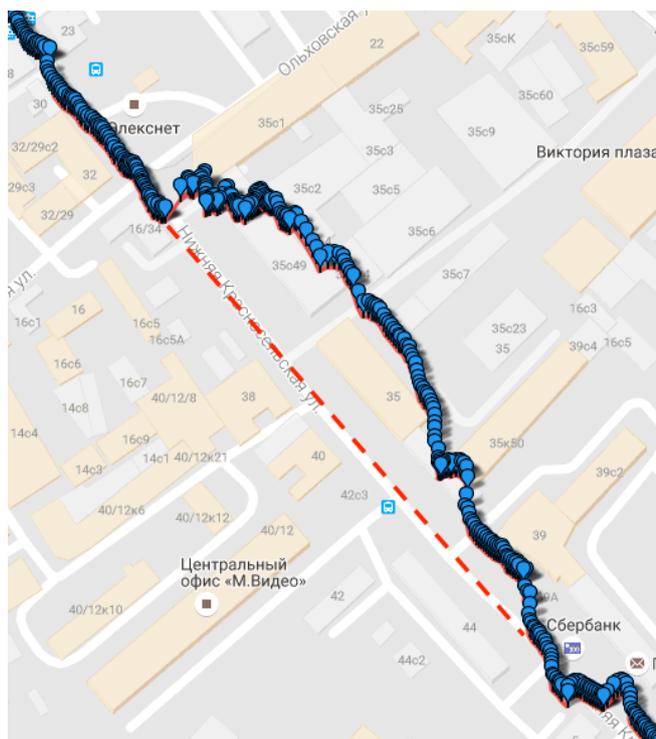


Рисунок 1.4. Отклонение трека вблизи высотных зданий

4. Ошибка определения времени фиксации геокоординат

Периодически возникают случаи, когда при движении пользователя следующая точка находится на предыдущем отрезке пути или рядом с ним (см. Рисунок 1.5).

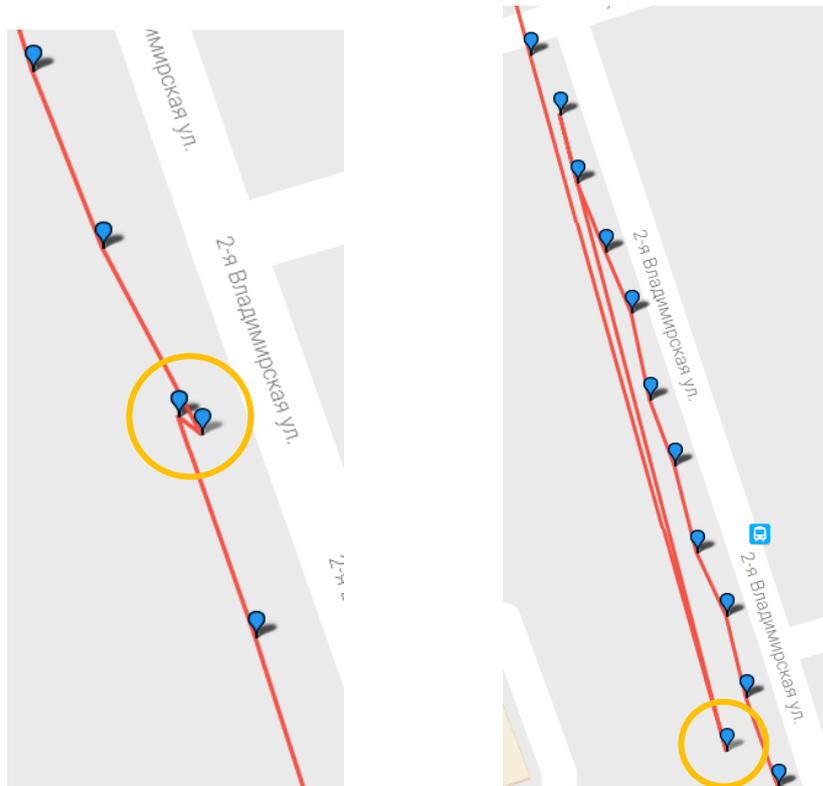


Рисунок 1.5. Ошибка определения времени фиксации геокоординат

Такое запаздывание можно определить по направлению движения, скорости и интервалу времени между точками.

Несмотря на рассмотренные проблемы обработанная информация о местоположении от спутниковых навигационных приемников как правило имеет наиболее высокую точность.

Wi-Fi точки доступа

Источник позволяет получить информацию о местоположении Wi-Fi роутера, к сети которого подключено мобильное устройство. Можно отметить следующие особенности:

- WLAN роутер, если он поддерживает привязку широты и долготы, должен иметь данные о своем местоположении;
- технически невозможно определить скорость пользователя и его перемещение только на основании одной координаты в радиусе действия роутера.

В целом данные источника лишены практически всех проблем, присущих навигационным приемникам, однако область применения ограничивается наличием роутеров вблизи пользователя.

Перспективной представляется технология WiPS/WFPS – локализация местоположения мобильного устройства на основе анализа мощности сигналов нескольких WLAN точек доступа [19, 20, 33, 80, 86]. Метод ориентирован преимущественно на навигацию внутри помещений.

Информация о местоположении от Wi-Fi точек доступа источника имеет среднюю точность.

Вышки сотовой связи

Координаты мобильного устройства можно получить по триангуляции сигнала вышек сотовой связи [87-89] при наличии GSM модуля и активном подключении к сети.

Из-за особенностей определения широты и долготы, параметры фиксации координат могут указывать на довольно высокую точность данных, однако на практике это не всегда верно (см. Рисунок 1.6).

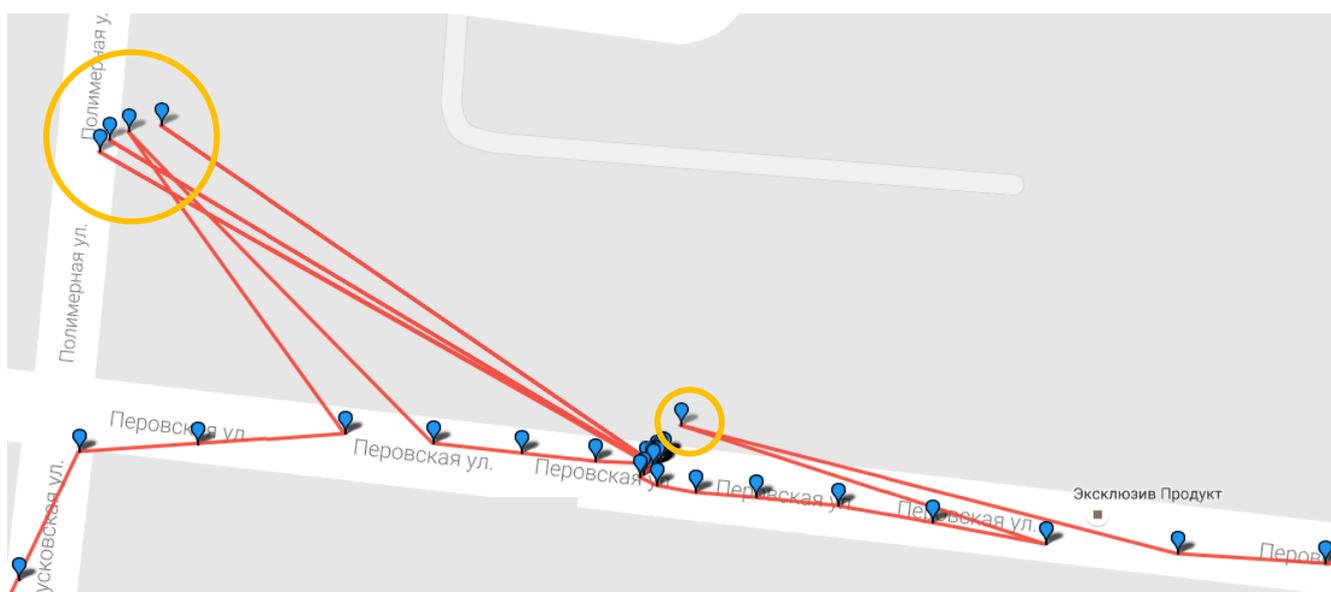


Рисунок 1.6. Выбросы координат от сотовых вышек

Источник данных имеет самую низкую точность из рассмотренных.

1.2.2. Дополнительные источники

Bluetooth маячки

Технология в общем случае аналогична WiPS/WFPS, однако вместо Wi-Fi точек доступа здесь используются Bluetooth маячки [21, 22, 24, 35]. Метод позволяет получить высокоточные данные местоположения [23], однако ограничен радиусом действия маяков и используется как правило только внутри помещений.

Пассивные источники

Принимая во внимание особенности мобильных операционных систем и мобильных технологий геолокации отметим также пассивные источники – те, которые непосредственно не инициируют определение местоположения устройства, не являются «прямыми поставщиками» геоданных. Пассивными источниками могут быть: кэш, устаревшие данные от других источников, сохраненная геоинформация от других приложений и др.

Дополнительная информация из конечного приложения

Геопозицию пользователя можно определить по действиям в мобильном

приложении. Допустим, сервисный инженер выполняет ремонт какого-либо технического объекта, точные координаты которого известны. Очевидно, что при получении достаточного объема данных о местоположении сотрудника вблизи этого объекта мы можем с большой степенью уверенности сказать, что сотрудник в данный момент находится непосредственно на этом объекте. Тем самым мы искусственно увеличиваем точность координат, полученных из основных источников. Мы называем такой метод обратной связью: пользователь – модуль обработки геоданных.

Сенсоры

В мобильных устройствах как правило присутствуют различные сенсоры: акселерометры, гироскопы, магнетометры. Анализ данных этих сенсоров может позволить определять девиации сигнала при нулевой скорости, что как следствие существенно улучшит точность позиционирования [25, 26, 31].

1.2.3. Сравнение характеристик основных и дополнительных источников

Сведем результаты анализа основных и дополнительных источников геолокационных данных в Таблицу 1.

Исходя из сравнительного анализа источников, можно выделить наиболее приоритетные источники для определения местоположения: навигационные приемники, Wi-Fi точки доступа и дополнительная информация из конечного приложения. Такие источники присутствуют практически во всех мобильных устройствах, могут использоваться как при навигации внутри помещений, так и снаружи. Однако стоит отметить, что анализ данных от многих дополнительных источников наряду с основными может существенно улучшить качество траектории движения пользователя.

Сравнение источников геолокационной информации в мобильных устройствах

Типы источника данных	Источник данных	Точность геолокации вне помещений	Точность геолокации внутри помещений	Вероятность наличия в мобильных устройствах	Особенности
Основные	Навигационные приемники	Высокая	Низкая	Высокая	Выбросы внутри помещений, блуждание координат, отклонение траектории, ошибка определения времени фиксации, очень высокая точность вне помещений
	Wi-Fi точки доступа	Средняя	Средняя	Высокая	Необходимо наличие большого числа WLAN маршрутизаторов, привязка роутеров к координатам, невозможно определить скорость пользователя в пределах одной точки доступа
	Вышки сотовой связи	Низкая	Низкая	Высокая	Очень низкая точность
Дополнительные	Bluetooth маячки	-	Высокая	Низкая	Необходимо наличие большого количества Bluetooth маячков, вне помещений практически не используются, очень высокая точность внутри помещений
	Пассивные источники	-	-	Высокая	Данные источника могут отсутствовать
	Дополнительная информация из конечного приложения	Средняя	Средняя	Средняя	Данные источника могут отсутствовать
	Сенсоры (акселерометр, магнетометр, гироскоп)	-	-	Средняя	Некоторые сенсоры могут отсутствовать

1.3. Существующие методы обработки геоданных

Проблема выбросов координат внутри помещений является одной из основных при проектировании систем фильтрации. Чтобы показать нетривиальный характер задачи, выполним анализ выбросов координат на примерах нескольких стоянок внутри помещений. На Рисунке 1.7 в верхнем левом углу изображены 86 913 координат, полученных от всех основных геолокационных источников в течение 12 часов. При этом устройство находилось в здании неподвижно.

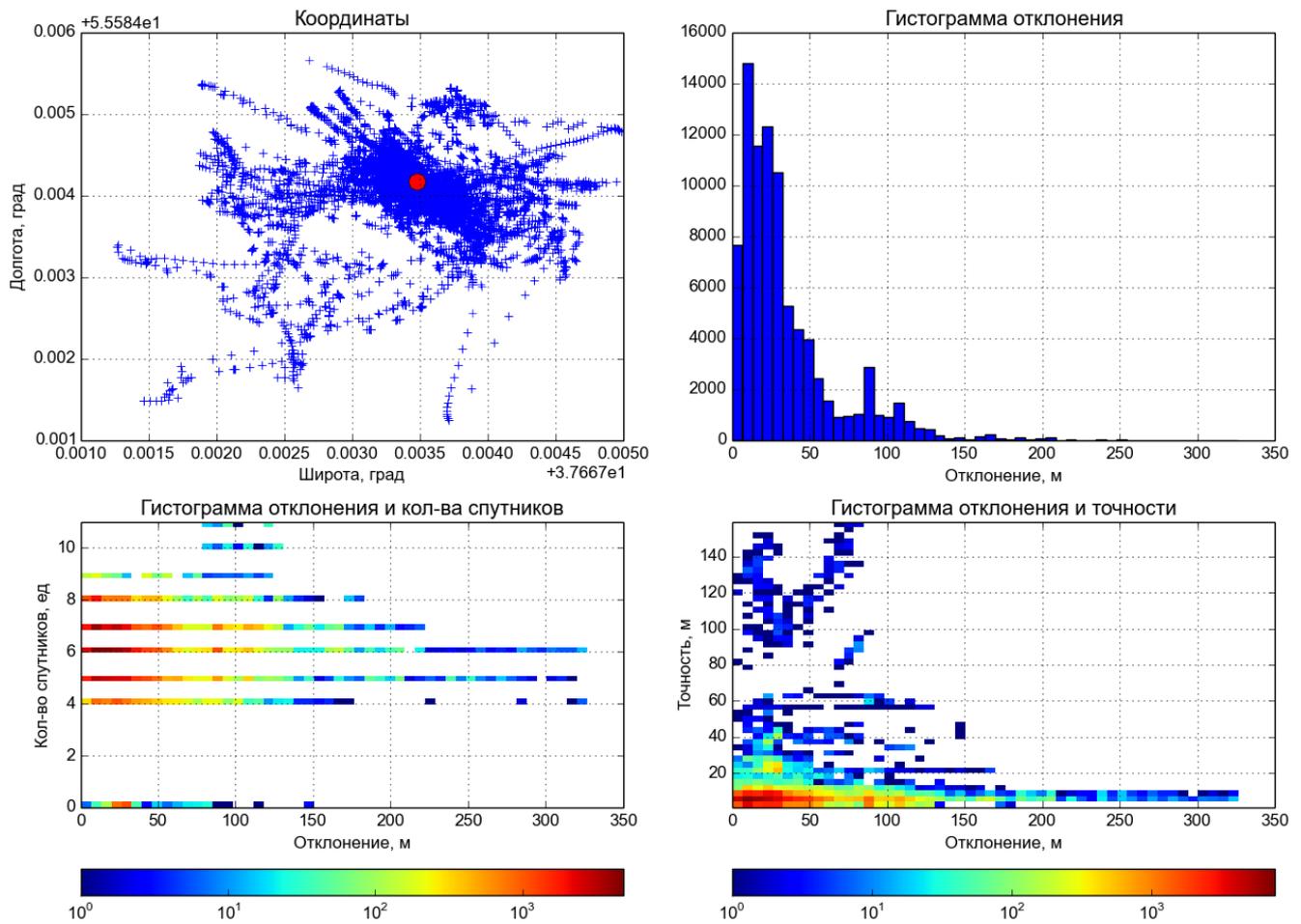


Рисунок 1.7. Анализ отклонений координат при 12-ти часовой стоянке внутри помещения

Красной точкой отмечен центр кластера, координаты которого рассчитаны как среднее арифметическое всех точек из набора. В правой верхней части показана гистограмма отклонения точек от центра кластера. Очевидно, что максимум разброса точек приходится на интервал от 0 до 50 метров.

В левой нижней части показана двумерная гистограмма отклонения точек от центра кластера и количества используемых спутников в этих точках для координат, полученных от навигационного приемника. Координаты с нулевым числом спутников относятся к Wi-Fi или сотовым источникам данных.

В правой нижней части показана двумерная гистограмма отклонения точек от центра кластера и параметра точности точек, который представлен здесь как радиус круга в метрах, в котором находится текущие координаты с вероятностью $\sim 68\%$ (1σ).

На Рисунке 1.8 приведены аналогичные графики для 3-х часовой стоянки, общее число точек 13 409.

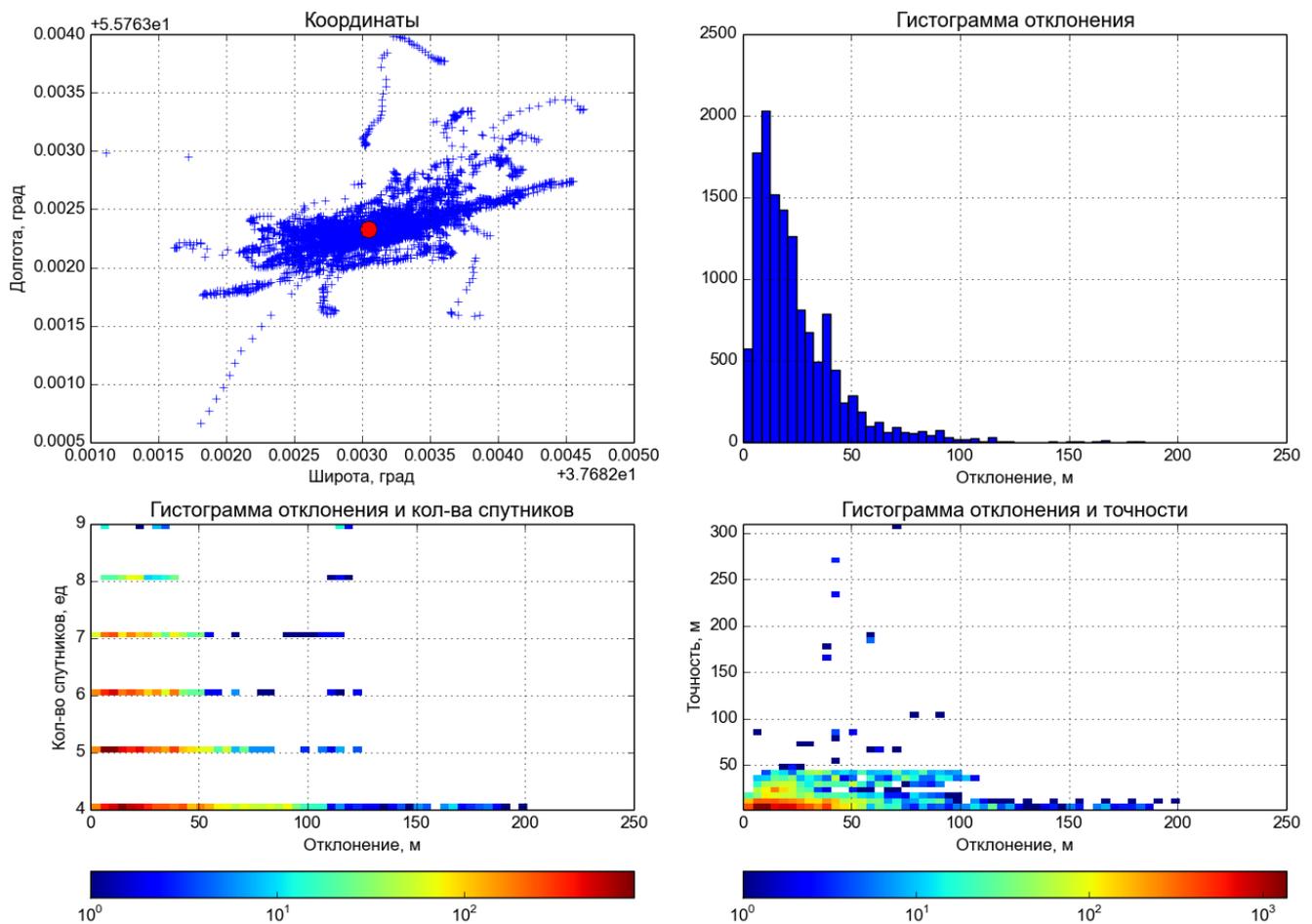


Рисунок 1.8. Анализ отклонений координат при 3-х часовой стоянке внутри помещения

Исходя из анализа графиков можно с уверенностью утверждать, что:

- максимум выбросов от 0 до 50 метров приходится на интервал точности от 0 до 20 м и интервал количества спутников от 4 до 8;

- размер выброса не зависит от кол-ва спутников и параметра точности, можно говорить лишь о зависимости их количественного распределения от числа спутников и точности;
- наибольшие выбросы как правило происходят при меньшем кол-ве спутников и меньшем значении параметра точности.

Способы и методы фильтрации геокоординат, ориентированные на решение озвученных выше задач, были детально исследованы в работах [1, 2, 4, 6, 8-15, 54, 60, 61, 74, 79, 81-83, 85, 90]. Рассмотрим некоторые из них.

Хруль и др. в работе [2] предлагает алгоритм потоковой фильтрации, основанный на обработке данных в реальном времени непосредственно на мобильных устройствах. Метод предполагает комплексный подход, важным компонентом которого является диагностическая фильтрация геоинформации на основе анализа точности смежных координат. В качестве критерия фильтрации используется расстояние между двумя точками, которое должно быть не меньше суммы пороговых значений $Dl = HDOP * HFactor + VDOP * VFactor$ для каждой точки. $HFactor$ и $VFactor$ представляют собой константы, которые зависят от типа исследуемых транспортных средств.

Проиллюстрируем идею диагностической фильтрации на примере. Входные данные представляют собой последовательности точек местоположения пользователя (см. Рисунок 1.9). Положение точек определено параметрами широты и долготы.

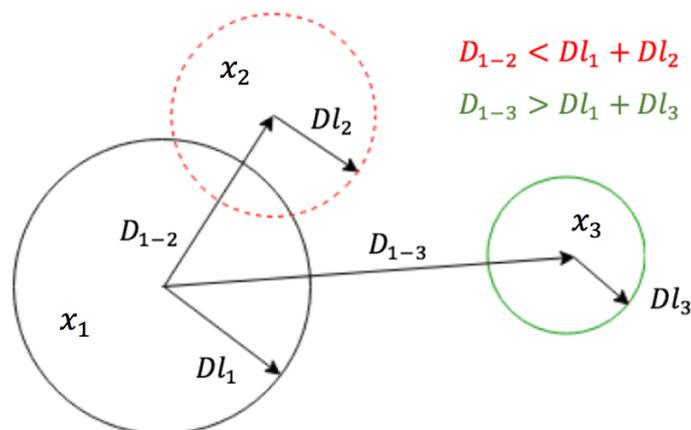


Рисунок 1.9. Иллюстрация диагностического метода

Точка x_2 не удовлетворяет критерию, поскольку $D_{1-2} < Dl_1 + Dl_2$. Следовательно, точка отбрасывается фильтром. Точка x_3 удовлетворяет критерию, поскольку $D_{1-3} > Dl_1 + Dl_3$. Следовательно, точка проходит через фильтр.

Данный подход ориентирован на уменьшение выбросов в местах стоянок, повышение достоверности и уменьшение объема входных данных, что позволяет получить качественные траектории движения объектов. Одним из главных недостатков метода является отсутствие зависимости поведения фильтра от скоростей пользователя в смежных координатах. Критерий фильтрации динамически зависит только от точности смежных координат, что в ряде случаев дает визуально некачественные треки, особенно в движении на поворотах [1].

Известно применение фильтра Калмана [4, 5, 13, 37, 84] для сглаживания геокоординат. В качестве вектора состояния здесь обычно принимают вектор $s_k = [x_k \ y_k \ v_k^x \ v_k^y]^T$, состоящий из координат объекта x_k и y_k и проекций скорости v_k^x и v_k^y , при этом без учета управляющего воздействия:

$$s_k = A_k s_{k-1} + G_k a_k, \quad (1.1)$$

$$\text{где } A_k = \begin{bmatrix} 1 & 0 & dt_k & 0 \\ 0 & 1 & 0 & dt_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$G_k = \begin{bmatrix} \frac{dt_k^2}{2} & 0 \\ 0 & \frac{dt_k^2}{2} \\ dt_k & 0 \\ 0 & dt_k \end{bmatrix};$$

$$a_k = \begin{bmatrix} a_k^x \\ a_k^y \end{bmatrix}.$$

С учетом ограничений и допущений [4] фильтр эффективно сглаживает единичные выбросы координат, однако не способен справиться с продолжительными отклонениями траектории движения (см. Рисунки 1.2 и 1.4) в условиях недостоверных значений точности и/или скорости.

Ввиду высокой сложности задачи и большого количества разнородных данных, исследователи стали часто применять модели и методы машинного

обучения [28, 48] для решения задач навигации в различном окружении [29, 30, 32, 34, 36, 53, 59]. Ключевая особенность этих подходов и их существенное преимущество над другими методами заключается в том, что для использования не требуется разрабатывать точную математическую модель процесса движения пользователя или разброса координат в точках стоянок. Например, для задачи классификации, где необходимо определить достоверность очередной координаты, необходимы только промаркированные тренировочные данные. В процессе работы алгоритма обучения модель подстраивается под входные данные и неявный процесс, которые они отражают, минимизируя ошибку – разность между ожидаемым результатом и выходным результатом работы модели. Тем самым моделируется процесс навигации пользователя; автоматически выявляются сложные паттерны движения и неявные связи между разнородными данными разнотипных источников с целью предсказания степени достоверности очередной точки местоположения. Здесь крайне важен параметр времени: GPS данные как правило распределены во времени, и именно временные характеристики очень часто помогают классифицировать стоянки, определять тип движения и в результате увеличивать качество результирующего трека пользователя. В этом случае целесообразно использовать модели машинного обучения, как-либо связанные с временными параметрами входных данных: рекуррентные нейронные сети (англ. RNN – Recurrent Neural Network) [38], LSTM (англ. Long Short-Term Memory) [39], GRU (англ. Gated Recurrent Unit) [40], ESN (англ. Echo State Network) [41], NTM (англ. Neural Turing Machine) [42].

Системные API геолокации в мобильных ОС как правило консолидируют несколько источников, предоставляя единый интерфейс для получения геоданных. Иногда API осуществляют предварительную фильтрацию, однако качество такой геоинформации оставляет желать лучшего.

Стоит отметить Fused Location Provider – интерфейс в ОС Android, призванный обобщить все источники геолокационных данных и решить проблему геолокации в помещениях [43]. Проведем сравнительный анализ Fused Location Provider API и стандартных API геолокации.

На Рисунках 1.10 и 1.11 представлены треки движения и стоянки пользователя, снятые с двух идентичных Android устройств посредством Fused Location Provider API и стандартных API геолокации соответственно. Оранжевыми кругами помечено место стоянки.

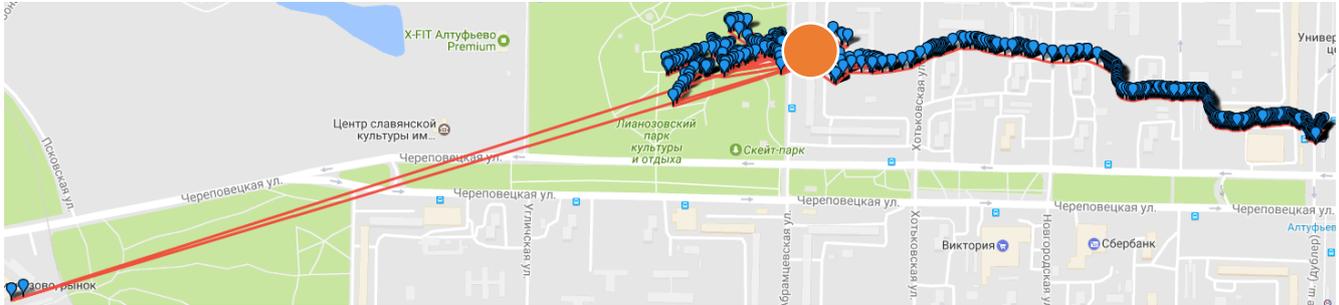


Рисунок 1.10. Трек, полученный посредством Fused Location Provider API

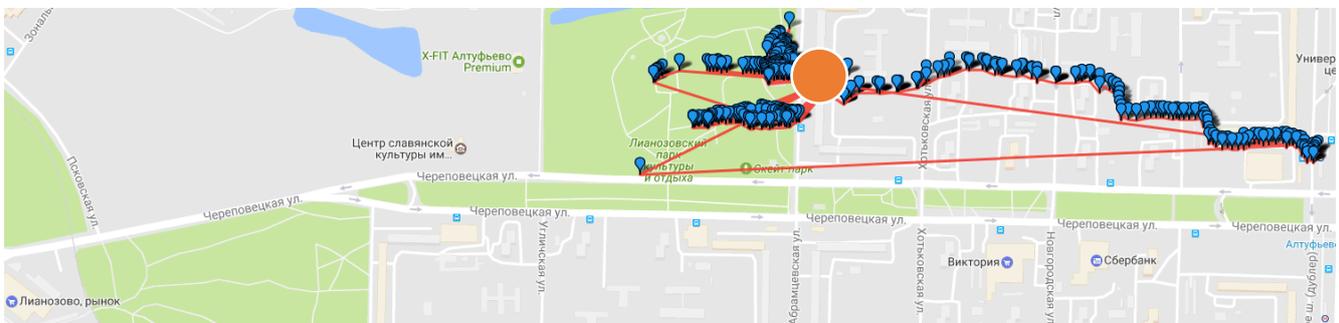


Рисунок 1.11. Трек, полученный посредством стандартных API геолокации

Треки практически идентичны. В правой части карты на участке движения в случае стандартного API можно заметить бóльшую разреженность точек. Выбросы координат в момент стоянки присутствуют в обоих случаях.

На Рисунке 1.12 представлены гистограммы отклонения, числа спутников, точности и общее распределение точек на треке. Из-за специфики работы Fused API число спутников для координат данного источника всегда равно нулю.

Графики распределения данных, полученных с помощью Fused Location Provider API, мало отличаются от графиков стандартных навигационных API. Большинство выбросов от 0 до 50~100 м происходит на интервале точности от 0 до 50 м, самые большие выбросы имеют самое низкое значение точности. Нельзя утверждать, что Fused Location Provider API производит более качественные треки так же, как и использование только Fused «поставщика» данных в дальнейшей фильтрации даст преимущество над стандартным API.

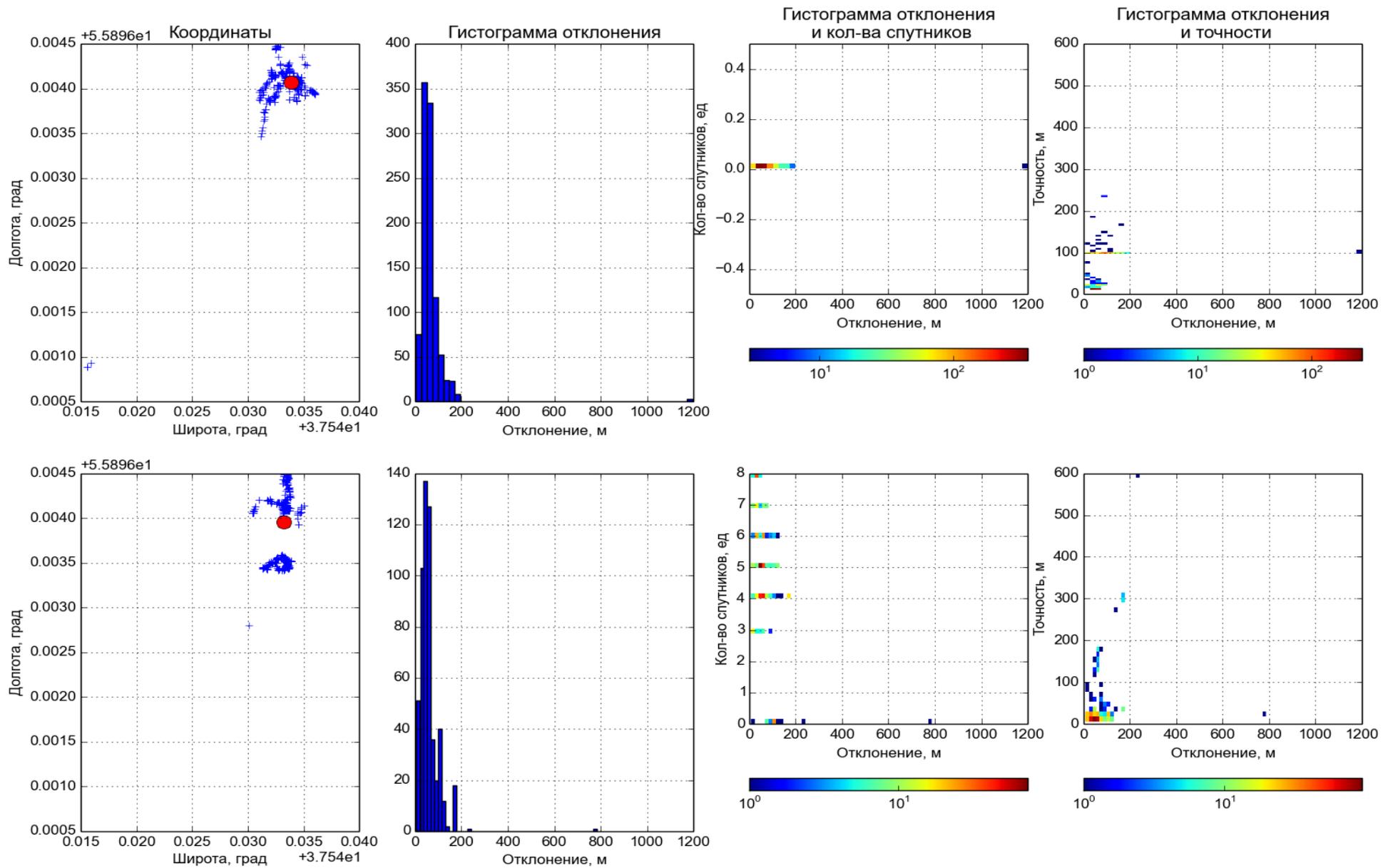


Рисунок 1.12. Сравнительный анализ Fused Location Provider API и стандартного API геолокации

1.4. Формальная постановка задачи

Выполним формальную постановку задачи увеличение степени соответствия трека реальному маршруту движения.

Пусть:

- 1) $L_{groundTruth} = L(t, x^g, y^g)$ – реальная траектория движения объекта;
- 2) $L_{raw} = \{r_i, i = \overline{1, I}\}$ – упорядоченное множество геокоординат движущегося объекта, получаемых мобильным устройством, $r_i = \langle t_i, x_i^r, y_i^r, Q_i \rangle$, характеризующихся временем получения геокоординаты t_i , широтой x_i^r , долготой y_i^r и множеством дополнительных параметров Q_i , при этом:
 - $\forall r_i = \langle t_i, x_i^r, y_i^r, Q_i \rangle \in L_{raw}: \exists! g_i = \langle t_i, x_i^g, y_i^g \rangle \in L_{groundTruth}$, то есть каждой точке из исходных данных L_{raw} соответствует точка с тем же временем t_i на реальной траектории движения $L_{groundTruth}$;
 - $\exists L'_{raw} = \{r_i \in L_{raw}, i \in M \subseteq I\} : D(g_i, r_i) \leq \varepsilon$, где D – расстояние между точками g_i и r_i , то есть точки из некоторого подмножества L_{raw} отстоят от соответствующих точек на реальном маршруте $L_{groundTruth}$ на не более ε ;
- 3) $L_{predicted} = F(L_{raw}) = \{p_i, i \in N \subseteq I\}$ – упорядоченное множество обработанных геокоординат $p_i = \langle t_i, x_i^p, y_i^p \rangle = f(r_k, k = \overline{i-j, i})$, где j – количество точек, участвующих в обработке для получения точки p_i . Условие отражает обработку в реальном времени, при этом $\forall p_i = \langle t_i, x_i^p, y_i^p \rangle \in L_{predicted}: \exists! r_i = \langle t_i, x_i^r, y_i^r, Q_i \rangle \in L_{raw}$, то есть каждой точке обработанного трека $L_{predicted}$ соответствует точка с тем же временем t_i на треке, составленном из исходных данных L_{raw} ;
- 4) $L_{approx} = L(L_{predicted})$ – кривая, интерполирующая $L_{predicted}$, при этом $\forall p_i \in L_{predicted}: \exists! a_i = p_i \in L_{approx}$, то есть каждая точка результирующего трека $L_{predicted}$ лежит на интерполирующей кривой L_{approx} .

Постановка задачи для получения оптимального решения:

дано L_{raw} , необходимо найти $L_{predicted}$ с минимальным количеством точек N , такое, что точки интерполирующей кривой L_{approx} находятся на реальной траектории движения $L_{groundTruth}$:

$$\forall a_i \in L_{approx}: d_i = D(g_i, a_i) \rightarrow 0 \text{ и } N \rightarrow \min, \quad (1.2)$$

где D – расстояние между точками $g_i \in L_{groundTruth}$ и $a_i \in L_{approx}$.

Таким образом, в данной постановке задача относится к классу задач двухкритериальной оптимизации. Минимальное количество точек отражает условие минимальной плотности точек трека.

При заданных исходных данных нахождение точного решения задачи не представляется возможным, поэтому в работе была поставлена и решена задача улучшения достигнутых на сегодня характеристик приближенного решения.

Постановка задачи улучшения существующих решений:

дано L_{raw} , необходимо найти $L_{predicted}$ с минимальным количеством точек N , такое, что точки интерполирующей кривой L_{approx} с вероятностью, превышающей достигнутую $q \geq q_{\text{достигн.}}$, отстоят от реальной траектории движения $L_{groundTruth}$ на расстояние не более допустимого $d_{\text{доп.}}$, которое меньше достигнутого к настоящему времени $d_{\text{достигн.}}$:

$$[\forall a_i \in L_{approx}: d_i = D(g_i, a_i) \leq d_{\text{доп.}} < d_{\text{достигн.}}] \geq q_{\text{достигн.}} \text{ и } N \rightarrow \min, \quad (1.3)$$

Условие (1.3) также целесообразно рассматривать как критерий качества алгоритма фильтрации при известной реальной траектории $L_{groundTruth}$, которую следует представить в виде ломаной ввиду большой сложности получения непрерывного реального маршрута. В этом случае в качестве первого критерия оптимальности возможно рассматривать минимум функции ошибки, зависящей от $L_{groundTruth}$ и $L_{predicted}$, которая сравнивает идентичность двух ломаных. Указанная задача подробно рассмотрена в главе 4.

Выводы по главе 1

Из результатов анализа типовой архитектуры ГИС, процесса обработки геоданных, стратегий разделения функций обработки, особенностей источников геоданных в мобильных устройствах, причин и характера отклонений треков от реальных маршрутов движения, существующих моделей и методов увеличения качества треков пользователя и формальной постановки задачи следуют выводы:

- 1) архитектура ГИС, в которой приоритет обработки геоинформации отдается мобильной части, является предпочтительной. В этом случае существенно снижается нагрузка на серверную часть, уменьшается объем передаваемых данных, уменьшается время отклика системы. Однако этот подход предполагает применение эффективных алгоритмов в мобильных устройствах с точки зрения емкостной, вычислительной сложности и качества результирующих данных;
- 2) источники геоинформации разделяются на два вида: основные и дополнительные. Основные непосредственно определяют местоположение устройства, дополнительные могут предоставить только косвенную информацию о геопозиции, каким-либо образом характеризую физическое положение устройства: пространственные сенсоры и др. Наиболее приоритетные источники для определения местоположения: навигационные приемники, Wi-Fi точки доступа и дополнительная семантическая информация из конечного приложения. Они присутствуют практически во всех мобильных устройствах и могут использоваться как при навигации внутри помещений, так и вне;
- 3) погрешности геокоординат в треке можно разделить на следующие типы:
 - выбросы координат при навигации внутри зданий, за счет переотражения, преломления и затухания радиосигналов внутри помещений;
 - «блуждание» координат при нулевой скорости;
 - отклонение трека вблизи высотных зданий;
 - ошибки определения времени фиксации координат;

- случайные отклонения, присущие всем источникам геоданных, в связи с внутренними ошибками определения местоположения и различными внешними факторами, воздействующими на систему;
- 4) на настоящий момент не существует методов обработки геолокационной информации в мобильных устройствах, обладающих характеристиками:
- высокая точности выходных геоданных (от 1 до 30 метров);
 - минимизация разницы между пройденными дистанциями по результирующему треку и реальному маршруту;
 - эффективное определение стоянок пользователя, в том числе в зданиях;
 - отсутствие выбросов и отклонений координат в треках при движении и в моменты стоянок пользователя в различном окружении;
 - высокая разреженность трека, малый объем выходных данных.
- 5) точного оптимального решения задачи увеличения степени соответствия трека реальному маршруту движения при отсутствии информации о реальном маршруте движения не существует, поэтому в работе по этому вопросу ставится задача улучшения достигнутых на сегодня характеристик приближенного решения.

На основе перечисленных выводов поставим следующие задачи:

- 1) проектирование модели входных и выходных данных на основе детального анализа источников данных местоположения с целью уточнения модели описания;
- 2) разработка метода обработки и фильтрации потока геоданных, исследование различных типов «выбросов геокоординат», получение расчетных соотношений для их исключения при построении трека;
- 3) разработка алгоритма последовательной обработки потока геокоординат в реальном времени, ориентированного на мобильные операционные системы, с целью исключения «выбросов», уменьшения плотности точек на прямых участках маршрута и сглаживания получаемого трека;
- 4) разработка системы фильтрации и экспериментальная проверка полученных результатов на основе предлагаемых критериев эффективности.

Глава 2. Разработка метода обработки потока геолокационных данных

Анализ источников геоданных, выбросов координат и причин их появления в главе 1 показал, что задача фильтрации геоинформации и построения трека пользователя, максимально приближенного к реальному маршруту, носит нетривиальный характер. Существующие методы, которые используют определение тех или иных показателей для анализа, в общем случае малоэффективны.

Для увеличения качества трека предлагается использовать фильтрацию потока геоданных, состоящую из последовательной обработки геокоординат от различных источников комплексом фильтров. В первых фильтрах последовательности следует обрабатывать основные характеристики геокоординат, такие как точность или число спутников. В последующих – рассчитывать и анализировать интегральные и дифференциальные показатели трека, по которым возможно определить сложные характеристики, например, является ли очередная координата, имеющая существенную погрешность, подтверждением, что пользователь в данный момент находится в состоянии стоянки, или это единичный выброс при движении пользователя, допустим, рядом с высотным зданием. Рассмотрим подробно схему обработки потока геолокационных данных.

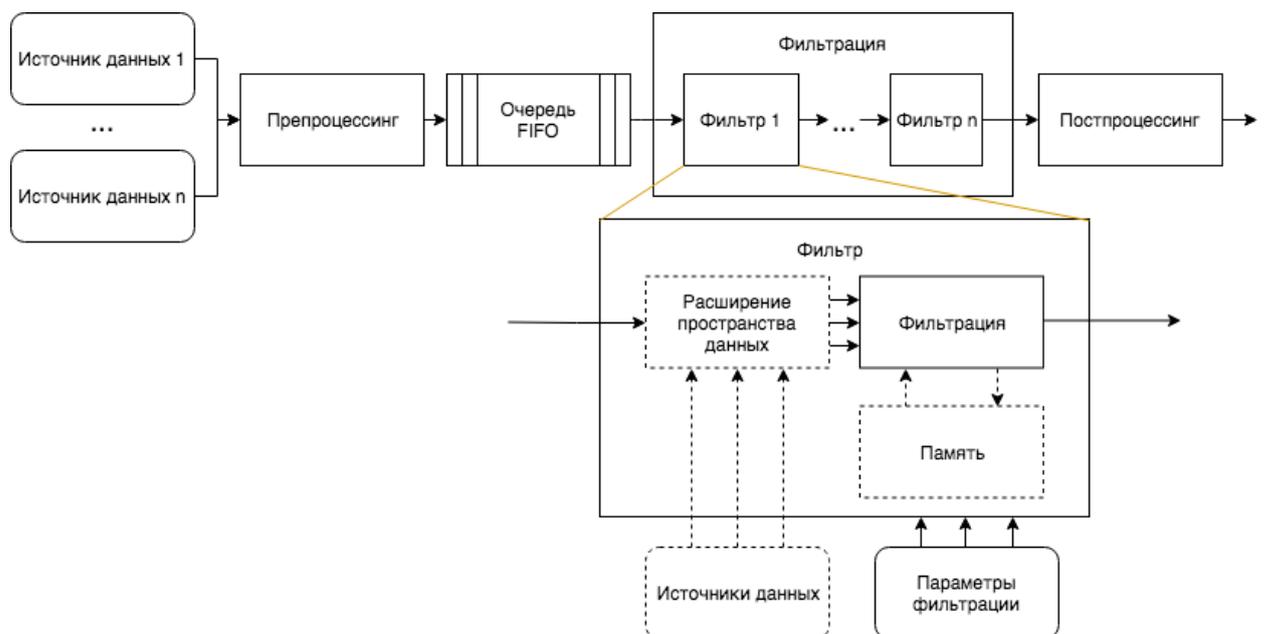


Рисунок 2.1. Схема обработки потока геолокационных данных

Поскольку данные от источников разнородны, они предварительно проходят блок препроцессинга, где происходит типизирование и унифицирование посредством приведения параметров к единому виду. После этого геокоординаты попадают в очередь FIFO.

Блок фильтрации состоит из последовательного каскада фильтров, при этом фильтр может обрабатывать как координаты от отдельных, так и от всех источников данных. Каждый блок может включать модуль расширения пространства данных, который дополняет входные данные информацией из других источников, например, акселерометра, вычисляет различные интегральные и дифференциальные характеристики точек местоположения для дальнейшей фильтрации. Параметры состоят из коэффициентов пороговых функций, параметров сглаживающих и разреживающих алгоритмов, которые регулируют поведения фильтра. Блок памяти сохраняет вычисленные на предыдущих шагах характеристики, временные и входные данные.

После фильтрации происходит постпроцессинг данных: окончательное сглаживание и разрежение.

Далее рассмотрим модель входных и выходных данных, препроцессинг и непосредственно обработку данных. Постпроцессинг рассматривается в рамках алгоритма фильтрации в главе 3, поскольку он тесно связан с особенностями ОС Android и сложно описывается на этапе проектирования.

2.1. Модель входных и выходных данных

Для разработки модели входных данных необходимо детально проанализировать информацию, поступающую с основных и дополнительных источников геоинформации, рассмотренных в главе 1. Введем переменные, которые будут использоваться при описании выходных данных от различных источников:

- $t \in \mathbb{N} > 0$ – время фиксации точки местоположения в формате Unix Epoch Time в миллисекундах;
- $lat \in \mathbb{R} \geq 0$ – широта точки местоположения в градусах;

- $lon \in \mathbb{R} \geq 0$ – долгота точки местоположения в градусах;
- $spd \in \mathbb{R} \geq 0$ – моментальная скорость в текущей точке в метрах в секунду;
- $acc \in \mathbb{R} > 0$ – радиус круга в метрах, в котором находится текущие координаты lat и lon с вероятностью $\sim 68\%$ (1σ); параметр относится к данным, полученным от API;
- $sat \in \mathbb{N} \geq 0$ – число спутников, принимающих участие в определении текущей точки местоположения;
- $hdop = \sqrt{std_{lat}^2 + std_{lon}^2} \in \mathbb{N} > 0$ – геометрическое снижение точности в горизонтальной плоскости. Параметр рассчитывается как квадратный корень из сумм квадратов среднеквадратичных отклонений измерений широты std_{lat} и долготы std_{lon} по двум осям соответственно (север-восток или юг-восток); параметр относится к данным, полученным от необработанных NMEA строк;
- $ax \in \mathbb{R} \geq 0$ – ускорение мобильного устройства по оси x в метрах в секунду в квадрате;
- $ay \in \mathbb{R} \geq 0$ – ускорение мобильного устройства по оси y в метрах в секунду в квадрате;
- $az \in \mathbb{R} \geq 0$ – ускорение мобильного устройства по оси z в метрах в секунду в квадрате.

Выходные данные всех источников будем представлять в виде кортежа перечисленных переменных.

Спутниковый навигационный приемник

Самыми распространенными NMEA типами сообщений, которые поддерживаются практически всеми навигационными приемниками являются:

- GGA – информация о фиксации геокоординат, содержит текущую широту, долготу, время съема, количество спутников и т.д.;
- VTG – данные о треке и скорости.

Препроцессинг источника заключается в разборе строк NMEA, отсеке сообщений типов, отличающихся от GGA и VTG и незначительных вычислениях: изменении размерности данных, переводе в другие единицы измерения и др.

Выходные данные спутникового навигационного приемника представляют собой кортеж:

$$x_i = \langle t_i, lat_i, lon_i, hdop_i, spd_i, sat_i \rangle, \quad (2.1)$$

где индекс $i \in \mathbb{N} \geq 0$ – порядковый номер точки местоположения. Будем придерживаться утверждения что для $\forall i: t_i - t_{i-1} \geq 0$.

Для точек, полученных с помощью стандартных API геолокации, данные представляются в виде:

$$x_i = \langle t_i, lat_i, lon_i, acc_i, spd_i, sat_i \rangle. \quad (2.2)$$

Поскольку данные спутниковых навигационных приемников могут быть получены как в качестве NMEA сообщений, так и в качестве данных API геолокации, препроцессинг здесь заключается в:

- типизировании кортежей, полученных от двух разных источников;
- дополнении кортежа (2.2) параметром $hdop$, равным нулю;
- дополнение кортежа (2.1) параметром $acc = \begin{cases} 2.5 * hdop, & hdop < 1 \\ 5 * hdop, & \text{иначе} \end{cases}$.

Wi-Fi точки доступа и вышки сотовой связи

Выходные данные источника:

$$x_i = \langle t_i, lat_i, lon_i, acc_i \rangle. \quad (2.3)$$

Параметр моментальной скорости движения здесь всегда равен нулю и в формуле опущен. При последовательной обработке координат из разных источников возникает проблема целостности данных. Рассмотрим пример. Последовательно получены 3 различных набора данных о местоположении с разницей в одну секунду. Точки x_1 и x_3 – от навигационного приемника, при этом скорость в точках равна $60 \frac{\text{км}}{\text{ч}}$. Точка x_2 – от сотовых вышек, и скорость в этой точке равна нулю. Все остальные параметры данных удовлетворяют требованиям по точности. Очевидно, что реальная скорость в точке x_2 ненулевая, т.к. скорости в точках x_1 и x_3 ненулевые, и разница во времени между точками слишком мала, чтобы

было возможно остановиться и набрать такую скорость. В этом случае возможно приближенно рассчитать примерную скорость в точке x_2 как среднее арифметическое между скоростями точек x_1 и x_3 . Однако в случае обработки потока координат, решение об актуальности координаты необходимо принимать сразу при ее получении, невозможно модифицировать скорость предыдущей координаты.

Препроцессинг заключается в:

- типизировании кортежа;
- дополнении кортежа параметрами $hdop, spd, sat$, равными нулю.

Bluetooth маячки

Выходные данные источника представляются кортежем, аналогичным (2.3). Однако точность данных, представленная параметром acc_i , здесь значительно выше. Препроцессинг аналогичен источникам Wi-Fi и сотовых вышек. Поскольку источник данных доступен мобильному приложению только при условии размещения Bluetooth маячков в каком-либо помещении или в ограниченном пространстве на улице, на этапе проектирования модели фильтрации источник рассматриваться не будет.

Пассивные источники

Как правило данные пассивных источников возможно получить с помощью API геолокации мобильной ОС и в общем случае они могут быть представлены кортежем (2.3). Препроцессинг аналогичен источникам Wi-Fi и сотовых вышек. Без ограничения общности, на этапе разработки метода фильтрации источник рассматриваться не будет.

Дополнительная информация из конечного приложения

Данные источника можно представить как:

$$x_i = \langle t_i, lat_i, lon_i \rangle. \quad (2.4)$$

Параметры точности и скорости здесь опущены, в реальных условиях их практически невозможно определить по действиям сотрудника в мобильном приложении. В общем случае данные источника не должны влиять непосредственно на фильтрацию данных, по ним можно только уточнить или заменить выходную координату после всей обработки. Поскольку источник тесно связан

со спецификой мобильного приложения, на этапе разработки метода фильтрации источник рассматриваться не будет.

Сенсоры

Практически во всех устройствах под управлением Android ОС присутствует акселерометр. Его использование в обработке геолокационной информации позволит с большей достоверностью определять движение устройства, что может существенно снизить выбросы координат в местах стоянок. Другие сенсоры в работе не рассматриваются. Данные источника представляются как:

$$a_i = \langle t_i, ax_i, ay_i, az_i \rangle. \quad (2.5)$$

Поскольку акселерометр способен выдавать только ускорения по трем осям, параметры широты, долготы и другие данные, относящиеся непосредственно к геопозиционированию здесь опущены. Препроцессинг отсутствует.

Итоговые расчетные соотношения блока препроцессинга для источников: NMEA, API, Wi-Fi, сотовые вышки представлены в Таблице 2.

Таблица 2.

Расчетные соотношения блока препроцессинга для различных источников геоданных

Источник данных	Исходные данные	Типизированные унифицированные данные и формулы для перевода
Навигационные приемники (NMEA)	$\langle t_i, lat_i, lon_i, hdop_i, spd_i, sat_i \rangle$	$\langle t_i, lat_i, lon_i, hdop_i, \mathbf{acc}_i, spd_i, sat_i, \mathbf{tp}_i, \mathbf{s}_i \rangle$ $acc_i(hdop_i) = \begin{cases} 2.5 * hdop_i, & hdop_i < 1 \\ 5 * hdop_i, & \text{иначе} \end{cases}$ $tp_i = 0$
Навигационные источники (API)	$\langle t_i, lat_i, lon_i, \mathbf{acc}_i, spd_i, sat_i \rangle$	$\langle t_i, lat_i, lon_i, \mathbf{hdop}_i, \mathbf{acc}_i, spd_i, sat_i, \mathbf{tp}_i, \mathbf{s}_i \rangle$ $hdop_i = 0, tp_i = 1$
Wi-Fi, сотовые вышки	$\langle t_i, lat_i, lon_i, \mathbf{acc}_i \rangle$	$\langle t_i, lat_i, lon_i, \mathbf{hdop}_i, \mathbf{acc}_i, \mathbf{spd}_i, \mathbf{sat}_i, \mathbf{tp}_i, \mathbf{s}_i \rangle$ $hdop_i = 0, spd_i = 0, sat_i = 0, tp_i = 2$

Параметры, выделенные **полужирным** шрифтом, отсутствуют в первоначальных данных и добавляются в блоке препроцессинга для приведения информации разнородных источников к единому виду:

- $tp_i \in \{0,1,2\}$ – тип источника;
- $s_i = \begin{cases} 1, & spd_i < P_{maxStandSpeed}^0 \\ 0, & \text{иначе} \end{cases}$ – признак стоянки, где $P_{maxStandSpeed}^0 \in \mathbb{R} \geq 0$ – общий параметр модели в метрах в секунду, характеризующий минимальную скорость пользователя, при которой точку местоположения можно считать стоянкой.

$\mathbb{R} \geq 0$ – общий параметр модели в метрах в секунду, характеризующий минимальную скорость пользователя, при которой точку местоположения можно считать стоянкой.

Ускорения, полученные от акселерометра, в таблице не рассмотрены, поскольку тесно не связаны с геоданными. Они важны для определения стоянок пользователя и будут дополнять входную информацию для фильтра стоянок, который будет рассмотрен далее.

Для геолокационной информации в качестве входных и выходных данных на данном этапе будем рассматривать кортеж, состоящий из всех параметров, рассмотренных ранее:

$$x_i = \langle t_i, lat_i, lon_i, hdop_i, acc_i, spd_i, sat_i, tp_i, s_i \rangle. \quad (2.6)$$

Данные акселерометра определим как:

$$a_i = \langle t_i, ax_i, ay_i, az_i \rangle. \quad (2.7)$$

2.2. Метод фильтрации геоданных

Далее в разделе рассматриваются 9 фильтров в последовательности, выделенных по функциям и группам параметров обработки, которые позволят значительно увеличить качество результирующего трека пользователя. Первые фильтры в наборе основаны на существующих методах обработки и фактически являются пороговыми функциями от различных параметров координат, например, порог по значению точности acc_i или числа спутников sat_i . С целью гибкой настройки модели каждый из порогов зависит от определенных коэффициентов, задаваемых статически перед этапом непосредственной работы модели или динамически в зависимости от различных параметров координат. При достижении значений максимальных и/или минимальных порогов фильтр отбросит очередную координату. Последующие фильтры рассчитывают сложные интегральные

или дифференциальные характеристики движения или стоянки пользователя, например, среднеквадратичное отклонение скорости, по которым возможно правильно идентифицировать выбросы или отклонения координат с заведомо высокой погрешностью параметра точности acc_i . Такой подход позволяет снизить вычислительную и емкостную сложность фильтрации за счет отбора координат в первую очередь менее ресурсоемкими алгоритмами. Порядок фильтров в общем случае может быть как статическим, так и динамическим, однако выявление оптимальной статической конфигурации или разработка алгоритмов динамического изменения последовательности – отдельная нетривиальная задача, что в работе рассматриваться не будет. Результатом фильтрации здесь является кортеж модифицированных геоданных и класс точки: валидная или нет, 1 или 0.

Каждый фильтр рассматривается далее как функция $F_j(\{x_i\}, \{P^j\}) \in \{0,1\}$, где $j \in \mathbb{N} \geq 1$ – порядковый номер фильтра в последовательности; $i \in \mathbb{N} \geq 0$ – порядковый номер координаты; $\{x_i\}$ – упорядоченное множество входных точек для фильтрации: фильтр может принимать на вход как только текущую координату, так и несколько предыдущих координат; если аргументы функции содержат переменные с индексами, меньше i , то фильтр должен содержать блок памяти; $\{P^j\}$ – параметры фильтрации. В качестве x_i будем описывать кортеж только из тех характеристик координаты, которые использует блок фильтрации. Функция F_j принимает значение 0, если фильтр не пропускает координату на выход, 1 – иначе.

2.2.1. Фильтр по точности

Данный метод широко применяется в обработке геолокационных данных, его цель состоит в том, чтобы отбросить на начальном этапе те координаты, значения параметров точности которых выше определенного порога. В общем случае в качестве параметра точности здесь используется acc_i – радиус круга, в котором находится пользователь с вероятностью 68 % (1σ). Дополнительно для

NMEA источников возможно анализировать параметр погрешности $hdop_i$, для всех спутниковых координат – число спутников sat_i .

На Рисунке 2.2 представлены гистограммы точности из выборки в 368528 координат, собранных с различных Android устройств, представляющие различные треки пользователей: стоянки, движения пешком, на автомобиле, поезде, метро, велосипеде; здесь и далее ось ординат логарифмическая; параметр точности acc_i для NMEA координат получен в соответствии с Таблицей 2:

- по всем типам координат – левый верхний угол, среднее значение $\mu = 37.883$ м, $\sigma = 154.134$ м;
- по типу NMEA – правый верхний угол, $\mu = 10.243$ м, $\sigma = 91.91$ м;
- по типу API – левый нижний угол, $\mu = 17.666$ м, $\sigma = 34.553$ м;
- по типу Wi-Fi и сотовых вышек – правый нижний угол, $\mu = 175.672$ м, $\sigma = 336.380$ м.

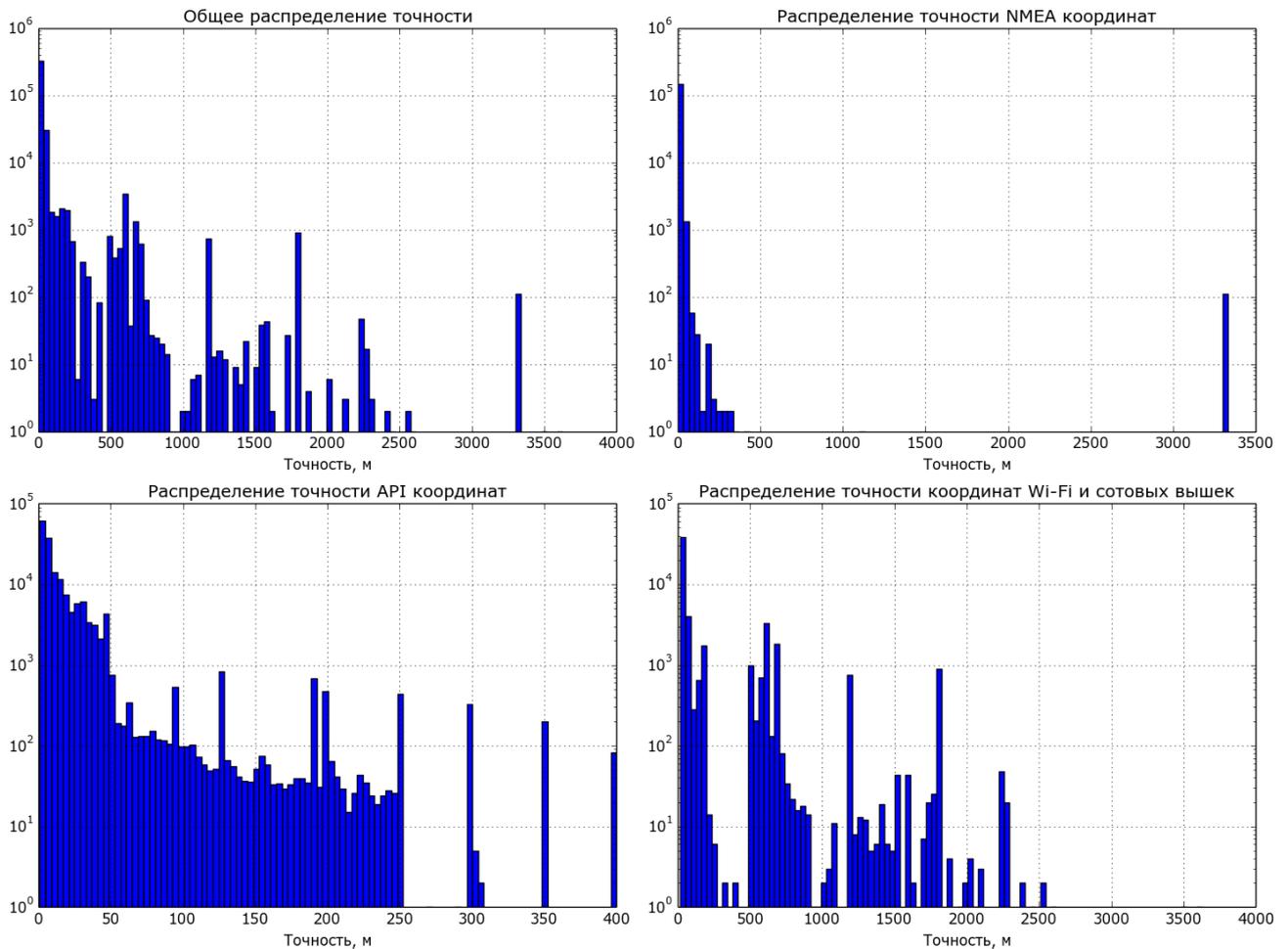


Рисунок 2.2. Распределение точности

С учетом допуска в 3σ точность координат:

- всех типов распределена на отрезке [0 м; 500.284 м];
- типа NMEA распределена на отрезке [0 м; 285.973 м];
- типа API распределена на отрезке [0 м; 121.324 м];
- типа Wi-Fi и сотовых вышек распределена на отрезке [0 м; 1184.813 м].

Координаты NMEA и API в общем случае находятся на относительно небольшом отрезке, однако привнесение в общий список координат от сотовых вышек и Wi-Fi роутеров существенно меняет распределение точности. На всех гистограммах можно заметить выбросы – кластеры координат во окрестностях многих точек. Причина такого поведения – внутренняя реализация алгоритмов определения местоположения для каждого типа источника координат. Рассмотрим гистограммы относительно каждого источника.

Отрезок [0 м; 450 м] распределения NMEA координат представлен на Рисунке 2.3.

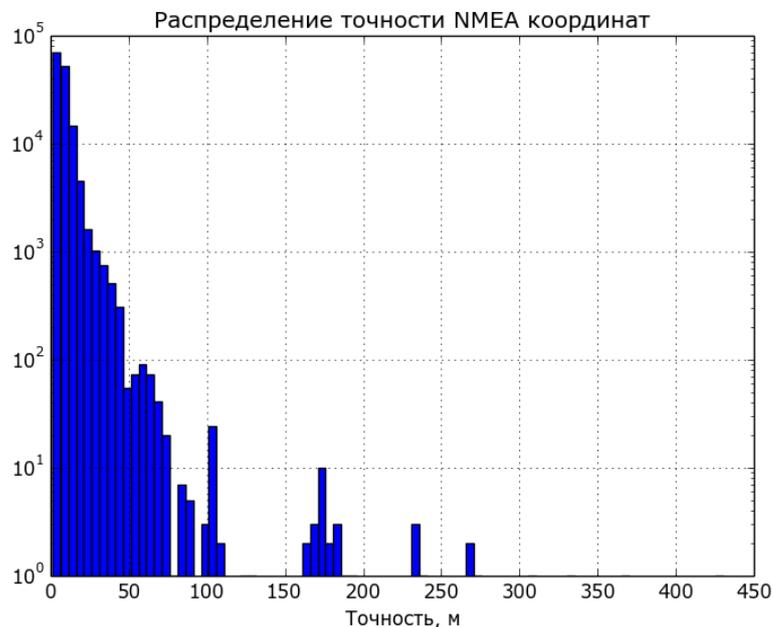


Рисунок 2.3. Распределение точности NMEA координат

В общем распределении NMEA координат с довольно большой степенью уверенности можно говорить всего о двух явных кластерах точек. Центры кластеров в результате анализа данных алгоритмом K-Means – 7.719 м и 3333 м. Количество координат окрестностях 175 м и 250 м мало относительно их числа в

полученных центрах двух кластеров, поэтому небольшие кластеры в этих точках допустимо не рассматривать.

Распределение API координат имеет 8 кластеров: 10 м, 47.259 м, 68.882 м, 96.362 м, 127.75 м, 158.682 м, 195.719 м, 244.762 м, 300.038 м, 350 м и 400 м.

Распределение сетевых координат имеет 5 кластеров: 100 м, 608.318 м, 1184.836 м, 1773.97 м и 2289.375 м.

Рассмотрим гистограмму распределения спутников (см. Рисунок 2.4) для NMEA и API координат.

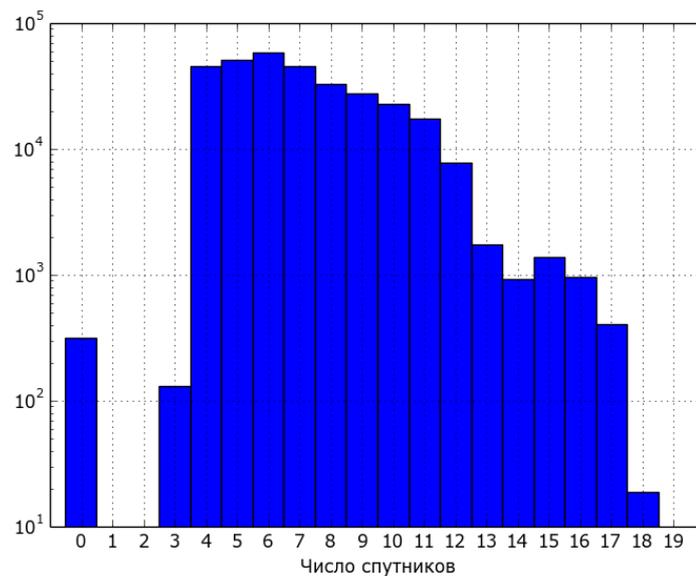


Рисунок 2.4. Распределение точности NMEA координат

Периодически Android устройства не могут определить точное число спутников, по которым была определена геопозиция. В таком случае координате устанавливается нулевое число спутников, что видно на гистограмме. Такие координаты не будут рассматриваться в дальнейшем анализе. Среднее значение числа спутников $\mu = 7$, $\sigma = 2.409$. С учетом допуска в 3σ координаты распределены на отрезке $[0; 14]$. Можно предположить, что точки, определенные по большему количеству спутников, всегда имеют большую точность, однако данное предположение неверно. Рассмотрим зависимость распределения точности от числа спутников (см. Рисунки 2.5 и 2.6).

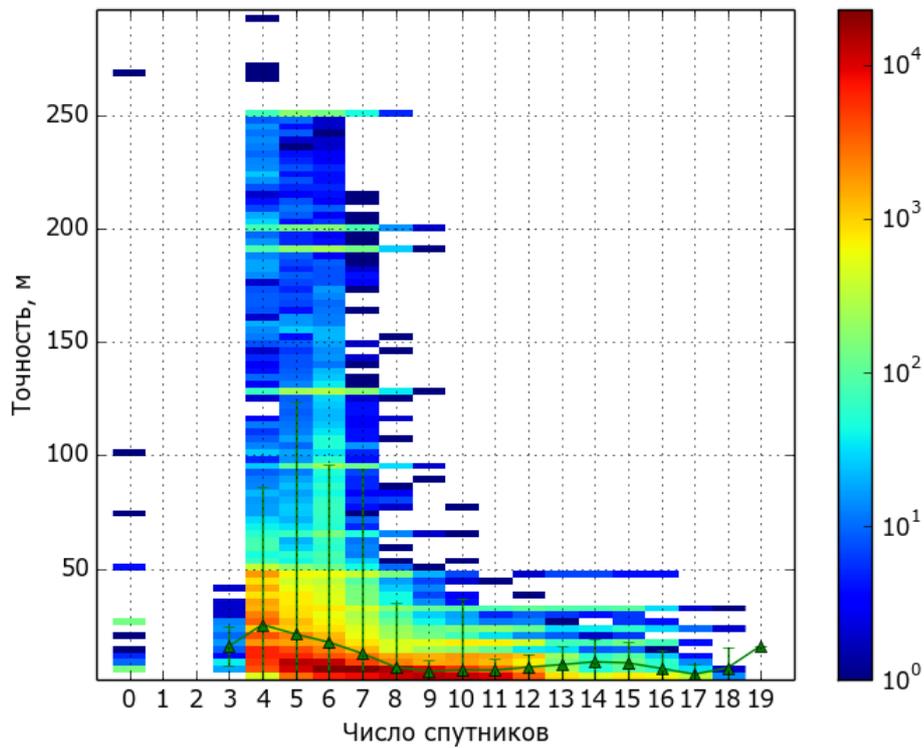


Рисунок 2.5. Распределение точности и числа спутников

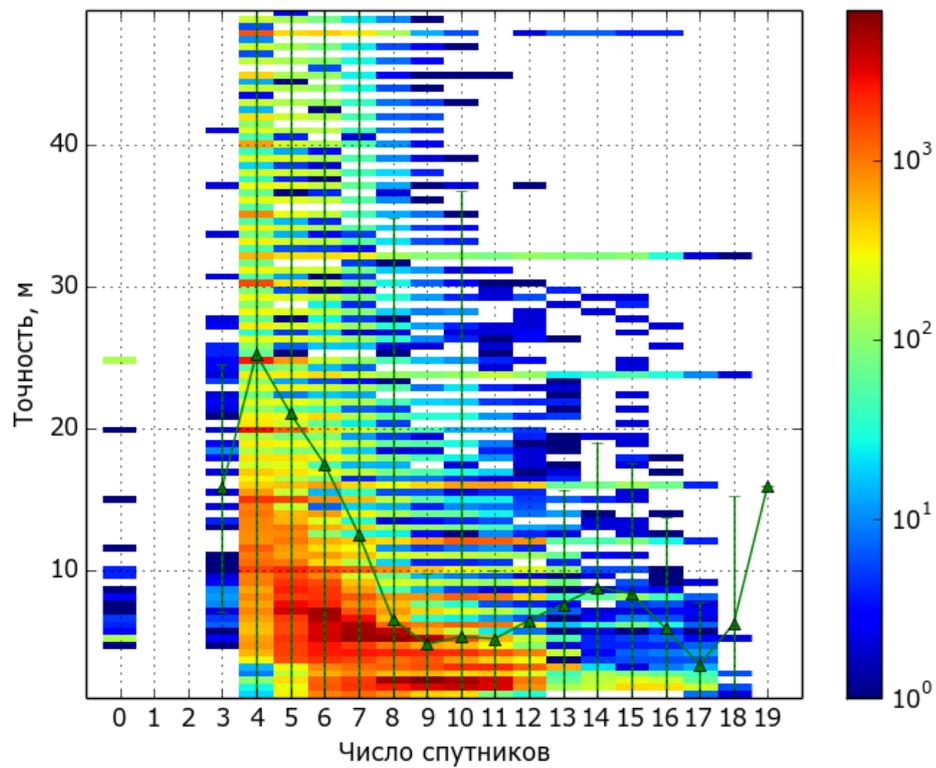


Рисунок 2.6. Распределение точности и числа спутников

Зеленая ломанная линия на графике проведена через точки, соответствующие средним значениям точности для каждого числа спутников, отклонение в каждой точке определяется одной сигмой (см. Таблицу 3).

Таблица 3.

Среднее значение и среднеквадратичное отклонение точности от числа спутников

Число спутников	Среднее значение μ , м	Среднеквадратичное отклонение σ , м
3	15.843	8.745
4	25.239	60.924
5	21.067	102.73
6	17.489	78.402
7	12.555	81.75
8	6.553	28.282
9	4.794	5.023
10	5.3	31.502
11	5.176	4.825
12	6.478	5.817
13	7.612	8.008
14	8.83	10.16
15	8.342	9.271
16	5.916	7.856
17	3.352	4.437
18	6.208	9.051
19	16	0

Максимальное среднее значение точности 25.239 м соответствует количеству спутников, равному 4, минимальное 3.352 м достигается при числе спутников 17. Общая тенденция явно прослеживается на гистограмме: при увеличении количества спутников средние значения точности убывают, следовательно,

предположение о том, что точная координата всегда определяется по большему числу спутников верно только для средних значений параметров. Ковариация между числом спутников и точностью равна -15.862 , коэффициент корреляции составляет -0.097 .

Можно выявить следующие интересные аспекты поведения графика:

- среднее значение и среднеквадратичное отклонение точности при числе спутников 4 значительно больше, чем при количестве спутников 3, это противоречит здравому смыслу;
- среднеквадратичное отклонение точности при числе спутников 10 примерно в 6 раз больше, чем при количестве спутников 9 или 11;
- при числе спутников больше 12, график, проведенный через средние значения точности, незначительно плавно возрастает и убывает на минимум при числе спутников 17.

Причин такого поведения может быть несколько: недостаточно большая выборка для анализа, специфика алгоритмов определение местоположения по спутникам, недостоверность параметра точности: отражение, преломление и затухание GPS сигналов внутри зданий и вблизи высотных домов.

В общем случае в фильтре по точности целесообразно статически ограничивать верхний порог точности значением от 5 до 50 м. В первом случае, в результирующий трек будут попадать примерно 32% всех точек, во втором – 94%. Точность выше 50 м довольно высока и в ГИС обычно не представляет никакой ценности для построения трека пользователя. Нижний порог числа спутников следует взять равным 8 или 9.

Можно также предложить динамически ограничивать порог точности по результатам анализа среднего значения и дисперсии координат, пришедших в устройство за определенный промежуток времени. Таким образом возможно гибко настраивать фильтрацию в зависимости от качества GPS приемника, сдвигая допустимый предел точности вниз, если среднее значение и дисперсия низкие, вверх – если высокие. В этом случае следует также дополнительно статически ограничивать верхний порог. Аналогичный метод возможно применить для

анализа характеристики $hdop$ и/или количества спутников. Динамический порог может также определяться исходя из вычисленных центров кластеров точек.

Реализация динамических порогов предполагает детальный анализ распределения точности координат в различных условиях для разных спутниковых навигационных приемников, что является нетривиальной задачей и выходит за рамки данной работы.

Работа фильтра зависит от достоверности параметров точности acc и отклонения $hdop$. Если ошибка их вычисления значительна, допустим при движении пользователя около высоких зданий или в местах стоянок, фильтр будет работать менее эффективно с точки зрения качества результирующего трека. Итоговое упрощённое математическое выражение для фильтрации координат статическим порогом:

$$F_1(\langle sat_i, hdop_i, acc_i, tp_i \rangle, P_{minSat}^1, P_{maxHdop}^1, P_{maxAcc}^1) = \begin{cases} 0, & \left[\begin{array}{l} (tp_i = 0 \text{ или } tp_i = 1) \text{ и } sat_i < P_{minSat}^1 \\ tp_i = 0 \text{ и } hdop_i > P_{maxHdop}^1 \\ acc_i > P_{maxAcc}^1 \end{array} \right. \\ 1, & \text{иначе} \end{cases}, \quad (2.8)$$

где $P_{minSat}^1 \in \mathbb{N} > 0$ – параметр модели, определяющий минимальное количество спутников, для координат, полученных из навигационных приемников;

$P_{maxHdop}^1 \in \mathbb{R} > 0$ – параметр модели, определяющий максимальное значение характеристики $hdop$ для NMEA координат;

$P_{maxAcc}^1 \in \mathbb{R} > 0$ – параметр модели, определяющий максимальное значение точности координат всех типов в метрах.

2.2.2. Фильтр сетевых координат

В общем случае точность координат, полученных от Wi-Fi роутеров или вышек сотовой связи, намного ниже, чем точность спутниковых геокоординат. Если в данный момент навигационный приёмник включен, и устройство имеет хорошую связь со спутниками, для предотвращения влияния низкой точности

сетевых координат на следующие фильтры целесообразно ограничивать этот тип. Однако если продолжительное время мобильное устройство не имеет связи со спутниками, для восстановления определения геопозиции следует вернуть в общую последовательность координаты от Wi-Fi и сотовых вышек.

На Рисунке 2.7 представлена гистограмма распределения времени отсутствия спутниковых координат: среднее значение $\mu = 3.775$ с, $\sigma = 84.512$ с.

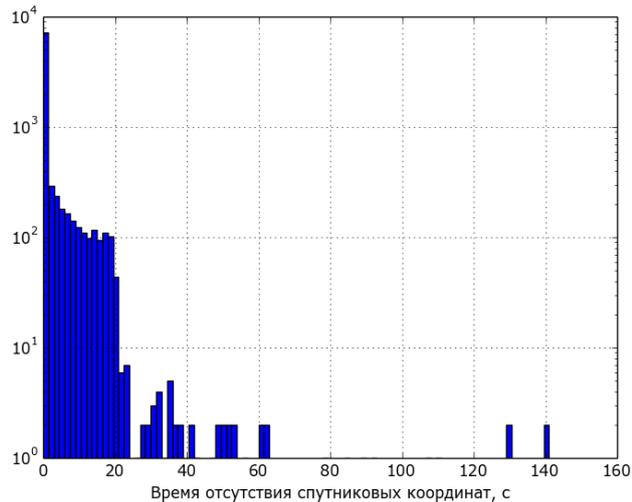


Рисунок 2.7. Распределение времени отсутствия спутниковых координат

На Рисунке 2.8 показано распределение времени отсутствия сетевой координаты с появления спутниковой в зависимости от времени отсутствия спутниковых координат в момент появления сетевой.

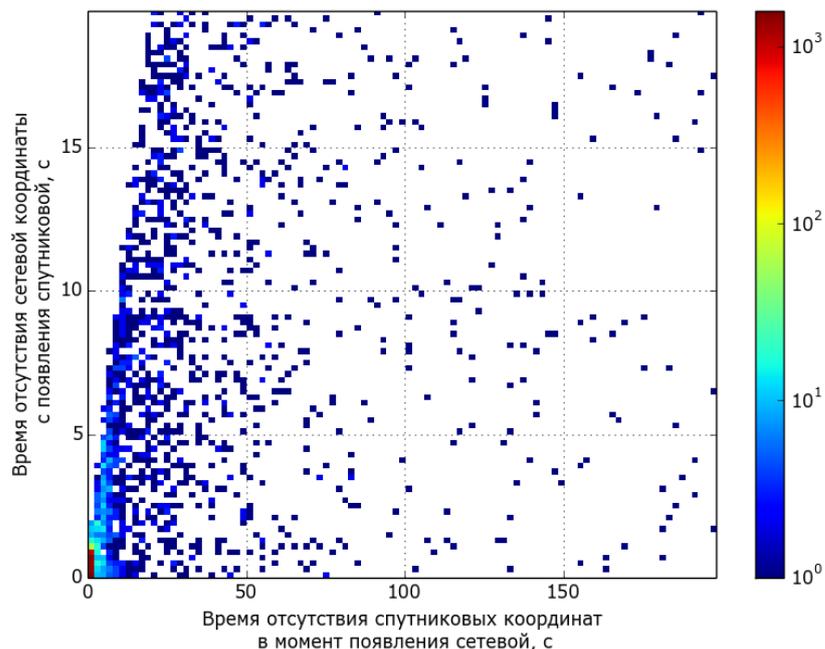


Рисунок 2.8. Распределение времени отсутствия спутниковых координат

Анализируя графики, можно отметить, что время отсутствия спутниковых координат в момент появления сетевой распределено на отрезке $[0 \text{ с}; 4972.255 \text{ с}]$ ($3\sigma; \mu = 90.513 \text{ с}, \sigma = 1627.247 \text{ с}$), а время отсутствия сетевой координаты с появления спутниковой – $[0 \text{ с}; 257.325 \text{ с}]$ ($3\sigma; \mu = 3.775 \text{ с}, \sigma = 84.517 \text{ с}$). Поскольку последний отрезок полностью входит в первый, логично предположить, что возвращение сетевых координат в фильтрацию позволит существенно повысить точность и сократить время отклика системы.

Поскольку стоянки являются одним из главных характеристик трека пользователя, целесообразно ограничивать типы координат в этом фильтре, только если пользователь в данный момент находится в движении. Тем самым фильтр сможет пропускать все координаты, которые потребуются для более точного определения времени стоянки на следующих шагах.

Итоговое выражение фильтра сетевых координат:

$$F_2(\langle t_i, tp_i \rangle, P_{lastGPSTime}^2, P_{networkPeriod}^2, Q_{isStand}) = \begin{cases} 0, & Q_{isStand} = 0 \text{ и } tp_i = 2 \text{ и } t_i - P_{lastGPSTime}^2 < P_{networkPeriod}^2, \\ 1, & \text{иначе} \end{cases} \quad (2.9)$$

где $P_{lastGPSTime}^2 \in \mathbb{N} \geq 0$ – время последней спутниковой координаты в миллисекундах, которое должно динамически рассчитываться при поступлении координаты с типом 0 или 1,

$P_{networkPeriod}^2 \in \mathbb{N} > 0$ – параметр модели, определяющий максимальное время в миллисекундах, в рамках которого будут исключены сетевые координаты;

$Q_{isStand} \in \{0,1\}$ – общий параметр модели, который определяет находится ли пользователь в состоянии стоянки или нет. Параметр будет подробно рассмотрен в фильтре стоянок далее.

2.2.3. Фильтр по времени

Минимальное время между получением координат из различных источников может составлять менее одной секунды. В целях разрежения данных или при

ограничении какой-либо ГИС на минимальный интервал времени между координатами, следует разработать фильтр, основное назначение которого – уменьшение плотности координат. На Рисунке 2.9 представлена гистограмма распределения времени отсутствия координат, среднее значение $\mu = 3.336$ с, $\sigma = 78.14$ с.



Рисунок 2.9. Распределение времени отсутствия координат

Исходя из предпосылки, что API координаты от спутниковых навигационных приемников в общем случае точнее NMEA, что было доказано результатом анализа собранной статистики, на этом этапе также целесообразно расширить поведение фильтра так, чтобы он не пропускал координаты NMEA, отдавая предпочтение API, если за настроенный интервал времени были координаты с типом API. На Рисунке 2.10 показаны гистограммы распределения времени отсутствия NMEA и API координат.

Средние значения равны 2.387 с и 6.435 с, среднеквадратичные отклонения – 151.04 с и 393.857 с соответственно. С уверенностью можно сказать, что NMEA координаты появляются на мобильном устройстве гораздо чаще API.

Аналогично предыдущему фильтру, здесь также следует ограничивать типы координат, только если пользователь в данный момент находится в движении.

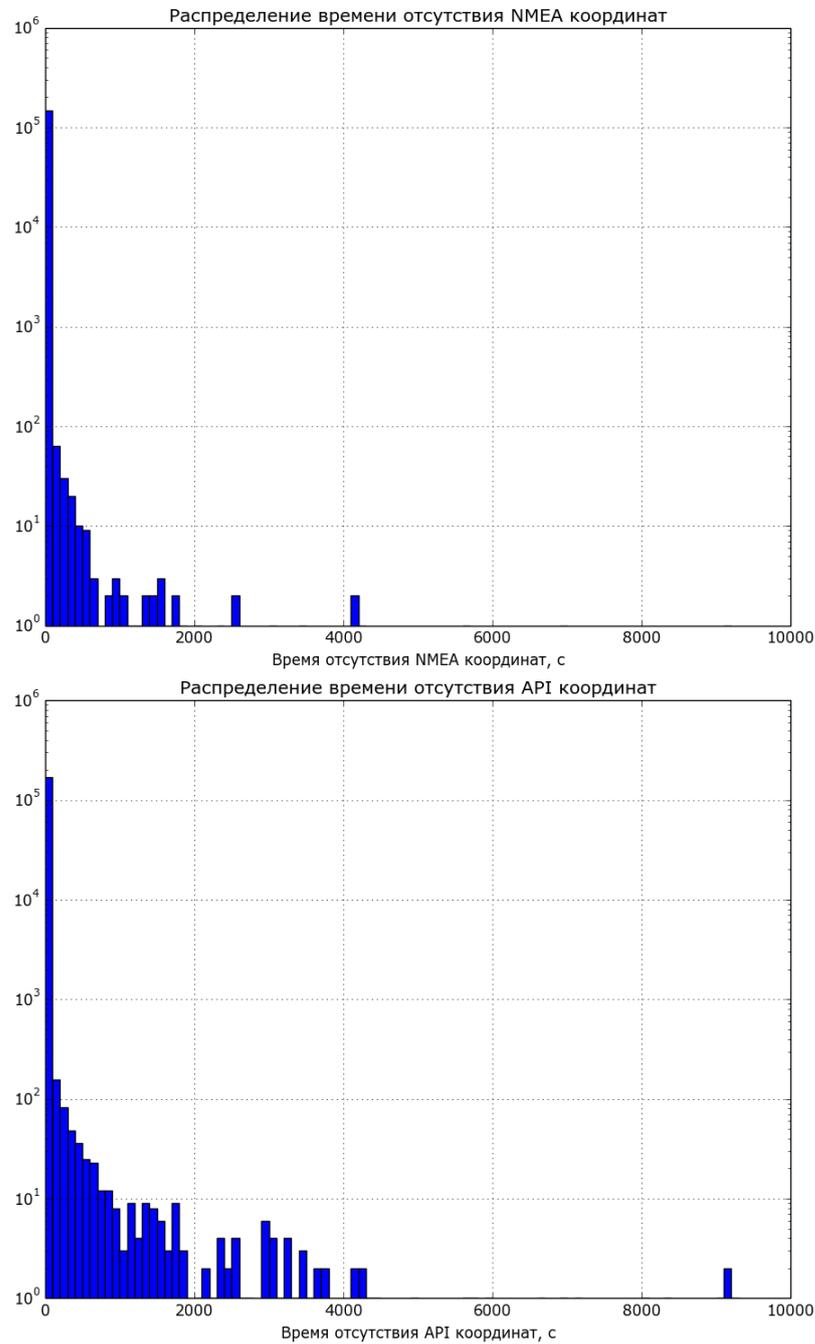


Рисунок 2.10. Распределение времени отсутствия NMEA и API координат

Итоговая функция F фильтра отдает приоритет API координатам:

$$F_3(\{\langle t_i, tp_i \rangle, \langle t_{i-1} \rangle\}, P_{period}^3, P_{lastLocationTime}^3, P_{locationPeriod}^3, Q_{isStand}) = \begin{cases} 0, & \left[t_i - t_{i-1} < P_{period}^3 \right. \\ & \left. Q_{isStand} = 0 \text{ и } tp_i = 0 \text{ и } t_i - P_{lastLocationTime}^3 < P_{locationPeriod}^3 \right] \\ 1, & \text{иначе} \end{cases} \quad (2.10)$$

где $P_{period}^3 \in \mathbb{N} > 0$ – параметр модели, период съема координат в миллисекундах; блок обработки будет гарантировать, что на выход будут поступать координаты, интервал времени между которыми не меньше значения этого параметра;

$P_{lastLocationTime}^3 \in \mathbb{N} \geq 0$ – время последней API координаты в миллисекундах, которое должно динамически рассчитываться при поступлении координаты с типом 1;

$P_{locationPeriod}^3 \in \mathbb{N} > 0$, мс – параметр модели, время в миллисекундах, в рамках которого будут исключены NMEA координаты.

2.2.4. Фильтр по скорости

Все NMEA и API координаты содержат параметр моментальной скорости. Выполним анализ распределения скорости пользователя в треках (см. Рисунок 2.11)

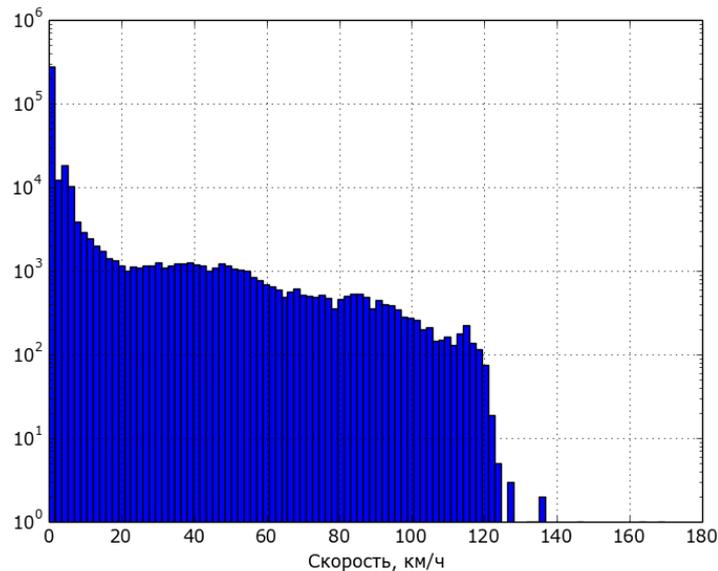


Рисунок 2.11. Распределение скорости

Скорость распределена относительно равномерно: среднее значение $\mu = 6.622 \frac{\text{км}}{\text{ч}}$, $\sigma = 18.177 \frac{\text{км}}{\text{ч}}$; при условии допуска в 3σ скорость распределена на отрезке $[0 \frac{\text{км}}{\text{ч}}; 61.154 \frac{\text{км}}{\text{ч}}]$. Рассмотрим зависимость распределения точности от скорости (см. Рисунки 2.12 и 2.13).

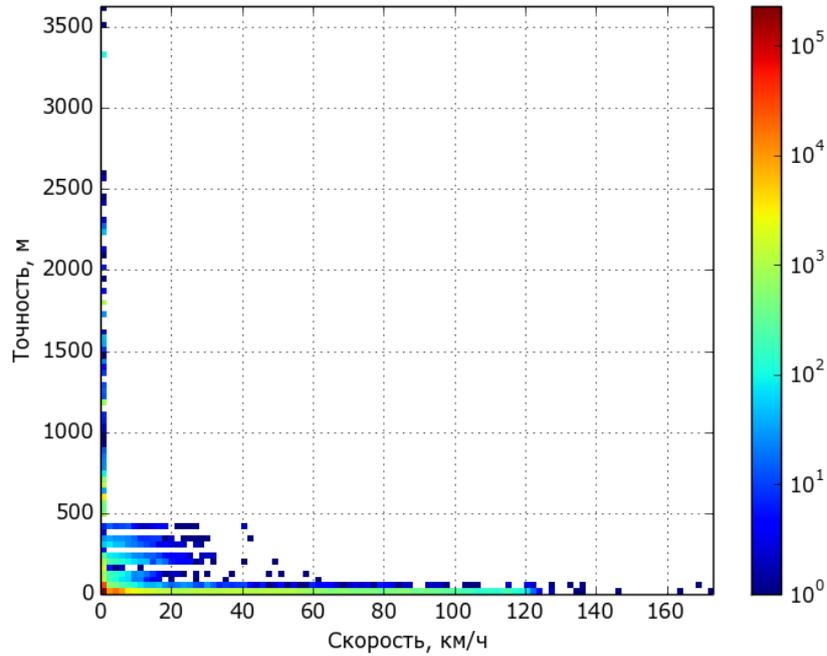


Рисунок 2.12. Зависимость распределения точности от скорости (1)

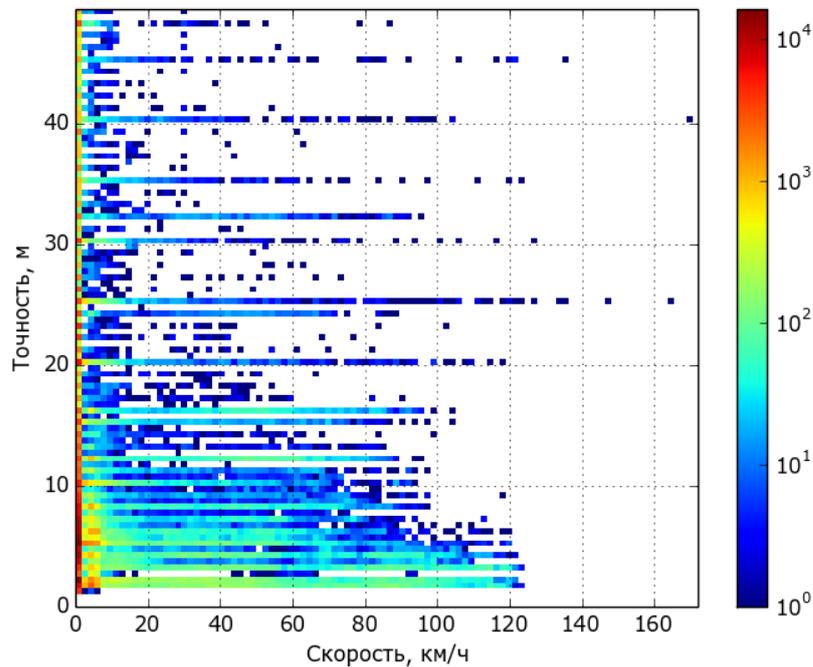


Рисунок 2.13. Зависимость распределения точности от скорости (2)

Стоит отметить, что ковариация здесь равна -196.730 , а коэффициент корреляции -0.070 , что означает уменьшение параметра точности при увеличении скорости. Если рассматривать выборку без нулевых значений скорости, то ковариация станет равной -156.683 , а коэффициент корреляции -0.146 . Очевидно, что

при больших значениях скорости, погрешность определения местоположения становится ниже. Следовательно, целесообразно разработать, ограничивающий координаты по параметру скорости, это может улучшить качество результирующего трека.

По параметру максимальной скорости и интервалу времени между предыдущей и текущей координатой можно также определить максимальную дистанцию, которую мог преодолеть пользователь, и сравнить ее с реальной. В случае, если отношение реальной дистанции к максимальной больше определенного коэффициента, следует отфильтровать текущую координату, так как с большой степенью вероятности она не точная. Между предыдущей и текущей координатой может быть довольно большой промежуток времени, анализ координат здесь следует ограничить максимальным интервалом времени. Указанный метод отчасти решает задачу фильтрации координат с недостоверными параметрами точности *acc* и отклонения *hdop* в местах стоянок пользователя, косвенно определяя реальную погрешность определения местоположения.

Итоговое выражение для фильтрации состоит из двух условий: во-первых, фильтр не пропускает координаты с моментальной скоростью больше настроенного параметра, во-вторых, оценивает достоверность расстояния между двумя смежными точками в зависимости от коэффициента:

$$F_4(\{(t_i, lat_i, lon_i, spd_i), (t_{i-1}, lat_{i-1}, lon_{i-1})\}, P_{maxSpeed}^4, P_{maxTime}^4, P_{distK}^4) = \begin{cases} 0, & \left[\begin{array}{l} spd_i > P_{maxSpeed}^4 \\ t_i - t_{i-1} \leq P_{maxTime}^4 \text{ и } \frac{D(lat_i, lon_i, lat_{i-1}, lon_{i-1})}{P_{maxSpeed}^4 * \frac{(t_i - t_{i-1})}{1000}} > P_{distK}^4 \end{array} \right. \\ 1, & \text{иначе} \end{cases} \quad (2.11)$$

где $P_{maxSpeed}^4 \in \mathbb{R} > 0$ – параметр модели, максимальная моментальная скорость в метрах в секунду;

$P_{maxTime}^4 \in \mathbb{N} > 0$ – параметр модели, максимальный интервал времени между координатами для анализа скорости в миллисекундах;

$D(lat_i, lon_i, lat_{i-1}, lon_{i-1}) \in \mathbb{R} \geq 0$ – функция расстояния в метрах между соответствующими координатами широты и долготы; расстояние в метрах по геодезическим координатам целесообразно рассчитывать на основе WGS84 [49];

$P_{distK}^4 \in \mathbb{R} \geq 1$ – параметр модели, безразмерный коэффициент анализа реального и максимально возможного расстояния; при $P_{distK}^4 = 1$, фильтр отбросит координаты, если пройденная дистанция больше максимально возможной.

Рассмотрим распределение отношения реального расстояния между координатами к максимально возможной дистанции для точек, разница времени которых меньше $P_{maxTime}^4 = 30\,000\text{мс}$ при $P_{maxSpeed}^4 = \frac{120\text{ м}}{3.6\text{ с}}$ (см. Рисунок 2.14).

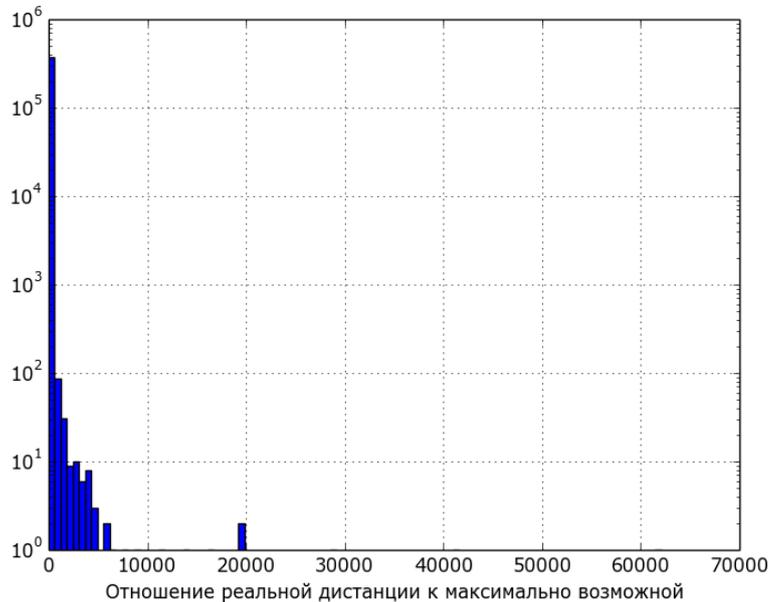


Рисунок 2.14. Распределение отношения реальной дистанции к максимально возможной

Максимальное значение отношения – 62068.13, это означает, что пользователь за временной отрезок меньше либо равный 30 с между получением двух координат преодолел дистанцию в 62068.13 раз больше, чем возможно при равномерном движении со скоростью $120 \frac{\text{км}}{\text{ч}}$. Очевидно, что вторая координата в последовательности имеет очень высокую погрешность. Среднее значение здесь $\mu = 2.23$, $\sigma = 152.645$, с учетом допуска в 3σ значения распределены на отрезке $[0; 460.166]$. Гистограмма с увеличенным масштабом по оси x представлена на Рисунке 2.15.

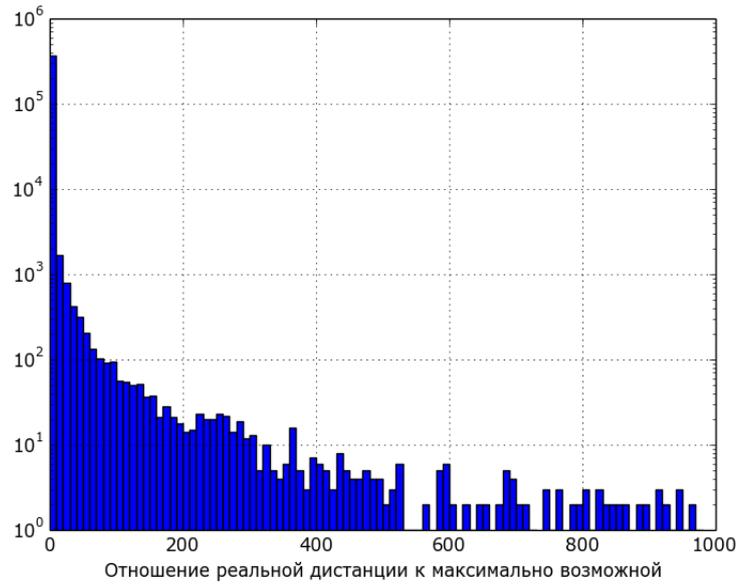


Рисунок 2.15. Распределение отношения реальной дистанции к максимально возможной

Для выборки, где рассмотренное отношение больше либо равно единице среднее значение составляет 48.880, $\sigma = 721.377$, отрезок в 3σ – $[0; 2213.013]$.

На Рисунках 2.16 и 2.17 представлена зависимость распределения точности от отношения реальной дистанции к максимально возможной.

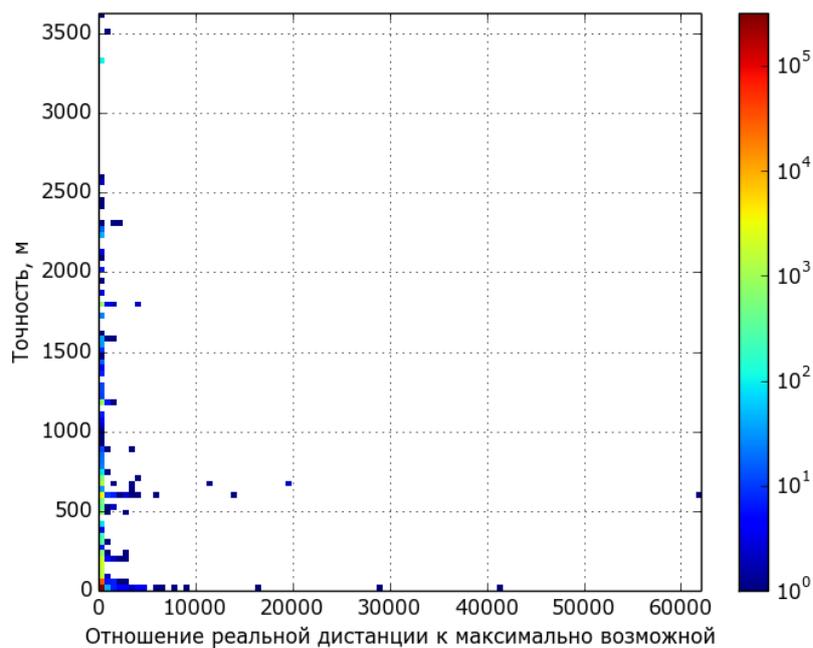


Рисунок 2.16. Зависимость распределения точности от отношения реальной дистанции к максимально возможной (1)

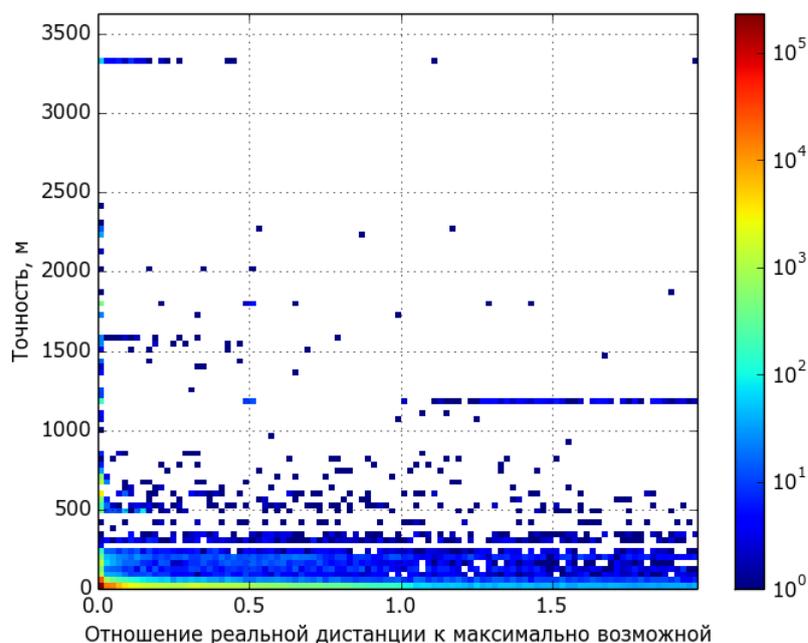


Рисунок 2.17. Зависимость распределения точности от отношения реальной дистанции к максимально возможной (2)

Ковариация величин равна 626.882, коэффициент корреляции составляет 0.027, что означает увеличение параметра точности при увеличении рассмотренного отношения реальной дистанции к максимально возможной. Малый коэффициент корреляции подтверждает, что параметр точности для координат с высоким значением отношения реальной дистанции к максимально возможной абсолютно недостоверен. Предложенный метод позволяет выявлять и правильно отфильтровывать случаи, когда значение точности координаты мало, однако по экспертному анализу трека пользователя очевидно, что точка определена с очень большой погрешностью.

Поскольку рассмотренный подход позволяет неявно определить погрешность координаты не используя параметр точности, на данном этапе возможно модифицировать точность координаты в зависимости от отношения пройденной дистанции к максимально возможной и коэффициента P_{distK}^4 , что увеличит достоверность трека и существенно упростит дальнейший анализ координат. Такой метод заслуживает отдельного исследования и в данной работе рассматриваться не будет.

Принимая коэффициент P_{distK}^4 равным единице, фильтр позволит отбросить порядка 4.458 % координат с большой погрешностью, при этом среднее значение точности в выборке уменьшится на 6.165 м (16.4 %), среднеквадратичное отклонение – на 22.905 м (15 %). Распределение точности до и после фильтрации представлено на Рисунке 2.18.

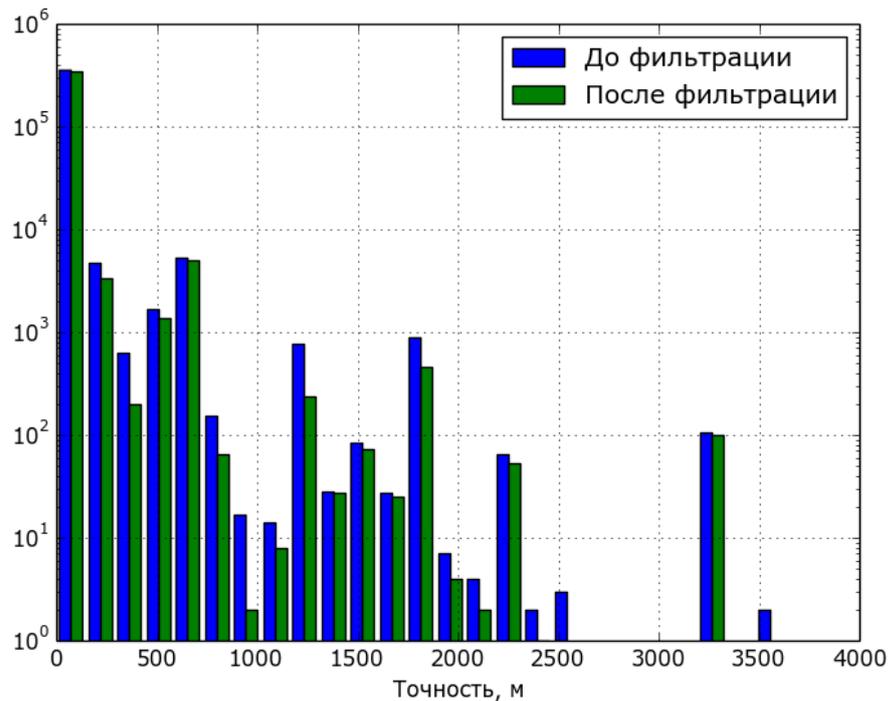


Рисунок 2.18. Распределение точности до и после обработки фильтром по скорости

2.2.5. Фильтр по ускорению

По интервалу времени между координатами и скорости пользователя в двух смежных точках возможно рассчитать моментальное ускорение пользователя, с учетом того, что движение между точками равноускоренное. Цель фильтрации – исключение координат с ускорением, больше максимально возможного. Распределение ускорения представлено на Рисунке 2.19.

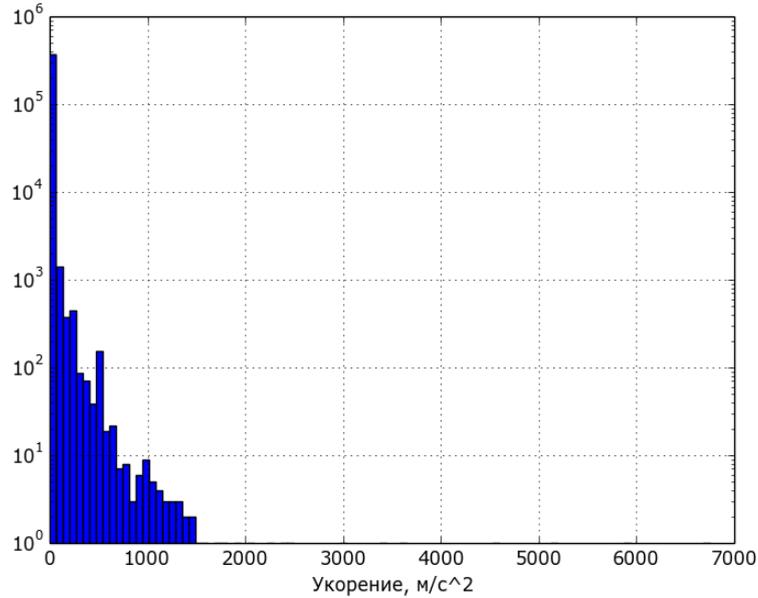


Рисунок 2.19. Распределение точности до и после обработки фильтром по скорости

Ускорение в общем случае нормально распределено с учетом 3σ на отрезке $[0 \frac{\text{м}}{\text{с}^2}; 99.817 \frac{\text{м}}{\text{с}^2}]$, среднее значение $\mu = 2.294 \frac{\text{м}}{\text{с}^2}$, среднеквадратичное отклонение $\sigma = 32.508 \frac{\text{м}}{\text{с}^2}$. Ускорение мало коррелируется с параметром точности (см. Рисунок 2.20): ковариация здесь равна 166.796, коэффициент корреляции 0.022.

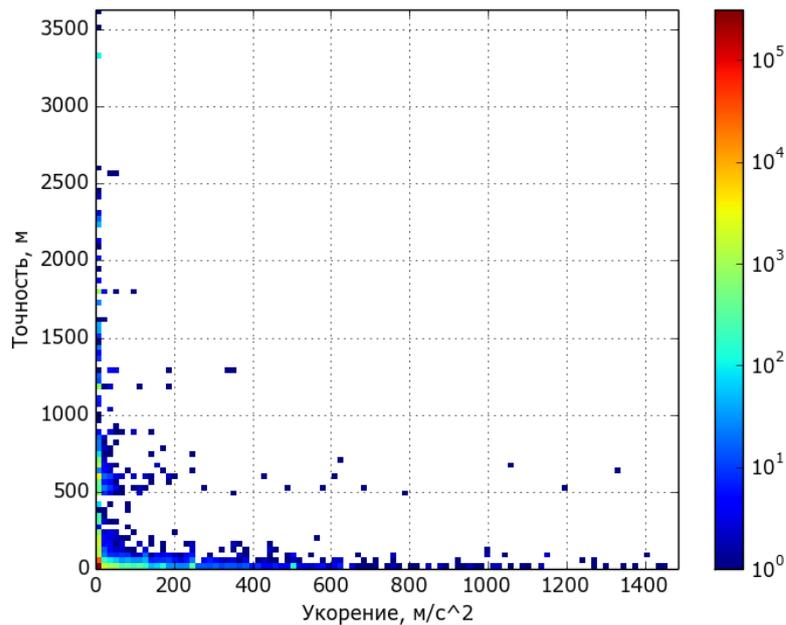


Рисунок 2.20. Распределение точности до и после обработки фильтром по скорости

Большие значения ускорений в распределении могут означать большую погрешность определения скорости и, как следствие, высокую недостоверность параметра точности. Установив порог максимального ускорения и анализируя моментальное ускорение по двум точкам, возможно отфильтровывать координаты, в предположении, что при большом ускорении очередная координата, как и ее параметр точности, определена с большой погрешностью. Предложенный метод аналогичен работе предыдущего фильтра. Здесь также возможно модифицировать параметр точности, анализируя ускорение, что увеличит достоверность параметра и упростит работу с потоком координат далее.

Итоговое выражение рассчитывает равномерное ускорение между двумя точками как отношение модуля скоростей к разнице времени и сравнивает его с предельным значением.

$$F_5(\{\langle t_i, spd_i \rangle, \langle t_{i-1}, spd_{i-1} \rangle\}, P_{maxAccel}^5) = \begin{cases} 0, & \frac{|spd_i - spd_{i-1}|}{\frac{t_i - t_{i-1}}{1000}} > P_{maxAccel}^5 \\ 1, & \text{иначе} \end{cases}, \quad (2.12)$$

где $P_{maxAccel}^5 \in \mathbb{R} > 0$ – параметр модели, максимальное ускорение в метрах в секунду в квадрате.

Поскольку существует слабая корреляция параметров ускорения и точности, распределение точности после работы фильтра изменится незначительно. При $P_{maxAccel}^5 = \frac{9.81}{2} \frac{м}{с^2}$ фильтр отбросит 3.23% координат, среднее значение точности изменится в большую сторону на 0.066 м (0.174 %), среднеквадратичное отклонение – на 0.389 м (0.253 %). Распределение точности представлено на Рисунке 2.21.

2.2.6. Фильтр стоянок

Данный фильтр является одним из основных блоков для достоверного определения стоянок пользователя. Выражение фильтра всегда возвращает единицу, однако метод фильтрации при определенных условиях может менять широту и долготу текущей координаты.

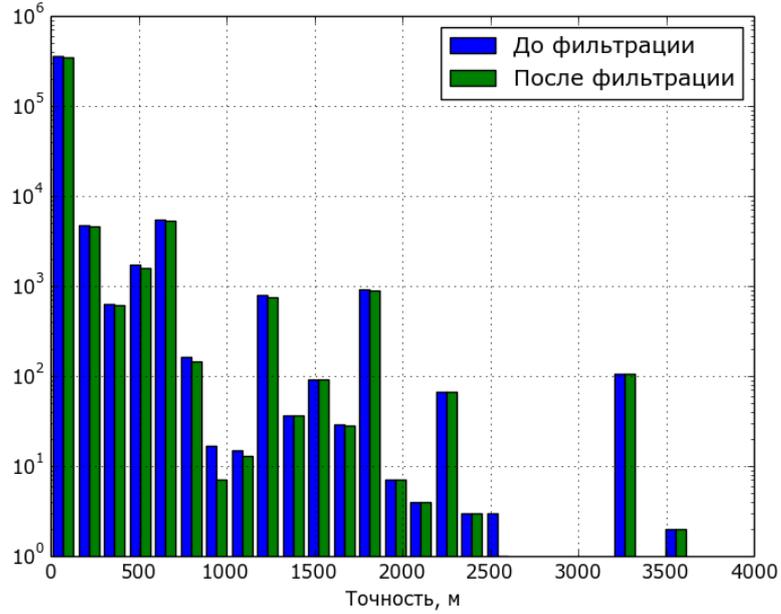


Рисунок 2.21. Распределение точности до и после обработки фильтром по ускорению

Определение стоянок состоит из вычисления двух параметров: s_i^{accel} и s_i^{speed} . Рассмотрим первый:

$$s_i^{accel} = \begin{cases} \sqrt{\left(\sum_{j=1-k+1}^i \left| \frac{\sum_{j=1-k+1}^i ax_i}{k} - ax_i \right| \right)^2 + \left(\sum_{j=1-k+1}^i \left| \frac{\sum_{j=1-k+1}^i ay_i}{k} - ay_i \right| \right)^2 + \left(\sum_{j=1-k+1}^i \left| \frac{\sum_{j=1-k+1}^i az_i}{k} - az_i \right| \right)^2} & , \\ 1, \frac{\sqrt{\left(\sum_{j=1-k+1}^i \left| \frac{\sum_{j=1-k+1}^i ax_i}{k} - ax_i \right| \right)^2 + \left(\sum_{j=1-k+1}^i \left| \frac{\sum_{j=1-k+1}^i ay_i}{k} - ay_i \right| \right)^2 + \left(\sum_{j=1-k+1}^i \left| \frac{\sum_{j=1-k+1}^i az_i}{k} - az_i \right| \right)^2}}{k} < P_{accelCut}^6 \\ 0, \text{ иначе} \end{cases} \quad (2.13)$$

где ax_i, ay_i, az_i – ускорения в метрах в секунду в квадрате по соответствующим осям, полученные от акселерометра мобильного устройства;

k – количество кортежей ускорений $\langle ax_i, ay_i, az_i \rangle$, полученное за период времени $[t_i - P_{periodAccel}^6; t_i]$;

$P_{periodAccel}^6 \in \mathbb{N} > 0$ – параметр модели, определяющий интервал времени в миллисекундах съема значений ускорения устройства;

$P_{accelCut}^6 \in \mathbb{R} > 0$ – верхний порог дисперсии ускорения в метрах в квадрате в секунду в четвертой степени, до которого фильтр считает текущую точку стоянкой.

Формула в первом уравнении системы есть ничто иное как расчет среднего значения дисперсии ускорений по трем осям за k предыдущих значений. Если дисперсия больше порога $P_{accelCut}^6$, текущая координата – стоянка, иначе – движение.

Если на основе акселерометра не удалось определить стоянку, или данные отсутствуют, вычисляется параметр:

$$s_i^{speed} = \begin{cases} 1, & \sqrt{\left(\frac{\sum_{j=1}^{i-n+1} lat_j * \sum_{j=1}^n (t_j - t_1) - n * \sum_{j=1}^n lat_j * (t_j - t_1)}{\sum_{j=1}^n (t_j - t_1)^2 - n * \sum_{j=1}^n (t_j - t_1)^2} * f_{latToMet}(lat_i) \right)^2 + \left(\frac{\sum_{j=1}^n lon_j * \sum_{j=1}^n (t_j - t_1) - n * \sum_{j=1}^n lon_j * (t_j - t_1)}{(\sum_{j=1}^n (t_j - t_1))^2 - n * \sum_{j=1}^n (t_j - t_1)^2} * f_{lonToMet}(lon_i) \right)^2} < P_{speedCut}^6, (2.14) \\ 0, & \text{иначе} \end{cases}$$

где n – количество координат, полученное за период времени $[t_i - P_{periodCoord}^6; t_i]$;

$P_{periodCoord}^6 \in \mathbb{N} > 0$ – параметр модели, определяющий интервал времени в миллисекундах съема значений скорости устройства;

$f_{latToMet}(lat_i) = 111132.92 - 559.82 * \cos\left(2 * \frac{lat_i}{180} * \pi\right) + 1.175 * \cos\left(4 * \frac{lat_i}{180} * \pi\right) - 0.0023 * \cos\left(6 * \frac{lat_i}{180} * \pi\right)$ – функция перевода широты в метры;

$f_{lonToMet}(lon_i) = 111412.84 * \cos\left(\frac{lon_i}{180} * \pi\right) - 93.5 * \cos\left(3 * \frac{lon_i}{180} * \pi\right) + 0.118 * \cos\left(4 * \frac{lon_i}{180} * \pi\right)$ – функция перевода долготы в метры;

$P_{speedCut}^6 \in \mathbb{R} > 0$ – верхний порог дисперсии скорости в метрах в квадрате в секунду в четвертой степени, до которого фильтр считает текущую точку стоянкой.

Формула в первом уравнении системы, аналогичная (2.13), является расчетом среднего значения дисперсии скорости устройства по двум осям.

Итоговый признак стоянки рассчитывается всегда при поступлении очередной координаты в фильтр как:

$$Q_{isStand} = s_i^{accel} \vee s_i^{speed}. (2.15)$$

Если $Q_{isstand} = 1$, следует модифицировать широту lat_i , долготу lon_i и признак стоянки s_i для текущей координаты в предположении, что она является продолжением стоянки. Итоговое выражение фильтра:

$$F_6 \left(\begin{array}{l} \{ \langle t_i, lat_i, lon_i \rangle, \dots, \langle t_{i-n+1}, lat_{i-n+1}, lon_{i-n+1} \rangle \}, \\ \{ \langle t_i, ax_i, ay_i, az_i \rangle, \dots, \langle t_{i-k+1}, ax_{i-k+1}, ay_{i-k+1}, az_{i-k+1} \rangle \}, \\ P_{accelCut}^6, P_{speedCut}^6, P_{periodCoord}^6, P_{periodAccel}^6 \end{array} \right) = 1. \quad (2.16)$$

Рассмотрим работу фильтра в случае продолжительных стоянок. На Рисунках 2.22 и 2.23 представлены стоянки пользователей, продолжительностью 3 дня и 4 часа соответственно. Данные акселерометра здесь не использовались. Красным цветом показаны точки, у которых параметр скорости равен нулю, синим – точки в движении.

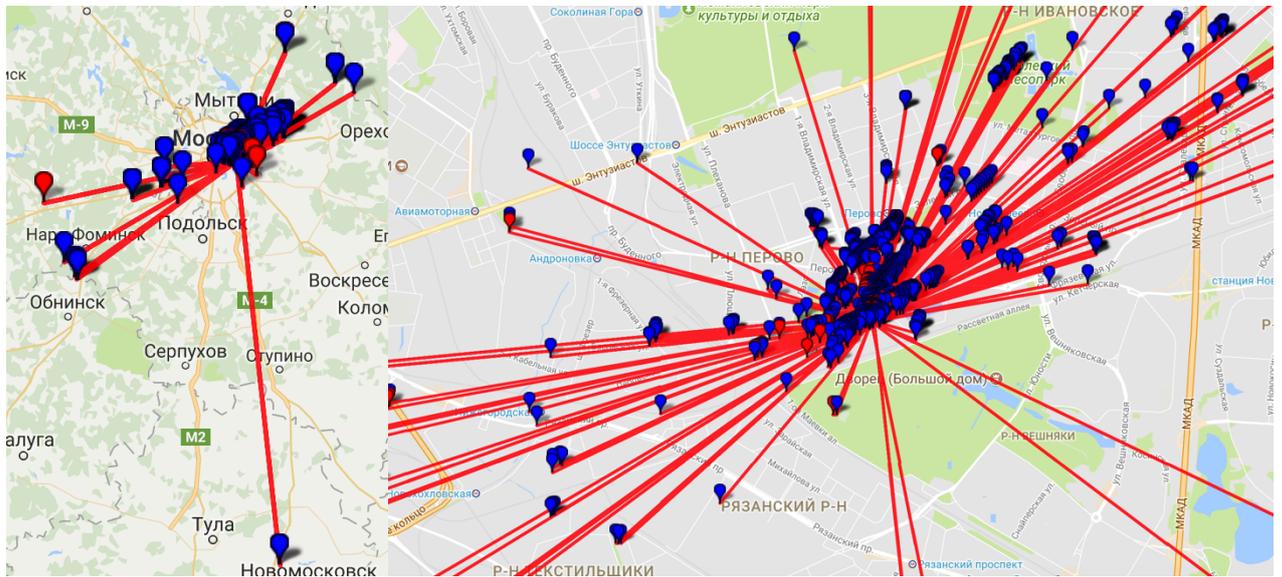


Рисунок 2.22. Трехдневная стоянка пользователя

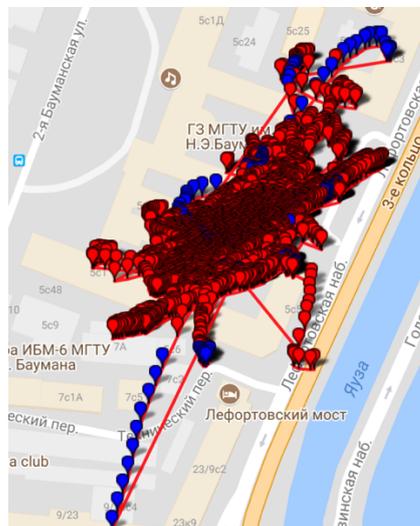


Рисунок 2.23. Стоянка пользователя, продолжительностью 4 часа

В первом случае можно заметить сильные выбросы координат. На Рисунке 2.24 показаны треки после фильтрации с параметрами $P_{periodCoord}^6 = 10 * 60 * 1000$ мс, $P_{speedCut}^6 = 1 \left(\frac{м}{с^2}\right)^2$.

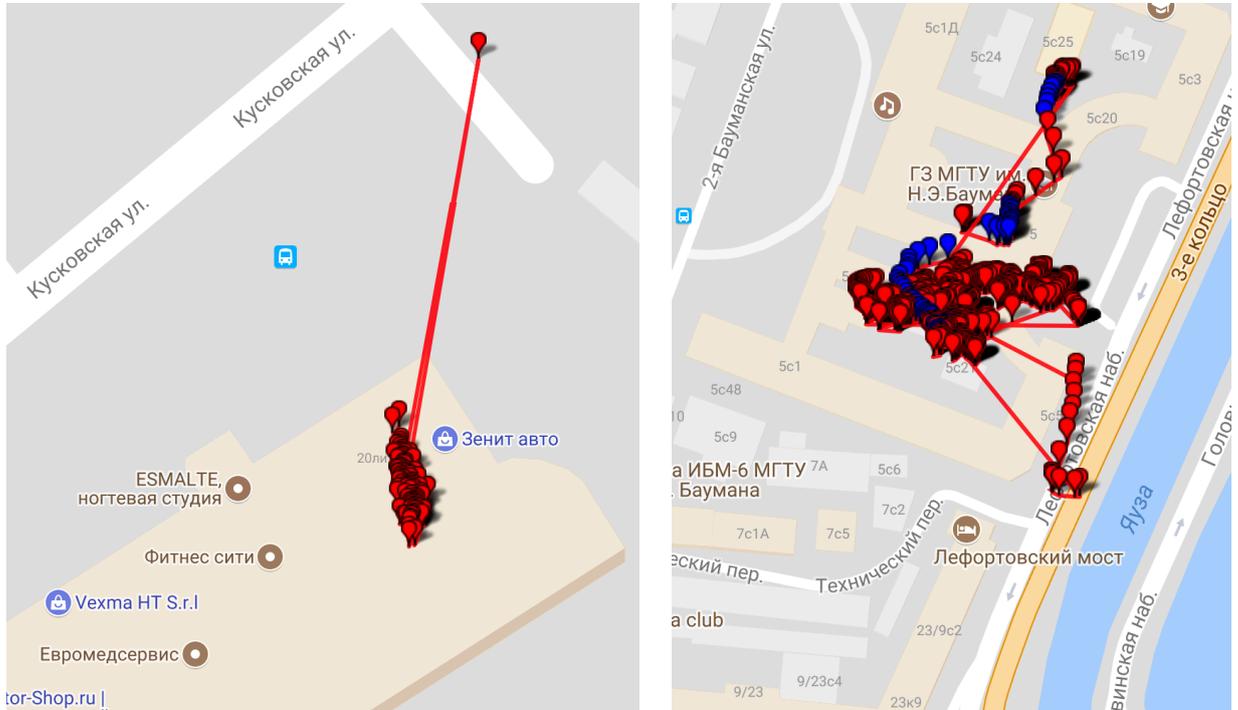


Рисунок 2.24. Треки после обработки фильтром стоянок

На Рисунке 2.25 представлена гистограмма распределения скорости до и после фильтрации.

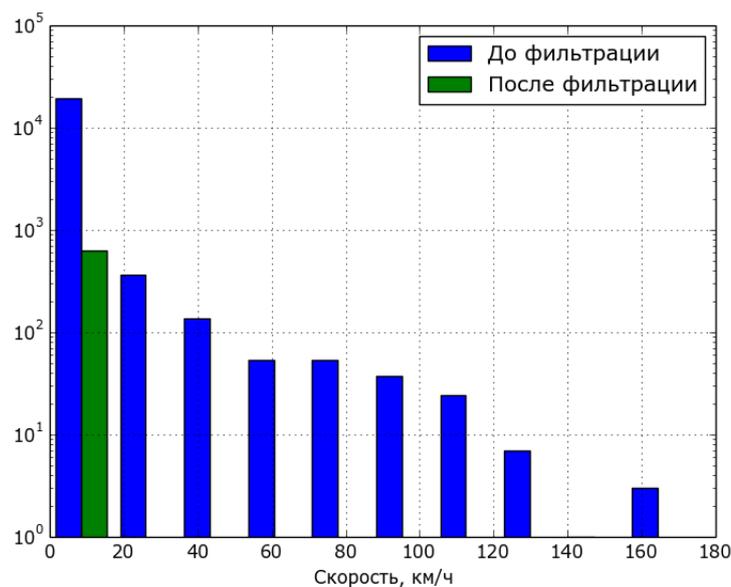


Рисунок 2.25. Распределение скорости до и после работы фильтра стоя-

Исходя из экспертного анализа результирующих треков и гистограммы выше, можно утверждать, что фильтр стоянок отбросил лишь часть координат, однако существенно увеличил качество трека пользователя, что особенно заметно для первой стоянки. Процент отфильтрованных координат здесь составляет 98.872, координат с нулевой скоростью – 99.006, координат с положительной скоростью – 96.085. При этом итоговая пройденная дистанция с 13763.687 м уменьшилась до 3545.094 м (74.24 %), что является существенным показателем эффективности фильтрации.

Неявной задачей фильтра также является анализ девиаций координат, что хорошо видно на Рисунке 2.23 слева внизу. Такие случаи часто происходят в местах стоянок, они трудно выявляются, поскольку параметр точности таких координат довольно высокий, однако только на этапе экспертного анализа совершенно очевидно, что он рассчитан с большой погрешностью. Определение дисперсии ускорения и скорости по входным данным за фиксированный промежуток времени позволяет определять такие артефакты трека.

2.2.7. Сглаживающий фильтр

Поскольку в каждой входной координате присутствуют параметры широты и долготы на данном этапе следует применить механизм сглаживания, что может существенно увеличить качества результирующего трека в движении, когда возможны единичные выбросы координат. В этом случае после сглаживания координаты будут располагаться намного ближе к реальному треку.

В качестве метода фильтрации целесообразно использовать экспоненциальное скользящее среднее. Практика показала, что использование данного метода достаточно для формирования гладкой траектории движения пользователя. Преимущество алгоритма – низкая вычислительная и емкостная сложность, и как следствие малое время выполнения, что особенно важно для программного обеспечения на мобильных устройствах.

Фильтр никогда не отбрасывает координату, всегда возвращая единицу, итоговое выражение фильтрации:

$$F_7(\{\langle lat_i, lon_i \rangle, \langle lat_{i-1}, lon_{i-1} \rangle\}, P_{smoothBase}^7, P_{maxSmoothDistance}^7) = 1, \quad (2.17)$$

где $P_{smoothBase}^7 \in \mathbb{R} \in [0.0; 1.0]$ – параметр модели, сглаживающая безразмерная константа, чем меньше значение, тем больше влияние широты и долготы предыдущей точки на широту и долготу текущей;

$P_{maxSmoothDistance}^7 \in \mathbb{R} > 0$ – параметр модели, максимальное расстояние между точками в метрах, при котором применяется сглаживание. В случае, когда точки находятся далеко друг от друга, сглаживать трек нецелесообразно.

Если $D(lat_i, lon_i, lat_{i-1}, lon_{i-1}) < P_{maxSmoothDistance}^7$, фильтр вычисляет новую широту и долготу i координаты по следующим формулам:

$$\begin{aligned} lat_i &= lat_i * P_{smoothBase}^7 + lat_{i-1} * (1 - P_{smoothBase}^7); \\ lon_i &= lon_i * P_{smoothBase}^7 + lon_{i-1} * (1 - P_{smoothBase}^7), \end{aligned} \quad (2.18)$$

Для оценки влияния сглаживающего фильтра на последовательность координат рассмотрим распределение угла α в градусах до и после сглаживания между двумя прямыми, построенными по двум парам из каждой тройки координат в исходной выборке (см. Рисунок 2.26), при значении $P_{smoothBase}^7 = 0.9$ (см. Рисунок 2.27). Угол в рассмотренном случае вычисляется по меньшей разнице против или по часовой стрелке, он не может быть больше 180° .

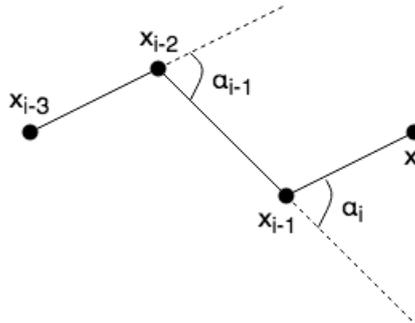


Рисунок 2.26. Угол α между направлениями движения

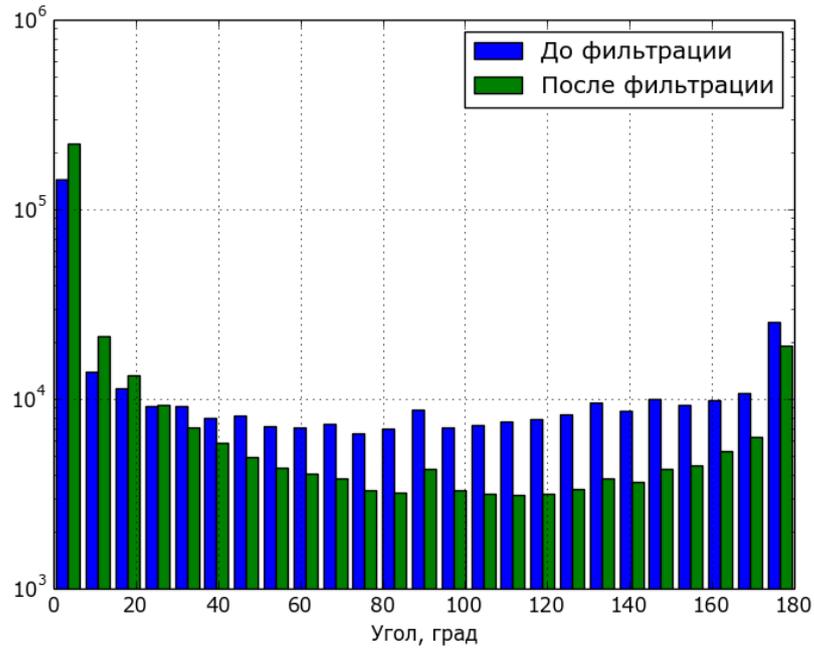


Рисунок 2.27. Распределение угла α до и после обработки сглаживающим фильтром

Среднее значение угла до и после сглаживания соответственно: 61.131° и 34.359° ; среднеквадратичное отклонение: 65.159° и 56.97° . Среднее значение уменьшилось на 26.772° (43.794 %), сигма на 8.189° (12.568 %). Множество значений в распределении, которые соответствуют большим углам – от 90° и более, есть ни что иное как выбросы в местах стоянок пользователя. Данные координаты успешно определяется и отбрасываются ранее предложенным фильтром стоянок и диагностическим фильтром, который будет рассмотрен далее.

2.2.8. Угловой фильтр

В предпоследнем блоке обработки целесообразно уменьшить плотности точек в треке координат, основанное на анализе угла между направлениями движения по трем координатам аналогично вычислению угла α выше. Направление движения рассчитывается по двум координатам: первой и второй, второй и третьей в последовательности.

Фильтр по углу в первую очередь сравнивает интервал времени между предыдущей и текущей координатами с максимальным интервалом времени, который рассчитывается в зависимости от угла α :

$$t_i - t_{i-1} < \Delta T_{max}(\alpha), \quad (2.19)$$

где $\alpha = |B(lat_i, lon_i, lat_{i-1}, lon_{i-1}) - B(lat_{i-1}, lon_{i-1}, lat_{i-2}, lon_{i-2})|$ – угол в градусах, если $\alpha > 180^\circ$, то α принимается равным $360^\circ - \alpha$;

$B(lat_i, lon_i, lat_{i-1}, lon_{i-1})$ – функция угла направления в градусах по соответствующим координатам;

ΔT_{max} – функция угла α , вычисляющая максимальный интервал времени в миллисекундах между текущей и предыдущей координатой в зависимости от угла α :

$$\Delta T_{max}(\alpha) = \begin{cases} 1000, & \alpha > P_{maxAngle}^8 \\ \frac{1000 - P_{maxTimeDiff}^8}{P_{maxAngle}^8} * \alpha + P_{maxTimeDiff}^8, & \alpha \leq P_{maxAngle}^8 \end{cases}, \quad (2.20)$$

где $P_{maxAngle}^8 \in \mathbb{R} \in [0; 180]$ – параметр модели, максимальный угол в градусах, при котором координата будет отфильтрована, если интервал времени меньше 1 с;

$P_{maxTimeDiff}^8 \in \mathbb{R} > 0$ вместе с параметром $P_{maxAngle}^8$, $^\circ$ задают максимальный интервал времени в миллисекундах, за который координаты будут отфильтрованы при угле $\alpha = 0^\circ$, и максимальный угол, в пределах значения которого координаты будут отфильтрованы, если интервал времени будут соответствовать рассчитанному диапазону. Иными словами, порог интервала времени динамически зависит от угла между направлениями движения. Если $\alpha \geq P_{maxAngle}^8$ и интервал времени меньше 1 с, координаты отфильтровываются, в предположении, что пользователь совершает поворот, и следует пропустить максимум координат для формирования гладкого результирующего трека. Если $\alpha < P_{maxAngle}^8$, то поворот либо не значительный, либо пользователь движется по прямой, и в этом случае целесообразно отфильтровывать большее число координат, так как для определения прямой линии необходимо и достаточно всего две точки,

остальные точки в треке будут неинформативны. Графическая интерпретация функции (2.20) представлена на Рисунке 2.28.

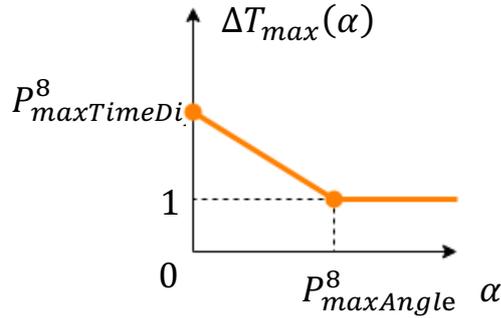


Рисунок 2.28. График функции $\Delta T_{max}(\alpha)$

Параметры $P_{maxTimeDiff}^8$ и $P_{maxAngle}^8$ фактически задают уравнения наклонной прямой. Однако, расширив параметры и подобрав значения коэффициентов, между точками $(0, P_{maxTimeDiff}^8)$ и $(P_{maxAngle}^8, 1)$ возможно провести более гладкую кривую, что может увеличить качество разрежения точек. Данный подход в работе не рассматривается.

Также возможно анализировать значение угла в зависимости от средней скорости пользователя в 3 точках:

$$\bar{\vartheta} = \frac{spd_i + spd_{i-1} + spd_{i-2}}{3}. \quad (2.21)$$

Если средняя скорость больше определенного порога, порог угла следует увеличить, так как с увеличением скорости максимально возможный угол поворота за определенное время снижается. Верно и обратное: чем ниже скорость, тем поворот на больший угол можно совершить за определенное время. В соответствии с этим предположением во втором этапе фильтрации будем анализировать угол α в зависимости от средней скорости пользователя:

$$t_i - t_{i-1} < P_{maxTimeDiff}^8 * 3 \text{ и } \alpha > \Delta A_{min}(\bar{\vartheta}), \quad (2.22)$$

где параметр $P_{maxTimeDiff}^8 > 0$, умноженный на 3, определяет интервал времени в миллисекундах, за который будет производиться анализ угла и средней скорости;

$\Delta A_{min}(\bar{\vartheta})$ – функция от средней скорости $\bar{\vartheta}$, вычисляющая минимальный угол в градусах, при достижении которого координата будет отфильтрована, по

формуле:

$$\Delta A_{min}(\bar{\vartheta}) = \begin{cases} P_{maxAngle}^8, \bar{\vartheta} > P_{maxSpeed}^8 \\ \frac{P_{maxAngle}^8 - 180}{P_{maxSpeed}^8} * \bar{\vartheta} + 180 \end{cases} \quad (2.23)$$

где $P_{maxSpeed}^8 > 0$ – параметр модели, максимальная скорость в метрах в секунду, при котором координата будет отфильтрована, если $\alpha > P_{maxAngle}^8$.

Система уравнений аналогична (2.20). С увеличением средней скорости фильтр в соответствии с обратной зависимостью динамически снижает максимальный порог, с уменьшением – увеличивает. Здесь также возможно использовать более гладкую кривую.

С учетом (2.20) и (2.22) итоговое выражение фильтра:

$$F_8 \left(\left\{ \langle t_i, lat_i, lon_i, spd_i \rangle, \langle t_{i-1}, lat_{i-1}, lon_{i-1}, spd_{i-1} \rangle, \langle lat_{i-2}, lon_{i-2}, spd_{i-2} \rangle \right\}, \right. \\ \left. P_{maxTimeDiff}^8, P_{maxAngle}^8, P_{maxSpeed}^8 \right) = \\ = \begin{cases} 0, & \left[\begin{array}{l} t_i - t_{i-1} < \Delta T_{max}(\alpha) \\ t_i - t_{i-1} < P_{maxTimeDiff}^8 * 3 \text{ и } \alpha > \Delta A_{min}(\bar{\vartheta}) \end{array} \right] \\ 1, \text{ иначе} \end{cases} \quad (2.24)$$

Рассмотрим распределение угла α до и после работы угловым фильтром на исходной выборке (см. Рисунок 2.29) при значениях параметров:

$$P_{maxTimeDiff}^8 = 15\,000 \text{ мс}, P_{maxAngle}^8 = 10^\circ, P_{maxSpeed}^8 = 25 \frac{\text{м}}{\text{с}}.$$

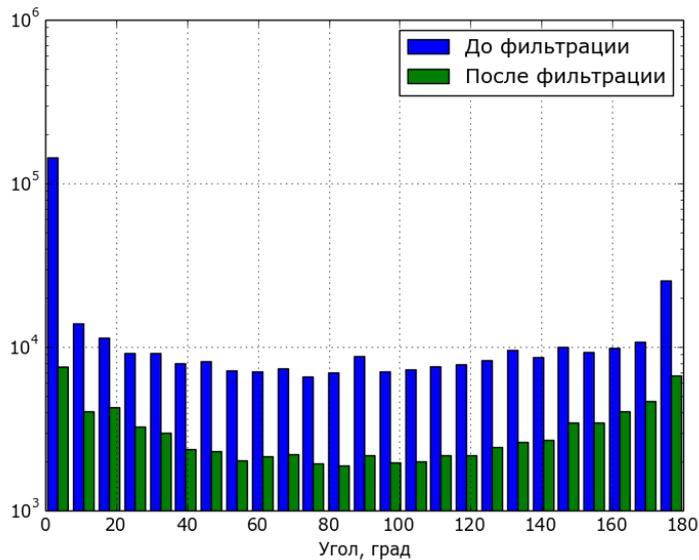


Рисунок 2.29. Распределение угла α до и после обработки угловым фильтром

Среднее значение угла до и после фильтрации составляет 61.131° и 90.411° соответственно, среднеквадратичное отклонение – 61.489° и 65.159° . Фильтр сместил среднее значение угла в центр распределения, однако дисперсия поменялась незначительно. Количество координат после фильтрации уменьшилось на 78.976 %.

2.2.9. Диагностический фильтр

Последним в каскаде целесообразно разместить диагностический фильтр. Предлагаемый метод адаптивной диагностической фильтрации основан на подходе [1, 2]. Напомним, что в соответствии с [1], критерий фильтра:

$$D_{1-2} > Dl_1 + Dl_2, \quad (2.25)$$

где D_{1-2} – расстояние в метрах между точками 1 и 2;

Dl_1 и Dl_2 , м – характеристики точек, прямо пропорциональные точности снятия координат.

Если неравенство выполняется, точка 2 – считается валидной и проходит фильтрацию, если нет – точка отбрасывается.

Цель фильтрации – исключение отклонений координат в местах стоянок. Идея адаптивной фильтрации заключается в 2 основных гипотезах:

- 1) уменьшая пороговые значения Dl_1 и Dl_2 при «входе» и «выходе» из стоянки, где $s_i = 1$ и $s_{i-1} = 1$ возможно сохранить координаты движения близ стоянок, что увеличит качество результирующего трека;
- 2) увеличивая пороговые значения Dl_1 и Dl_2 прямо пропорционально точности и обратно-пропорционально скорости пользователя, возможно уменьшить количество выбросов в местах стоянок, так как скорость в этом случае существенно ниже, чем в движении;

На основе рассмотренных гипотез и предложенной модели входных данных $Dl_1 + Dl_2$ или $Dist$ для очередной координаты i :

$$Dist(\{\langle acc_i, spd_i, tp_i, s_i \rangle, \langle acc_{i-1}, spd_{i-1}, tp_{i-1}, s_{i-1} \rangle\}) =$$

$$\begin{cases} f_{acc}(acc_{i-1}, tp_{i-1}) + f_{acc}(acc_i, tp_i), \text{ если } s_i = 1 \text{ и } spd_{i-1} > P_S^9 \text{ или } s_{i-1} = 1 \text{ и } spd_i > P_S^9 \\ f_{acc}(acc_{i-1}, tp_{i-1}) * f_K(spd_{i-1}) + f_{acc}(acc_i, tp_i) * f_K(spd_i), \text{ иначе} \end{cases}, (2.26)$$

где $f_{acc}(acc_i, tp_i) = \begin{cases} \frac{acc_i}{P_K^9}, tp_i = 2 \\ acc_i, \text{ иначе} \end{cases}$ – функция, учитывающая низкие точности

координат от Wi-Fi и вышек сотовой связи;

$P_S^9 \in \mathbb{R} \geq 0$ – параметр диагностического адаптивного фильтра, определяющий минимальную скорость в метрах в секунду, которая предположительно является скоростью пользователя после выхода из стоянки;

$f_K(spd_i)$ – функция, значением которой является безразмерная величина, зависящая от скорости spd_i .

Первое уравнение в системе соответствует первой гипотезе выше, значение $Dist$ здесь прямо пропорционально точности двух смежных точек без добавления какого-либо коэффициента множителя. Фактически это выражения является критерием выхода из стоянки. Второе уравнение отражает вторую гипотезу: параметры точности в каждой из точек домножаются на соответствующий коэффициент, который зависит от скорости пользователя в этих точках.

Введем новый безразмерный параметр $P_K^9 \in \mathbb{R} > 1$ и рассмотрим необходимые ограничения функции $f_K(spd)$:

- $spd \geq 0$ – скорость всегда неотрицательна;
- $f_K(spd) \in [1; P_K^9]$ – функция должна добавлять небесконечный вклад в $Dist$;
- $f_K(spd) = P_K^9$, при $spd = 0$ – значение функции при нулевой скорости. Коэффициент P_K^9 фактически определяет порог фильтра в случае стоянки;
- $f_K(spd) = 1$, при $spd \geq P_S^9$ – значение функции при скорости, большей или равной P_S^9 . Коэффициент P_S^9 определяет максимальную скорость, при которой фильтр вырождается в предложенный в [2];
- $\lim_{spd \rightarrow 0+0} f_K(spd) = P_K^9$ – непрерывность функции справа в точке $(0; P_K^9)$;
- $\lim_{spd \rightarrow b-0} f_K(spd) = 1$ – непрерывность функции слева в точке $(P_S^9; 1)$;

- $f_K(spд)$ – непрерывная убывающая функция: чем больше скорость, тем меньше выбросов координат, и меньшим должен быть порог фильтра.

Далее в соответствии с ограничениями необходимо доопределить функцию $f_K(spд)$ на отрезке $[0; P_S^9]$. Были рассмотрены несколько типов функций: экспоненты, степенные функции, линейные функции. Можно предположить, что для увеличения точности фильтрации $f_K(spд)$ должна быть гладкой на всей области определения, однако, как показал анализ экспериментальных результатов, гладкость функции не играет практически никакой роли, итоговый трек качественно не меняется, более того, при использовании сложных функций в общем треке возникают различные артефакты, которых очевидно быть не должно.

Выбор функции $f_K(spд)$, удовлетворяющей всем условиям, представляет собой нетривиальную задачу. Можно отметить, что из-за различий в типах GPS/GLONASS приемников, последнее ограничение в списке может быть как-либо модифицировано в зависимости от конкретных частотных параметров координат. Для этого в общем случае необходимо выполнить частотный анализ потока геоданных различных типов приемников для нахождения зависимостей параметров координат от частоты выбросов, что в данной работе рассматриваться не будет.

Исходя из минимизации вычислительных затрат была выбрана линейная функция:

$$f_K(spд) = \begin{cases} 1, & spд > P_S^9 \\ \frac{1-P_K^9}{P_S^9} * spд + P_K^9, & 0 \leq spд \leq P_S^9 \end{cases} \quad (2.27)$$

график которой приведен на Рисунке 2.30.

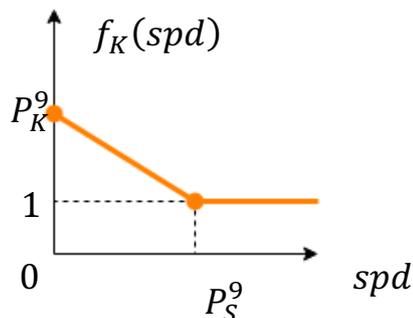


Рисунок 2.30. График функции $f_K(spд)$

Итоговая функция фильтрации:

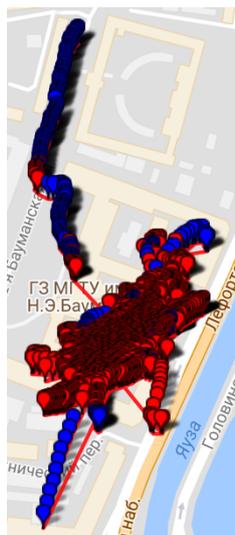
$$F_9 \left(\left\{ \langle lat_i, lon_i, acc_i, spd_i, tp_i, s_i \rangle, \langle lat_{i-1}, lon_{i-1}, acc_{i-1}, spd_{i-1}, tp_{i-1}, s_{i-1} \rangle \right\}, P_K^9, P_S^9 \right) = \begin{cases} 0, & Dist(\langle acc_i, spd_i, tp_i, s_i \rangle, \langle acc_{i-1}, spd_{i-1}, tp_{i-1}, s_{i-1} \rangle) \leq D(lat_i, lon_i, lat_{i-1}, lon_{i-1}), \\ & \text{и } s_{i-1} = 0, \text{ и } s_i = 0 \\ 1, & \text{иначе} \end{cases} \quad (2.28)$$

Если условие $Dist \leq D$ выполняется, и предыдущая координата – стоянка, можно с достаточной степенью уверенности утверждать, что текущая координата является продолжением стоянки. В этом случае метод пропускает координату и модифицирует ее параметры, в соответствии с предположением выше.

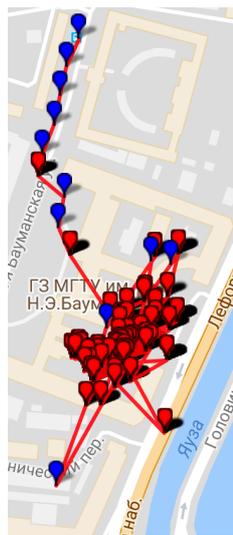
Коэффициенты P_K^9 и P_S^9 представляют собой опорные параметры фильтра. Очевидно, что при увеличении P_K^9 , фильтр способен определить более дальние выбросы координат при скорости от 0 до P_S^9 , а скорость P_S^9 фактически определяет границу фильтрации значительных выбросов в местах стоянок.

На Рисунке 2.31 представлены:

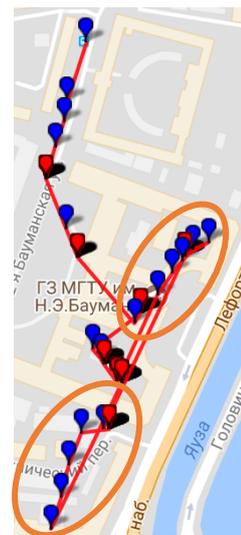
- а) трек четырехчасовой стоянки внутри главного здания МГТУ им. Н. Э. Баумана с небольшим участком движения;
- б) результат фильтрации диагностическим методом без учета скорости и входов/выходов из стоянки, при этом $D_{1-2} > 3 * (acc_1 + acc_2)$;
- в) результат фильтрации предложенным методом, при $P_K^9 = 20$, $P_S^9 = 2 \frac{м}{с}$.



(а) исходные данные



(б) метод [2]



(в) Предложенный метод

Рисунок 2.31. Сравнение диагностических методов фильтрации

Все коэффициенты были подобраны экспертным путем с учетом максимизации точек на участке трека движения и минимизации их в стоянке. Красными маркерами показаны точки, где скорость равна нулю, синим, где скорость положительна.

Фильтр отбросил большее количество точек непосредственно в стоянке, однако часть трека в движении слева сверху практически не изменилась. Стоит отметить, что девиации координат в левом нижнем и правом верхнем углу, выделенные оранжевым лучше отфильтровал метод диагностической фильтрации [2]. Однако в каскаде с фильтром стоянок, такие артефакты трека можно нивелировать. Гистограмма распределения скоростей представлена на Рисунке 2.32.

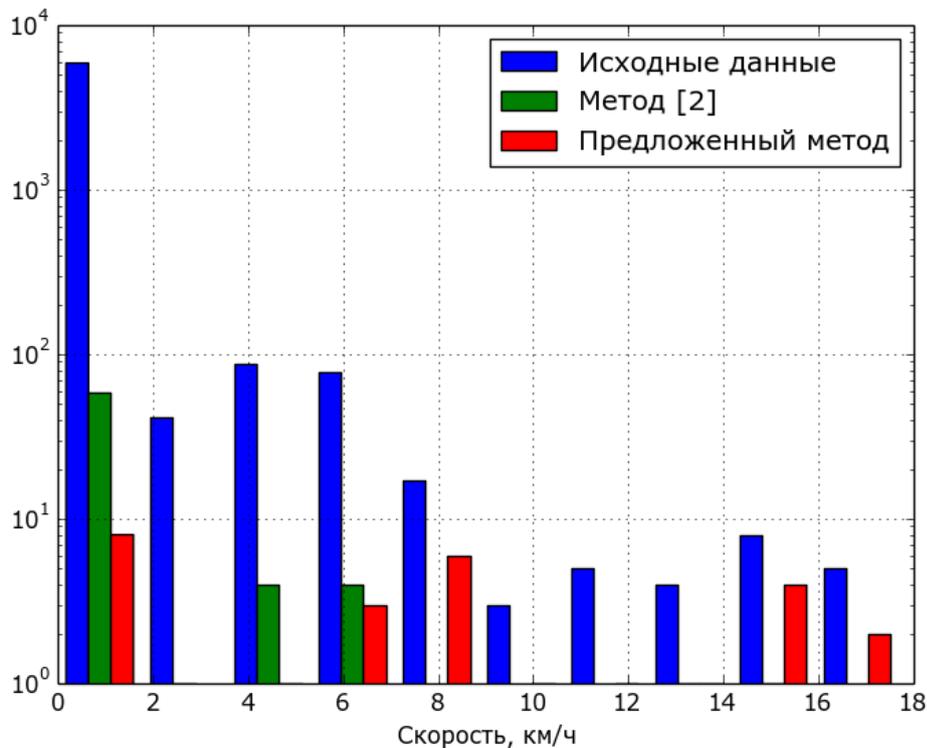


Рисунок 2.32. Распределение скорости после диагностической фильтрации

В Таблице 4 представлены характеристики метода диагностической фильтрации [2] и предложенного метода.

Сравнение диагностических методов фильтрации

Характеристика	Метод [2]	Предложенный метод	Предложенный метод относительно метода [2]
Процент отфильтрованных координат	98.872 %	99.559 %	60.87 %
Процент отфильтрованных координат с нулевой скоростью	99.006 %	99.863 %	86.207 %
Процент отфильтрованных координат с положительной скоростью	96.085 %	93.238 %	-72.727 %
Общее расстояние трека (реальное расстояние 357.4 м)	3545.094 м	1195.716 м	66.271 %
	74.243 %	91.313 %	

Исходя из анализа распределения скорости и сравнительной таблицы выше можно утверждать, что предложенный метод адаптивной диагностической фильтрации во многом более эффективней существующего. Особое внимание стоит обратить на характеристику общего расстояния трека. Во многих ГИС пройденная дистанция – один из основных параметров, практически все методы фильтрации нацелены на увеличение его достоверности.

Выводы по главе 2

Для решения нетривиальной задачи фильтрации геоинформации и построения трека пользователя, максимально приближенного к реальному маршруту, целесообразно использовать предложенный метод фильтрации потока геоданных, основанный на каскаде фильтров потоковой обработки, использующий 22 статических параметра, один общий параметр и три вычисляемых в процессе работы (см. Таблицу 5). Данная особенность вместе с обратными связями в фильтрах, памятью и, в ряде случаев, недостоверностью входных данных добавляет сложность при выявлении ошибок, в экспериментальных исследованиях и практической оценке результата. Существенное изменение какого-либо параметра

или функции фильтра в начале уже настроенного каскада в общем случае приводит к непредсказуемому поведению модели.

Предложенный метод фактически представляет собой бинарный классификатор с дискретными выходными значениями 0 или 1, которые являются критерием появления очередной координаты на выходе. В процессе работы также модифицируются параметры координат с целью сглаживания или определения стоянок пользователя на маршруте.

Для увеличения точности геопозиционирования входные и выходные данные для обработки следует представить в виде последовательности:

1) координат из основных источников: спутниковые навигационные приемники NMEA и API, Wi-Fi точки доступа, вышки сотовой связи, каждая из которых определена кортежем параметров: время получения, широта, долгота, hdp, точность, скорость, число спутников, тип, признак стоянки;

2) кортежей данных датчика-акселерометра: время, ускорение по оси x, y и z.

На основе результатов исследований различных типов выбросов геокоординат в качестве блока фильтрации предложено использовать последовательность из 9 фильтров, характер которых был взят исходя из характера ошибок входных параметров геоданных. При этом в начальных фильтрах обрабатываются основные характеристики геокоординат, такие как точность или число спутников. В последующих – рассчитываются и анализируются интегральные и дифференциальные показатели трека, на основе которых определяются статические и динамические пороги фильтров.

Результаты статистического анализа параметров отдельных координат и общих характеристик трека позволят наиболее эффективным образом подобрать параметры фильтрации. По результатам анализа и синтеза расчетных соотношений для исключения различного рода выбросов можно сделать следующие выводы:

1) фильтр по точности должен ограничивать координаты статическими пороговыми значениями параметров точности, hdp и числа спутников. Анализ

распределения точности выявил, что координаты NMEA и API с учетом допуска в 3σ в общем случае распределены на относительно небольшом отрезке: [0 м; 285.973 м] и [0 м; 121.324 м] соответственно, однако привнесение в общий список координат от сотовых вышек и Wi-Fi роутеров, которые распределены на отрезке [0 м; 1184.813 м], существенно увеличивает отрезок распределения точности – [0 м; 500.284 м]. Наиболее точные координаты имеют источник API. В распределениях явно прослеживаются кластеры координат в окрестностях многих точек, причина такого поведения – внутренняя реализация алгоритмов определения местоположения для каждого типа источника координат. Анализ распределения числа спутников и точности позволяет говорить об уменьшении среднего значения точности при увеличении среднего количества спутников. Минимальное среднее значение точности достигается при точности в 3.352 м и числе спутников, равном 17;

- 2) фильтр сетевых координат отдает приоритет спутниковым координатам, так как данный источник имеет большую точность. Время отсутствия спутниковых координат в момент появления сетевой распределено на отрезке [0 с; 4972.255 с] (3σ ; среднее значение $\mu = 90.513$ с, $\sigma = 1627.247$ с), а время отсутствия сетевой координаты с появления спутниковой – [0 с; 257.325 с] (3σ ; среднее значение $\mu = 3.775$ с, $\sigma = 84.517$ с). Поскольку последний отрезок полностью входит в первый, целесообразно вернуть сетевые координаты при прошествии определенного интервала времени, в котором отсутствуют спутниковые координаты. Данный метод позволит существенно повысить точность трека и сократить время отклика системы;
- 3) фильтр по времени отдает приоритет API координатам и предварительно снижает плотность распределения координат во времени для улучшения быстродействия дальнейшей фильтрации. Анализ распределения времени отсутствия и точности координат NMEA и API позволяет говорить о большей точности API координат, однако меньшей частотой приема от

источника: среднее время отсутствия NMEA координат составляет 2.387 с, против 6.435 с у API источника;

- 4) результаты анализа зависимости распределения параметра точности координаты от скорости показывают, что при больших значениях скорости, погрешность определения местоположения становится ниже. Фильтр по скорости ограничивает параметр скорости координаты максимальным значением и косвенно определяет погрешность измерения широты и долготы за счет сравнения пройденной дистанции с максимально возможной за определённое время. Анализ распределения точности координат до и после обработки фильтром показал, что фильтр отбрасывает порядка 4.458 % координат с большой погрешностью, при этом среднее значение точности в выборке уменьшается на 6.165 м (16.4 %);
- 5) фильтр по ускорению определяет порог максимального ускорения, что позволяет неявно определить погрешность определения скорости, широты и долготы координаты и как следствие, погрешность определения параметра точности. В этом случае увеличивается качество результирующего трека;
- 6) фильтр стоянок, анализируя дисперсию скорости и ускорения, определяет признак стоянки пользователя. Исходя из анализа треков до и после фильтрации можно сделать вывод, что фильтр позволяет исключить выбросы в местах стоянок с точностью в 98.872 % за продолжительный период относительно всех исходных координат и повысить точность определения пройденной дистанции на 74.24 %;
- 7) сглаживающий фильтр увеличивает качество результирующего трека в движении, когда возможны единичные выбросы координат. В качестве метода фильтрации целесообразно использовать экспоненциальное скользящее среднее. Преимуществом данного подхода является низкая вычислительная и емкостная сложность алгоритма и как следствие малое время выполнения, что особенно важно для программного обеспечения на мобильных устройствах;

- 8) угловой фильтр анализирует угол между направлениями движения по трем координатам в последовательности. Динамическое вычисление порогового значения максимального угла в зависимости от интервалов времени между координатами и скоростей пользователя в точках позволяет уменьшить плотность точек на прямых участках маршрута и увеличить при движении на поворотах. Из результатов анализа распределения угла между направлениями движения до и после обработки следует, что данный блок сместил среднее значение угла в центр распределения в значение 90° , однако дисперсия поменялась незначительно. Количество координат после фильтрации уменьшилось на 78.976 %, но информативность трека существенно не пострадала;
- 9) диагностический фильтр позволяет существенно увеличить качество выявления отклонений координат в местах стоянок посредством динамического определения пороговых значений расстояния между двумя смежными координатами в зависимости от параметра скорости и признаков входа и выхода из стоянки. На основе результатов анализа трека до и после фильтрации можно сделать вывод о том, что предложенный метод позволяет снизить плотность точек в треке на 60.87 %, увеличить достоверность определения координат в движении на 72.727 % и на 66.271 % повысить точность определения пройденной дистанции относительно метода [2].

Таблица 5.

Характеристики фильтров

№	Фильтр	Использует характеристики	Использует n предыдущих координат	Использует k предыдущих значений ускорения	Параметры фильтрации	Использует вычисленные параметры	Вычисляет внутренние параметры
0	Вся модель	-	-	-	$P_{maxStandSpeed}^0 \in \mathbb{R} \geq 0, \frac{M}{c}$	-	-
1	По точности	$sat, hdop, acc, tp$	-	-	$P_{minsat}^1 \in \mathbb{N} > 0$ $P_{maxhdop}^1 \in \mathbb{R} > 0$ $P_{maxAcc}^1 \in \mathbb{R} > 0, M$	-	-
2	Сетевых координат	t, tp	-	-	$P_{networkPeriod}^2 \in \mathbb{N} > 0, \text{мс}$	$Q_{isStand}$	$P_{lastGPSTime}^2 \in \mathbb{N} \geq 0, \text{мс}$
3	По времени	t, tp	1	-	$P_{period}^3 \in \mathbb{N} > 0, \text{мс}$ $P_{locationPeriod}^3 \in \mathbb{N} > 0, \text{мс}$	$Q_{isStand}$	$P_{lastLocationTime}^3 \in \mathbb{N} \geq 0, \text{мс}$
4	По скорости	t, lat, lon, spd	1	-	$P_{maxSpeed}^4 \in \mathbb{R} > 0, \frac{M}{c}$ $P_{maxTime}^4 \in \mathbb{N} > 0, \text{мс}$ $P_{distK}^4 \in \mathbb{R} \geq 1, M$	-	-
5	По ускорению	t, spd	1	-	$P_{maxAccel}^5 \in \mathbb{R} > 0, \frac{M}{c^2}$	-	-

Таблица 5 (продолжение).

№	Фильтр	Использует характеристики	Использует n предыдущих координат	Использует k предыдущих значений ускорения	Параметры фильтрации	Использует вычисленные параметры	Вычисляет внутренние параметры
6	Стоянок	t, lat, lon t, ax, ay, az	n	k	$P_{periodAccel}^6 \in \mathbb{N} > 0, \text{мс}$ $P_{accelCut}^6 \in \mathbb{R} > 0, \left(\frac{\text{M}}{\text{с}^2}\right)^2$ $P_{periodCoord}^6 \in \mathbb{N} > 0, \text{мс}$ $P_{speedCut}^6 \in \mathbb{R} > 0, \left(\frac{\text{M}}{\text{с}^2}\right)^2$	-	$Q_{isStand}$
7	Сглаживающий	lat, lon	1	-	$P_{smoothBase}^7 \in \mathbb{R} \in [0.0; 1.0]$ $P_{maxSmoothDistance}^7 \in \mathbb{R} > 0, \text{M}$	-	-
8	Угловой	t, lat, lon, spd	2	-	$P_{maxAngle}^8 \in \mathbb{R} \in [0; 180], ^\circ$ $P_{maxTimeDiff}^8 \in \mathbb{R} > 0, \text{мс}$ $P_{maxSpeed}^8 > 0, \frac{\text{M}}{\text{с}}$	-	-
9	Диагностический	t, acc, spd, tp, s	1	-	P_K^9, P_S^9	-	-

Глава 3. Алгоритм обработки потока геоданных в реальном времени

В главе 2 был рассмотрен метод обработки геоинформации, основой которого является каскад фильтров. Однако для полноценной реализации системы фильтрации потока геоданных необходимо доопределить модели входных, выходных данных и фильтрации путем расширения и модификации предложенных моделей и методов с учетом специфики операционной системы Android [45]. Рассмотренные далее дополнительные методы фильтрации следует добавить в исходную модель отдельными блоками, не модифицируя базовые модули, разработанные в главе 2. В этом случае комплексная модель потоковой фильтрации будет реализована в виде самодостаточной подсистемы, ее можно будет использовать на разных платформах, в разном окружении, обеспечив необходимыми параметрами и исходными данными. На основе уточненной модели построим алгоритм фильтрации потока геоданных, используя который возможно реализовать систему фильтрации на целевой мобильной платформе, и оценим его эффективность, определив время выполнения, количество необходимой памяти и вычислительную сложность.

3.1. Уточнение модели данных и метода обработки геокоординат

3.1.1. Получение геоданных и препроцессинг

В мобильных устройствах, работающих под управлением операционной системы Android, присутствуют два источника геокоординат, которые агрегируют все источники необработанных геолокационных данных: NMEA Receiver и Location API (см. Рисунок 3.1). Для данных акселерометра в операционной системе используется отдельный источник – Accelerometer API.



Рисунок 3.1. Источники данных в ОС Android

Данные, получаемые из источников, фактически представляют собой наборы параметров координат или ускорений мобильного устройства в исходной форме, рассмотренной в разделе 2.1. Дополнительная обработка здесь не требуется, типизирование и унифицирование кортежей координат и ускорений соответствует предложенному методу препроцессинга.

Обработка данных NMEA источника заключается в разборе NMEA строк типа GGA и VTG и дополнении исходной информации параметрами точности, типа источника и признака стоянки: *acc*, *tp* и *s* соответственно. В рамках преобработки данных спутникового навигационного приемника, полученных из Location API источника, в исходный набор характеристик координаты необходимо добавить параметры геометрического снижения точности в горизонтальной плоскости, типа источника и признака стоянки: *hdop*, *tp* и *s* в соответствии с Таблицей 2 соответственно; в данные Wi-Fi и сотовых вышек – параметры геометрического снижения точности, скорости, числа спутников, типа источника и признака стоянки: *hdop*, *spd_i*, *sat*, *tp* и *s* соответственно. Данные акселерометра

не нуждаются в препроцессинге. Алгоритмы препроцессинга координат будут подробно рассмотрены далее.

3.1.2. Фильтрация типов источников и спутников

Разные мобильные устройства могут содержать различные навигационные модули, работать на разных версиях операционных систем, содержать различные модификации аппаратной и программной части. Устройства также могут получать GPS координаты от разных типов спутников. В целях увеличения гибкости настройки системы фильтрации целесообразно добавить блок обработки, ограничивающий наборы источников геолокационных данных и типов спутников, по которым можно осуществлять геопозиционирование. В первом случае возможно анализировать параметр tp_i очередной координаты. Во втором необходимо расширить описание входных данных, введя параметр $styp_e_i$, определяющий тип спутников, по которым была определена очередная координата. Значение параметра предоставит NMEA источник.

Первый блок обработки пропускает только те координаты, источник которых есть во множестве допустимых:

$$F_1'(\{\{tp_i\}\}, P_{sourceTypes}^{1'}) = \begin{cases} 1, & tp_i \in P_{sourceTypes}^{1'} \\ 0, & \text{иначе} \end{cases}, \quad (3.1)$$

где $P_{sourceTypes}^{1'} \subseteq \{0,1,2\}$ является подмножеством допустимых типов источников.

Второй фильтр пропускает только NMEA координаты, полученные по спутникам, тип которых есть во множестве допустимых:

$$F_2'(\{\{tp_i, styp_e_i\}\}, P_{satelliteTypes}^{2'}) = \begin{cases} 1, & tp_i \neq 0 \text{ или } styp_e_i \in P_{satelliteTypes}^{2'} \\ 0, & \text{иначе} \end{cases}, \quad (3.2)$$

где $P_{satelliteTypes}^{2'} \subseteq \{0,1,2,3\}$ – подмножество допустимых типов спутников.

Фильтры целесообразно разместить в самом начале обработки после препроцессинга данных. В этом случае далее будут переданы только координаты требуемых типов, что увеличит скорость обработки.

3.1.3. Фильтрация ложных координат

В Android устройствах с целью отладки и тестирования возможно программно заменить источники координат. Однако, невозможно гарантировать, что на определенном устройстве этот механизм не будет использован приложениями. В этом случае система может получать ложные координаты, что недопустимо. Расширим исходный набор входных данных новым параметром и добавим фильтр, позволяющий определять ложные координаты:

$$F'_3(\{\{isMocked_i\}\}, P_{isMockAllowed}^{3'}) = \begin{cases} 1, & P_{isMockAllowed}^{3'} = 1 \text{ или } isMocked_i = 0 \\ 0, & \text{иначе} \end{cases}, \quad (3.3)$$

где $P_{isMockAllowed}^{3'} \in \{0,1\}$ – параметр модели, определяющий допустимость использования ложных координат;

$isMocked_i \in \{0,1\}$ – признак ложной координаты.

Аналогично предыдущему фильтру, данный блок необходимо разместить в самом начале последовательности фильтров.

3.1.4. Использование дополнительной информации из мобильного приложения

Как было рассмотрено ранее, по действиям пользователя в мобильном приложении возможно косвенно определить геопозицию. Можно предложить 4 варианта реализации использования дополнительной семантической информации:

- 1) источник данных. В этом случае в модели необходимо реализовать дополнительный источник, который будет использоваться внешним по отношению к фильтрации приложением как интерфейс передачи координат в модель обработки. Координаты соответственно должны обладать всеми начальными параметрами. Они будут подвергаться обработке всеми фильтрами и в результате могут быть исключены, допустим угловым фильтром, если пользователь движется по прямой;

- 2) источник особого вида данных. Подход аналогичен первому, здесь также необходима реализация дополнительного источника данных, однако координаты, полученные от такого источника, будут обладать дополнительным признаком. Фильтрам следует всегда пропускать координаты с установленным признаком, чтобы в результате они появились на выходе системы, и обновлять внутренние расчётные параметры и сохраняемые данные предыдущих координат. Подход, в отличие от первого, гарантирует получение координаты на выходе системы;
- 3) триггер. Зачастую, в конечном мобильном приложении необходимо определить текущее местоположение настолько быстро, насколько это возможно, но данные о предполагаемой широте и долготе в приложении отсутствуют, то есть подходы 1 и 2 не применимы. В случае продолжительной стоянки пользователя внутри здания, исходные координаты могут быть достаточно точными только для определения фильтром стоянок того, что пользователь еще находится в стоянке, однако такие координаты не будут появляться на выходе фильтрации. Они будут учтены фильтром стоянок только как продолжение текущей стоянки. Аналогичная ситуация возникает в угловом фильтре за счет прореживания координат. Чтобы обеспечить меньшее время отклика системы, целесообразно ввести в модель триггер, который будет сбрасывать текущие состояния и сохраненные значения параметров некоторых фильтров. Тем самым на непродолжительное время до получения следующих координат, когда завершится переходный процесс в фильтрах, незначительно уменьшится точность фильтрации, но увеличится скорость их выдачи. Исходя из анализа метода обработки, фильтры, регулируя поведение которых возможно уменьшить время отклика системы – это фильтры сетевых координат, по времени, по скорости, по ускорению, стоянок, сглаживающий, угловой и диагностический.
- 4) комбинация подходов 1-3.

Далее будет детально рассмотрен и применен третий подход.

3.1.5. Кеш координат и состояний

Поскольку многие блоки фильтрации хранят параметры нескольких координат за определенный период времени и вычисляемые внутренние параметры, целесообразно добавить в систему общую память – кеш координат. За счет этого система сможет не только восстанавливать свое состояние после перезагрузки системы или отдельного приложения, но и гибко настраивать параметры фильтров, например, число хранимых координат в фильтре стоянок, под размер общей памяти.

Поскольку фильтры образуют последовательность, стоит отметить, что в процессе работы предыдущие значения сохраняемых параметров в фильтрах могут отличаться. Например, значение параметра t_{i-1} предыдущей координаты в фильтре по скорости может отличаться от того же значения параметра в фильтре по времени, который размещён раньше в наборе. В кеше необходимо резервировать память под каждый хранимый параметр каждого фильтра. В этом случае реализация использования дополнительной информации из конечного мобильного приложения по третьему предложенному подходу будет относительно простой, достаточно обнулить параметры всех фильтров, чтобы сбросить состояние фильтрации в начальное, что увеличит время отклика системы.

Введем общий параметр $P_{cacheSize}^{0'} \in \mathbb{N} \geq 1$ – размер кеша и рассмотрим данные, сохраняемые рассматриваемым модулем (см. Таблицу 6).

Таблица 6.

Данные кеша координат и состояний

№	Тип	Данные	Используют фильтры
1	Координаты	$\{ \langle t_{i-P_{cacheSize}^{0'}+1} \rangle \dots \langle t_i \rangle \}$ $\{ \langle lat_{i-P_{cacheSize}^{0'}+1} \rangle \dots \langle lat_i \rangle \}$ $\{ \langle lon_{i-P_{cacheSize}^{0'}+1} \rangle \dots \langle lon_i \rangle \}$	Стоянок

Таблица 6 (продолжение).

№	Тип	Данные	Используют фильтры
2	Ускорения	$\{ \langle t_{i-P_{cacheSize}^{0'}} \rangle \dots \langle t_i \rangle \}$ $\{ \langle ax_{i-P_{cacheSize}^{0'}} \rangle \dots \langle ax_i \rangle \}$ $\{ \langle ay_{i-P_{cacheSize}^{0'}} \rangle \dots \langle ay_i \rangle \}$ $\{ \langle az_{i-P_{cacheSize}^{0'}} \rangle \dots \langle az_i \rangle \}$	Стоянок
3	Параметр	$P_{lastGPSTime}^2$	Сетевых координат
4	Параметр	$P_{lastLocationTime}^3$ t_{i-1}	По времени
5	Параметр	t_{i-1} lat_{i-1} lon_{i-1}	По скорости
6	Параметр	t_{i-1} spd_{i-1}	По ускорению
7	Параметр	$Q_{isStand}$	Стоянок
8	Параметр	lat_{i-1} lon_{i-1}	Сглаживающий
9	Параметр	t_{i-1} lat_{i-1}, lat_{i-2} lon_{i-1}, lon_{i-2} spd_{i-1}, spd_{i-2}	Угловой
10	Параметр	lat_{i-1} lon_{i-1} acc_{i-1} spd_{i-1} tp_{i-1} s_{i-1}	Диагностический

Таблица может быть также расширена всеми настроечными параметрами фильтрации. В этом случае кеш координат и состояний будет представлять собой блок хранения всей информации как о системе фильтрации, так и о ее текущем

состоянии. Это делает удобным разработку и отладку алгоритмов модели, ее использование, дальнейшую модификацию, расширение и дополнение.

3.1.6. Метод фильтрации геоданных в реальном времени

На основе анализа источников геолокации Android OS и блоков фильтрации, приведенных в главах 1 и 2, итоговые входные и выходные данные должны быть представлены кортежем координат x_i и ускорений a_i :

$$x_i = \langle t_i, lat_i, lon_i, hdop_i, acc_i, spd_i, sat_i, tp_i, s_i, stype_i, isMocked_i \rangle, \quad (3.4)$$

$$a_i = \langle t_i, ax_i, ay_i, az_i \rangle, \quad (3.5)$$

где $t \in \mathbb{N} > 0$ – время фиксации точки местоположения в формате Unix Epoch Time в миллисекундах;

$lat \in \mathbb{R} \geq 0$ – широта точки местоположения в градусах;

$lon \in \mathbb{R} \geq 0$ – долгота точки местоположения в градусах;

$spd \in \mathbb{R} \geq 0$ – моментальная скорость в текущей точке в метрах в секунду;

$acc \in \mathbb{R} > 0$ – радиус круга в метрах, в котором находится текущие координаты lat и lon с вероятностью $\sim 68\%$ (1σ); параметр относится к данным, полученным от API;

$sat \in \mathbb{N} \geq 0$ – число спутников, принимающих участие в определении текущей точки местоположения;

$hdop = \sqrt{std_{lat}^2 + std_{lon}^2} \in \mathbb{N} > 0$ – геометрическое снижение точности в горизонтальной плоскости;

$ax \in \mathbb{R} \geq 0$ – ускорение мобильного устройства по оси x в метрах в секунду в квадрате;

$ay \in \mathbb{R} \geq 0$ – ускорение мобильного устройства по оси y в метрах в секунду в квадрате;

$az \in \mathbb{R} \geq 0$ – ускорение мобильного устройства по оси z в метрах в секунду в квадрате.

Параметры, выделенные **полужирным** шрифтом, являются характеристиками базовой модели, разработанной в главе 2.

Система фильтрации должна использовать все параметры Таблицы 5, а также новые предложенные характеристики для работы фильтров по типам источников и спутников, ложных координат, кеша координат и состояний (см. Таблицу 7).

Таблица 7.

Данные кеша координат и состояний

№	Фильтр	Использует характеристики	Параметры фильтрации
0	Вся модель	-	$P_{cacheSize}^{0'} \in \mathbb{N} \geq 2$
1	Типов источников	<i>tp</i>	$P_{sourceTypes}^{1'} \in \{0,1,2\}$
2	Типов спутников	<i>stype</i>	$P_{satelliteTypes}^{2'} \in \{0,1,2,3\}$
3	Ложных координат	<i>isMocked</i>	$P_{isMockAllowed}^{3'} \in \{0,1\}$

Итоговый метод фильтрации геокоординат на основе каскада 12-ти фильтров потоковой обработки использует 25 статических параметров, два общих параметра и три вычисляемых в процессе работы. На Рисунке 3.2 представлена его графическая интерпретация. Зеленым выделены базовые фильтры, рассмотренные во второй главе. Существенное отличие предложенного метода от существующих систем фильтрации состоит в том, что он использует все возможные источники геоданных, акселерометр мобильного устройства, блок разрежения точек трека, а также диагностическую обработку, учитывающую скорость устройства, что позволяет снизить выбросы в местах стоянок и уменьшить плотность точек трека.

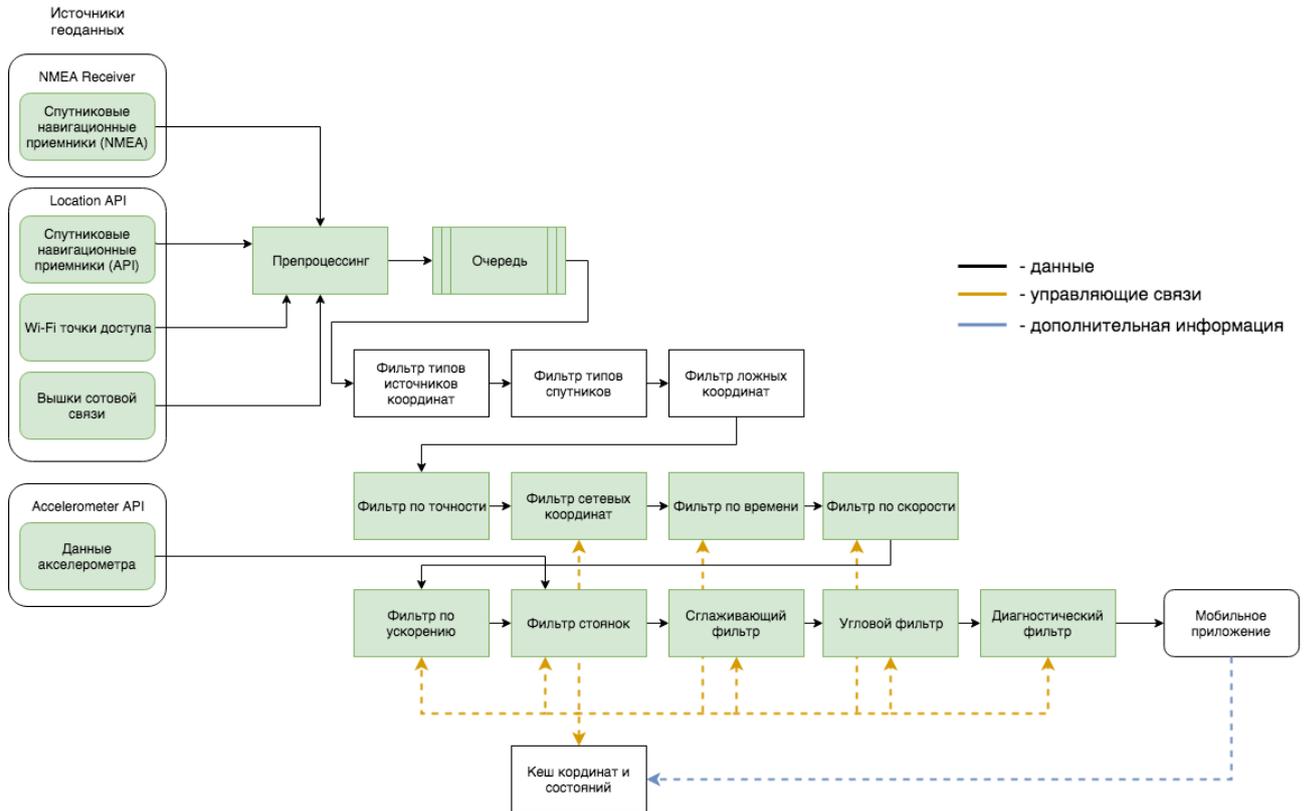


Рисунок 3.2. Схема фильтрации потока геоданных

3.2. Разработка алгоритмов препроцессинга и фильтрации

Построим алгоритмы работы блоков препроцессинга и фильтрации. Функцию $F_j(\{x_i\}, \{P^j\})$ каждого фильтра будем представлять в виде функции *filter* в схеме алгоритма, которая на вход принимает необходимые значения параметров очередной координаты s , соответствующей координате x_i , пришедшей из какого-либо источника. Параметры фильтрации $\{P^j\}$ в наборе входных переменных опущены, так как могут быть заданы статически после их подбора. На выход функция либо возвращает *False*, что соответствует $F_j(\{x_i\}, \{P^j\}) \equiv 0$, либо *True* в случае $F_j(\{x_i\}, \{P^j\}) \equiv 1$.

Будем также оценивать время выполнения алгоритмов фильтрации в лучшем и худшем случаях, объем памяти, необходимой для работы алгоритмов, их вычислительную сложность. Оценку времени выполнения в лучшем случае будем обозначать как $T_{k,l}^{min}$, где $k = 2$, если фильтр относится к базовой модели,

$k = 1$, когда фильтр относится к расширенной; $l \in \mathbb{N} > 0$ – порядковый номер фильтра в последовательности. Оценка времени выполнения в худшем случае обозначена $T_{k,l}^{max}$, оценка емкостной сложности – $E_{k,l}$, вычислительная сложность алгоритма – $C_{k,l}$. В рамках определения необходимой памяти будем анализировать только входные параметры алгоритма и необходимые переменные для вычислений. Параметры фильтрации в общем случае могут быть заданы константами, за хранение параметров предыдущих координат полностью отвечает кеш координат и состояний.

3.2.1. Алгоритмы препроцессинга данных

Алгоритм препроцессинга данных NMEA источника должен обрабатывать NMEA строки типа GGA и VTG, заполняя необходимую информацию согласно модели данных (см. Рисунок 3.3).

Функция *parse* вызывается при каждом получении сообщения от NMEA источника. Входными параметрами являются *time* – время получения сообщения в формате UNIX-time и *nmea* – NMEA сообщение в формате строки. Поскольку обработка предусмотрена только для GGA и VTG сообщений NMEA, алгоритм разделен на две общие ветви, в первой из которых анализируется сообщение GGA, создается объект координаты и сохраняется в локальный для блока препроцессинга словарь; во второй – сообщение VTG, где сохраненной координате присваивается время получения, скорость и признак стоянки. Определения типа сообщения GGA или VTG производится функцией *contains*, проверяющей вхождение подстрок «GGA» и «VTG» в строку *nmea*.

В случае приема GGA сообщения параметры координат сохраняются в массив строк *params* функцией *split*. Далее проверяется параметр *quality*, полученный путем приведения строки к числу типа *int* функцией *parseInt*. Затем из строки выделяются необходимые данные для формирования координаты:

- *lat* – долгота;
- *lon* – широта;

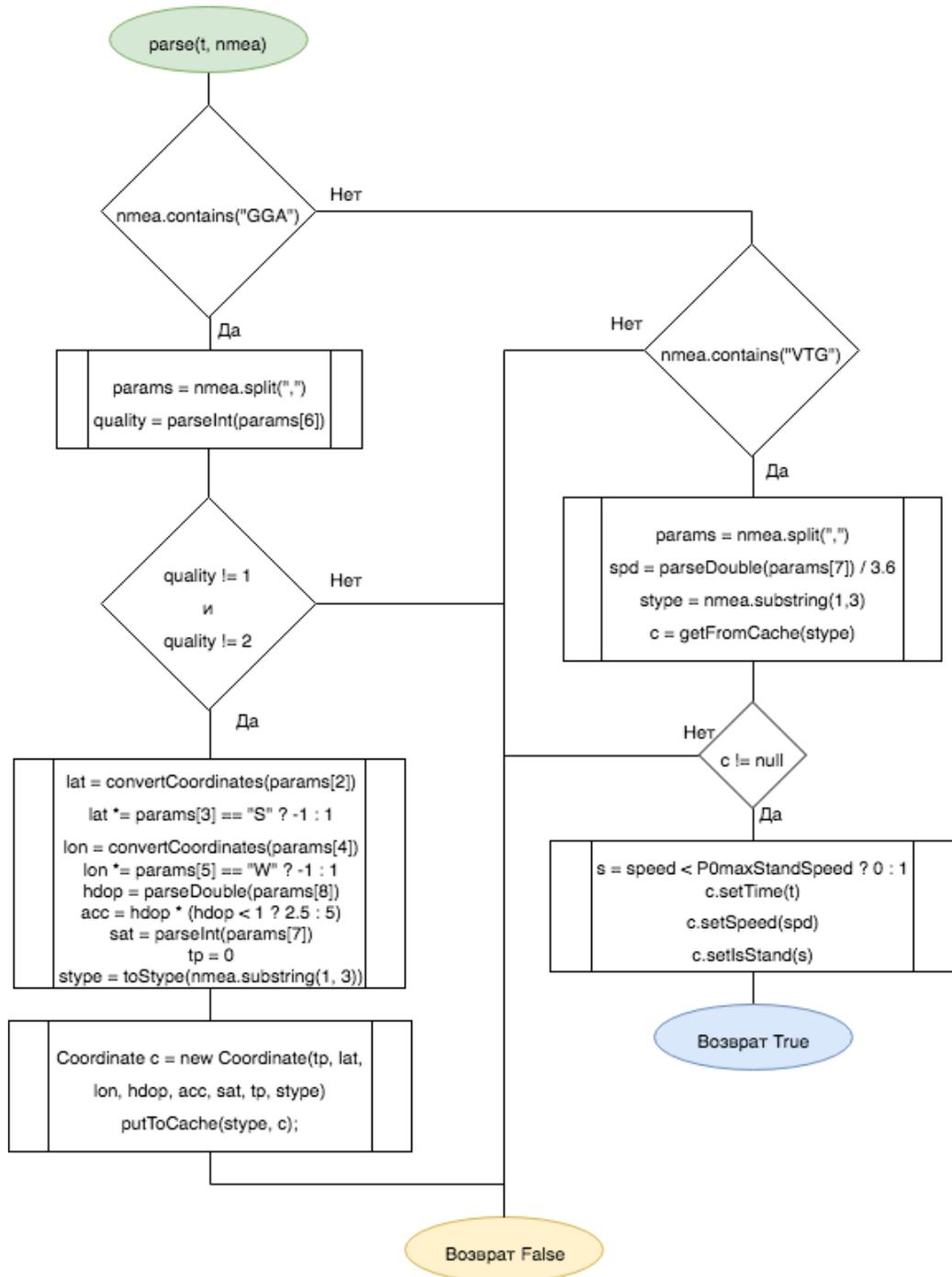


Рисунок 3.3. Схема алгоритма преобработки данных NMEA источника

- *hdop* – параметр снижения точности в горизонтальной плоскости;
- *acc* – точность;
- *sat* – число спутников;
- *tp* – тип источника;

stype – тип спутников.

При этом используются функции:

- *convertCoordinates*, которая преобразует широту и долготу из формата градусов-минут-секунд в формат градусов с десятичной точкой;
- *parseDouble* – приводит строку к числу типа *double*;
- *substring* – возвращает подстроку длиной 3 начиная с первого символа;
- *toStype* – приводит строковый тип спутников к числу типа *int*.

В завершении создается объект координаты *c*, которая сохраняется функцией *putToCache* в локальный кеш-словарь по ключу *stype*, что позволяет дополнить координату параметром скорости далее.

При получении сообщения VTG из исходной строки выделяется параметр скорости и вычисляется признак стоянки *s* в зависимости от параметра фильтрации *P0maxStandSpeed*. Значения заносятся в координату, полученную функцией *getFromCache* из кеша по ключу *stype*. Алгоритм завершается возвратом управления с координатой *c*, в которой присутствуют все необходимые данные для дальнейшей фильтрации.

Рассмотрим алгоритм предобработки геоданных API источника (см. Рисунок 3.4). На рисунке представлены алгоритмы двух функций предобработки данных API источника:

- для спутниковых координат – *parse1*;
- для координат Wi-Fi и сотовых вышек – *parse2*.

Первый алгоритм вычисляет значение параметра *acc* координаты в зависимости от значения *hdop*. Далее рассчитывается признак стоянки *s* с учетом переменной *P0maxStandSpeed*, которая соответствует общему параметру $P_{maxStandSpeed}^0$ модели фильтрации. Затем координате присваивается тип *tp* и параметр *stype*. Алгоритм завершается созданием объекта координаты *c* со всеми необходимыми параметрами.

Второй алгоритм значительно менее трудозатратен с точки зрения вычислительной сложности, чем первый. В нем многие переменные координаты устанавливаются константными значениями в соответствии с Таблицей расчетных соотношений 2. Как и в предыдущем случае он завершается созданием объекта координаты *c*.

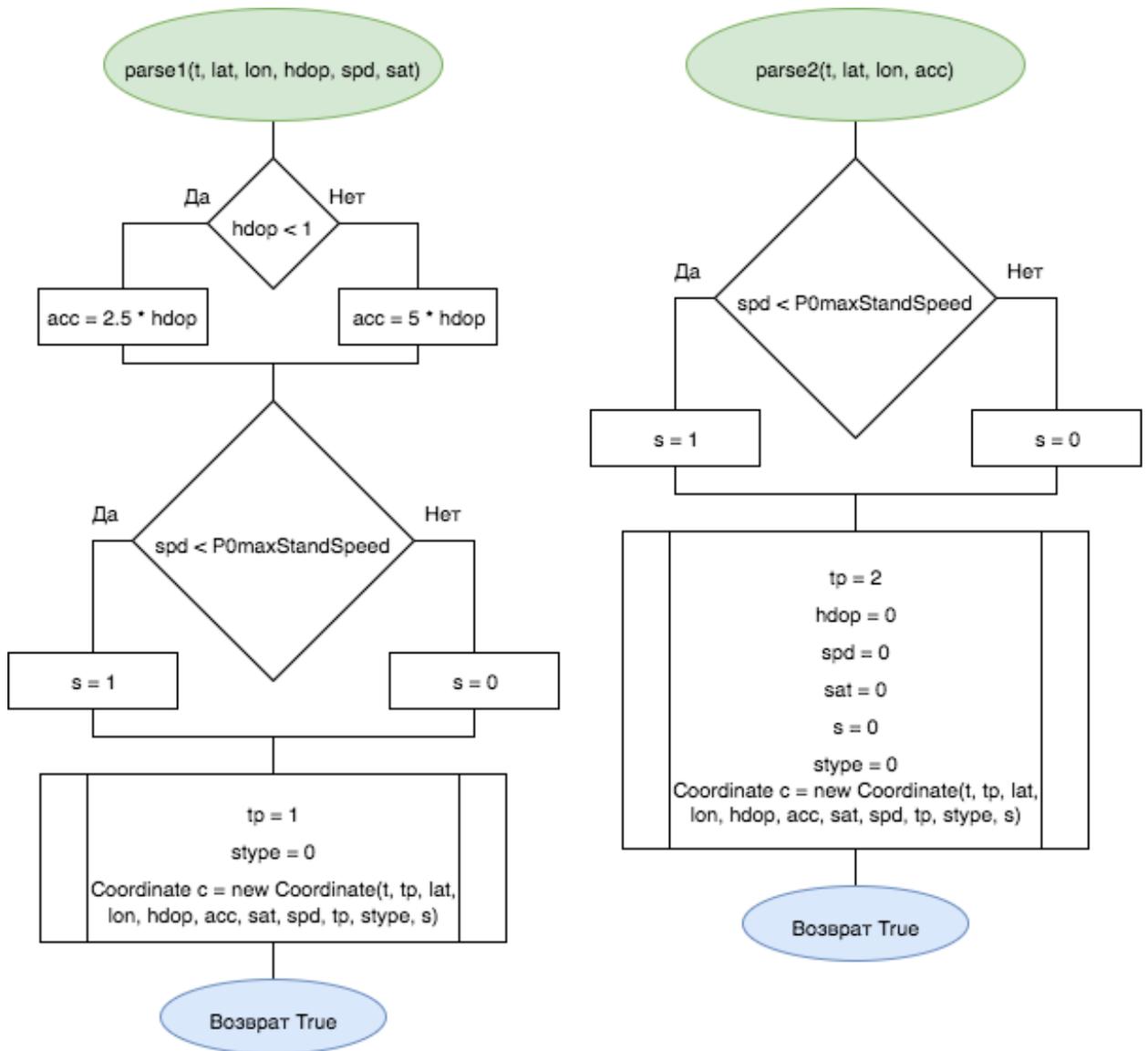


Рисунок 3.4. Схема алгоритма препроцессинга данных API источника

3.2.2. Алгоритм фильтрации по типам источников

Цель фильтрации по типам источников – ограничить обработку координат, типа источника которых нет во множестве допустимых. Алгоритм работы фильтра представлен на Рисунке 3.5.

Множество $P1_1_sourceTypes$ соответствует параметру $P1'_{sourceTypes}$, функция *contains* проверяет нахождение элемента во множестве. Если тип источника tp координаты c присутствует во множестве допустимых типов $P1_1_sourceTypes$, координата проходит фильтрацию, иначе – алгоритм возвращает *False*.

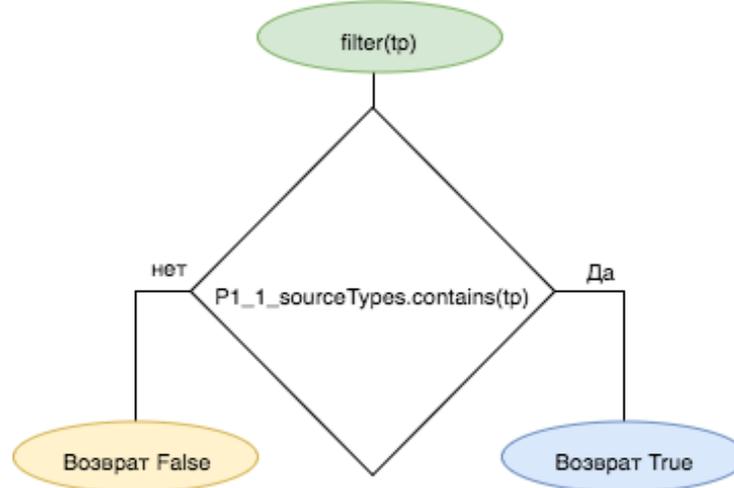


Рисунок 3.5. Схема алгоритма работы фильтра по типу источников

При условии, что множество $P1_1_sourceTypes$ реализуется хеш таблицей, время выполнения алгоритма проверки нахождения элемента во множестве и в лучшем, и в худшем случае соответствует константе:

$$T_{1,1}^{min} = T_{1,1}^{max} = t_{base}, \quad (3.6)$$

где t_{base} – время проверки нахождения элемента в хеш таблице.

Поскольку фильтр не хранит предыдущие значения параметров координат и вычисляемые параметры, оценка памяти, которая требуется алгоритму, состоит только из объема памяти для хранения переменной tp , представленной в виде целочисленного типа int . Оценка объема памяти:

$$E_{1,1} = e_{int}, \quad (3.7)$$

где e_{int} – объем памяти, необходимой для хранения числа типа int .

Вычислительная сложность алгоритма соответствует:

$$C_{1,1} = O(1). \quad (3.8)$$

3.2.3. Алгоритм фильтрации по типам спутников

Цель фильтрации по типам спутников – ограничить обработку координат, типа спутников которых нет во множестве допустимых. Алгоритм работы фильтра представлен на Рисунке 3.6.

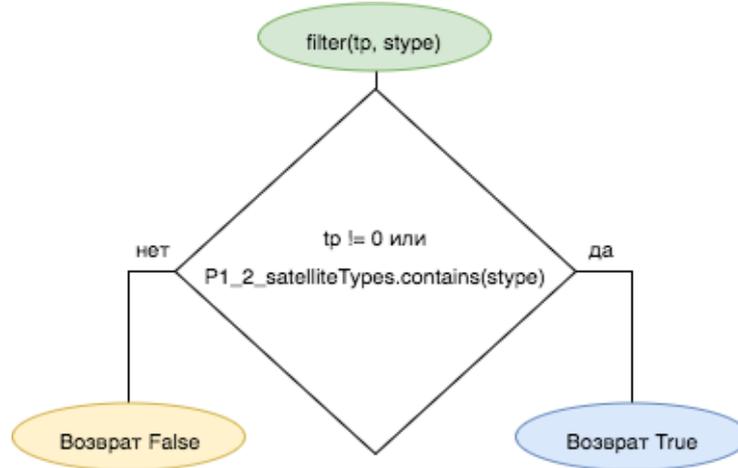


Рисунок 3.6. Схема алгоритма работы фильтра по типу спутников

Множество $P1_2_satelliteTypes$ соответствует параметру $P_{satelliteTypes}^{2'}$. В процессе обработки проверяется нахождение типа спутников $stype$ очередной координаты s во множестве допустимых только для координат с типом 0, что соответствует NMEA источнику.

Оценка времени выполнения алгоритма в лучшем случае представляет собой время выполнения операции сравнения параметра tp с нулем; в худшем случае – состоит из операции сравнения и операции проверки нахождения элемента в хеш таблице. Для упрощения вычисления оценки, примем время выполнения операции сравнения, равным t_{base} . Итоговые оценки в лучшем и худшем случаях соответственно:

$$T_{1,2}^{min} = t_{base}; \quad (3.9)$$

$$T_{1,2}^{max} = 2t_{base}. \quad (3.10)$$

Фильтр не хранит предыдущие значения параметров координат и вычисляемые параметры, поэтому оценка необходимой памяти определяется объемом памяти для переменных tp и $stype$. Для переменной $stype$ следует также использовать целочисленный тип int . Аналогично предыдущему фильтру:

$$E_{1,2} = 2e_{int}. \quad (3.11)$$

Вычислительная сложность алгоритма соответствует:

$$C_{1,2} = O(1). \quad (3.12)$$

3.2.4. Алгоритм фильтрации ложных координат

Цель обработки – определение и фильтрация ложных координат. Алгоритм работы фильтра представлен на Рисунке 3.7.

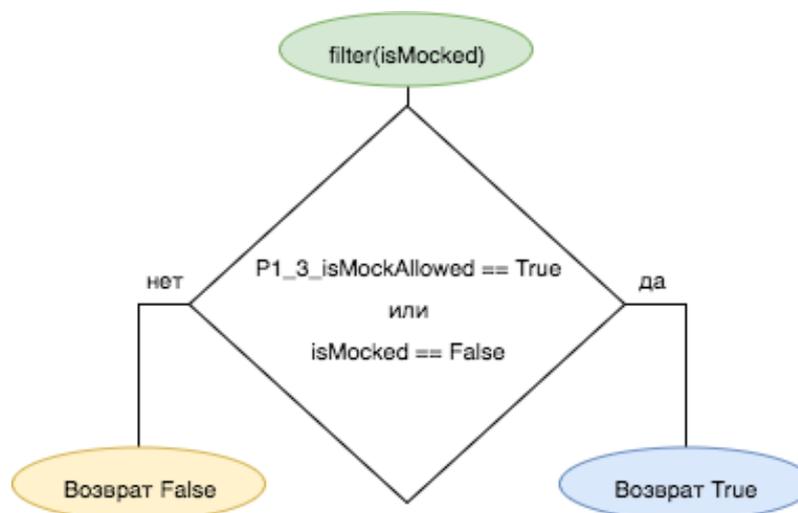


Рисунок 3.7. Схема алгоритм работы фильтра фальшивых координат

Переменная $P1_3_isMockAllowed$ соответствует параметру $P_{isMockAllowed}^{3'}$. Фильтр проверяет признак ложной координаты $isMocked$. Если координата ложная, и модель не допускает наличия поддельных координат, фильтр не пропускает координату.

Время выполнения алгоритма в лучшем случае состоит из времени выполнения одной операции сравнения, в худшем случае – из двух:

$$T_{1,3}^{min} = t_{base}; \quad (3.13)$$

$$T_{1,3}^{max} = 2t_{base}. \quad (3.14)$$

Память, необходимая для алгоритма, определяется только входной переменной $isMocked$, которая может быть представлена типом $boolean$:

$$E_{1,3} = e_{boolean} \quad (3.15)$$

где $e_{boolean}$ – объем памяти, необходимой для хранения числа типа $boolean$.

Вычислительная сложность алгоритма соответствует:

$$C_{1,3} = O(1). \quad (3.16)$$

3.2.5. Алгоритм фильтрации по точности

Цель фильтрации – исключить на начальном этапе координаты со значительной погрешностью, анализируя параметры точности acc_i , $hdop_i$ и sat_i . На Рисунке 3.8 представлен алгоритм работы фильтра.

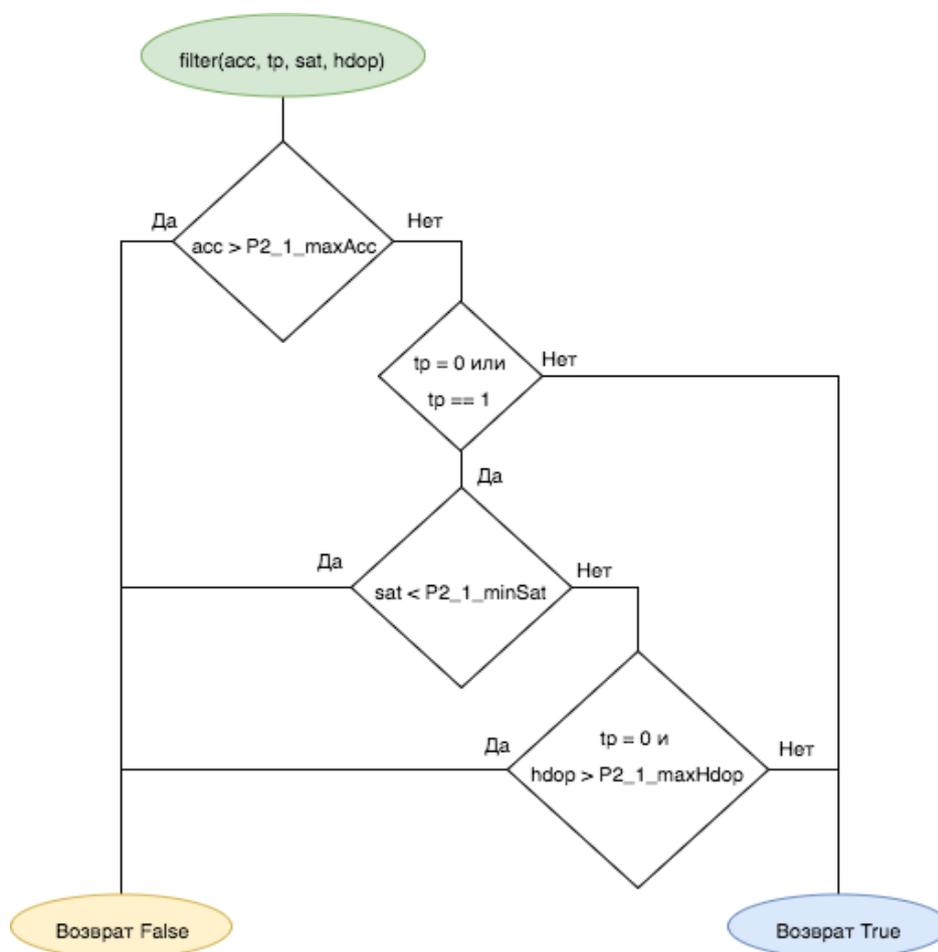


Рисунок 3.8. Схема алгоритма работы фильтра по точности

Параметр $P2_1_maxAcc$ соответствует параметру модели P_{maxAcc}^1 , $P2_1_minSat - P_{minSat}^1$, $P2_1_maxHdop - P_{maxHdop}^1$. В первую очередь алгоритм сравнивает параметр точности координаты acc с порогом фильтрации $P2_1_maxAcc$. Если координата получена от источников NMEA или API, фильтр анализирует число спутников sat , если очередная координата получена от NMEA источника, сравнивается параметр $hdop$.

Оценка времени выполнения в лучшем случае определяется одной операцией сравнения по точности в первом блоке алгоритма:

$$T_{2,1}^{min} = t_{base}. \quad (3.17)$$

В худшем случае алгоритм проведет 5 операций сравнения перед тем, как пропустить координату:

$$T_{2,1}^{max} = 5t_{base}. \quad (3.18)$$

Память, необходимая для алгоритма, определяется входными переменными acc , tp , sat и $hdop$. Переменную sat целесообразно хранить в целочисленном формате int , для переменных $hdop$ и acc следует использовать вещественный тип $double$. Итоговая оценка памяти:

$$E_{2,1} = 2e_{int} + 2e_{double}, \quad (3.19)$$

где e_{double} – объем памяти, необходимый для хранения числа типа $double$.

Вычислительная сложность алгоритма соответствует:

$$C_{2,1} = O(1). \quad (3.20)$$

3.2.6. Алгоритм фильтрации сетевых координат

Цель фильтрации – предотвратить влияние низкой точности сетевых координат на следующие фильтры в последовательности. На Рисунке 3.9 представлен алгоритм работы фильтра.

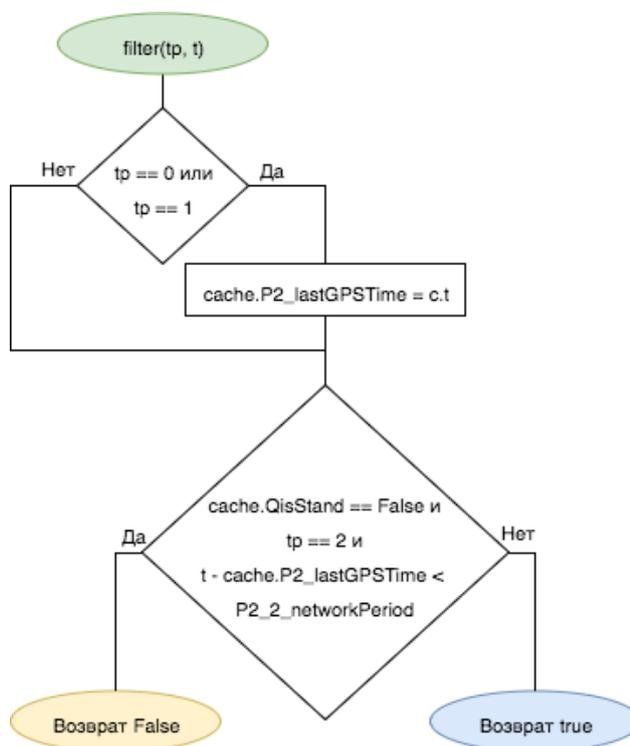


Рисунок 3.9. Схема алгоритма работы фильтра сетевых координат

В соответствии с методом фильтрации, разработанным в главе 2, если координата получена от спутниковых навигационных приемников фильтр запоминает значение времени получения координаты t в переменную $P2_lastGPSTime$, которая соответствует параметру $P2_{lastGPSTime}^2$ кеша координат и состояний *cache*. Затем анализирует признак стоянки $QisStand$, соответствующий параметру $QisStand$, полученному из кеша, ограничивая проход через алгоритм сетевых координат. $P2_2_networkPeriod$ соответствует параметру $P2_{networkPeriod}^2$, который определяет максимальное время, в рамках которого будут исключены сетевые координаты.

Оценка времени выполнения в лучшем случае определяется двумя операциями сравнения в первом блоке алгоритма и сравнением параметра $QisStand$ с *False*:

$$T_{2,2}^{min} = 2t_{base}. \quad (3.21)$$

Для упрощения вычислений, примем время выполнения арифметической операции и операции присваивания, равным t_{base} . С учетом этого время выполнения алгоритма в худшем случае состоит из одной операции сравнения, одной операции присваивания времени получения координаты в переменную кеша, трех операций сравнения и одной операции вычитания в последнем условии. Итоговая верхняя оценка:

$$T_{2,2}^{max} = 6t_{base}. \quad (3.22)$$

Память, необходимая для алгоритма, определяется входными переменными tp и t . Для последней целесообразно использовать тип *long*, что достаточно для формата времени UNIX-Time. Оценка памяти:

$$E_{2,2} = e_{long} + e_{int}, \quad (3.23)$$

где e_{long} – объем памяти, необходимой для хранения числа типа *long*.

Вычислительная сложность алгоритма соответствует:

$$C_{2,2} = O(1). \quad (3.24)$$

3.2.7. Алгоритм фильтрации по времени

Цель фильтрации – уменьшение плотности точек трека за счет использования порога по времени и учета приоритета API источника перед NMEA. На Рисунке 3.10 представлен алгоритм работы фильтра.



Рисунок 3.10. Схема алгоритма работы фильтра по времени

В первом блоке алгоритма для координат с типом 1 (API) в переменную $P3_lastLocationTime$, которая соответствует параметру фильтра $P_{lastLocationTime}^3$ кеша координат и состояний сохраняется время получения координаты. Поскольку метод фильтрации, разработанный в главе 2, предполагает наличие у фильтра значения времени получения предыдущей координаты, будем представлять этот параметр переменной кеша $F3_prev_t$. После сохранения $P3_lastLocationTime$ фильтр проверяет наличие последней координаты в кеше и сравнивает

интервал времени между текущей и предыдущей координатами с переменной $P2_3_period$, которая соответствует параметру P_{period}^3 , тем самым уменьшая плотность координат в треке движения. Если координата соответствует этому условию, фильтр отдает предпочтение API координатам, анализируя признак стоянки $QisStand$ и тип координаты и сравнивая интервал времени между текущей координатой и временем получения последней API координаты с переменной $P2_3_locationPeriod$ (параметр фильтра $P_{locationPeriod}^3$). Если координата соответствует всем ограничениям блока обработки, то время ее получения сохраняется в переменную $F3_prev_t$ кеша.

Оценка времени выполнения алгоритма в лучшем случае определяется тремя операциями сравнения в первом и втором условных операторов алгоритма:

$$T_{2,3}^{min} = 3t_{base}. \quad (3.25)$$

Время выполнения алгоритма в худшем случае состоит из одной операции сравнения типа источника координаты, одной операции присваивания, двух операций сравнения и операции вычитания во втором блоке условий, трех операций сравнения и одной операции вычитания в последнем условии и операции присваивания. Итоговая верхняя оценка:

$$T_{2,3}^{max} = 10t_{base}. \quad (3.26)$$

Память, необходимая для алгоритма, определяется входными переменными tp и t . Оценка памяти:

$$E_{2,3} = e_{long} + e_{int}. \quad (3.27)$$

Вычислительная сложность алгоритма соответствует:

$$C_{2,3} = O(1). \quad (3.28)$$

3.2.8. Алгоритм фильтрации по скорости

Цель фильтрации – исключить координаты со скоростью, больше максимально возможной. На Рисунке 3.11 представлен алгоритм работы фильтра.

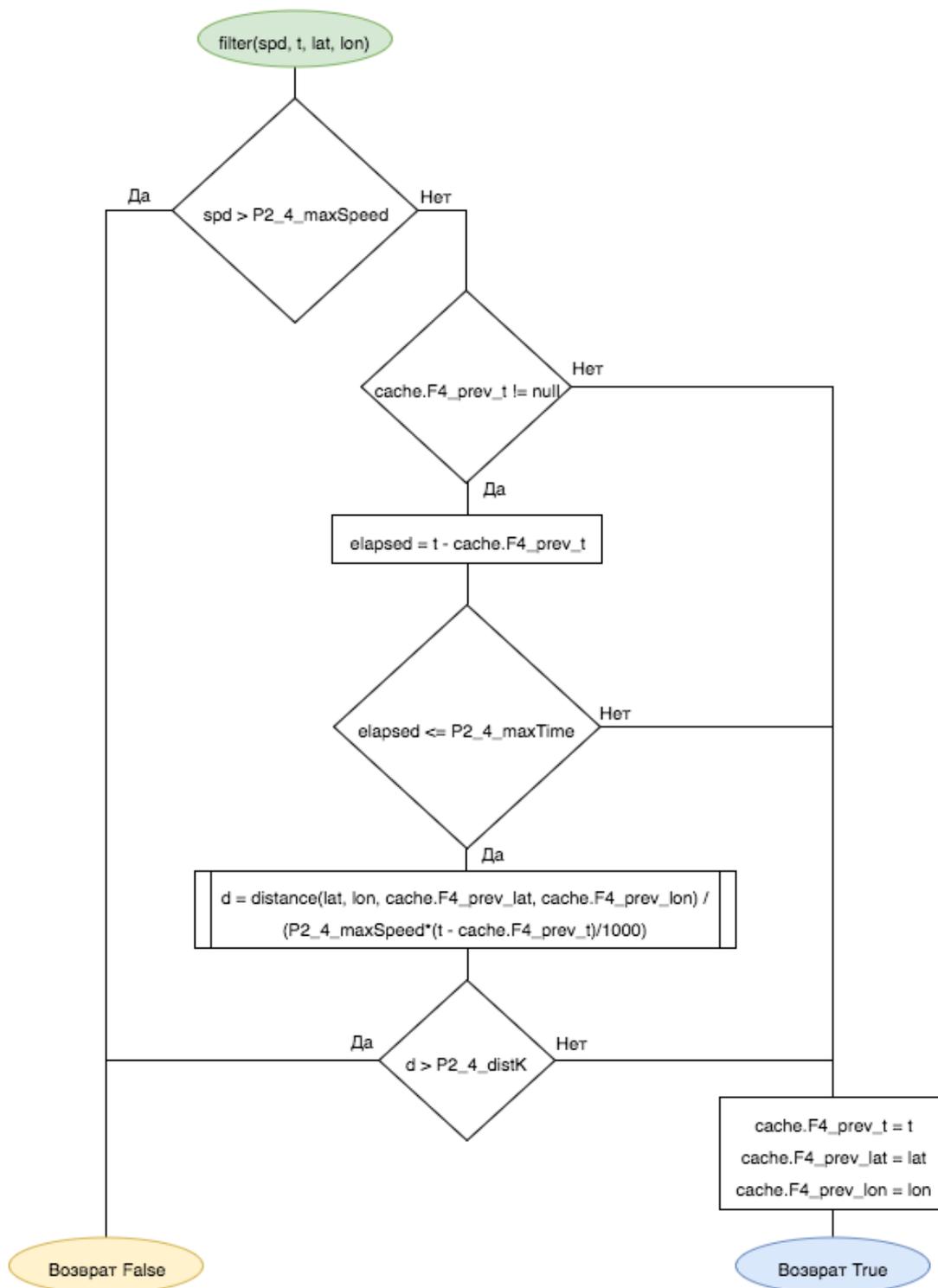


Рисунок 3.11. Схема алгоритма работы фильтра по скорости

Во-первых, фильтр не пропускает координаты с моментальной скоростью spd больше настроенного параметра $P2_4_maxSpeed$, соответствующего $P_{maxSpeed}^4$. Поскольку метод фильтрации, разработанный в главе 2, предполагает наличие у фильтра значений времени получения предыдущей координаты, ее широты lat и долготы lon будем представлять эти параметры переменными

$F4_prev_t$, $F4_prev_lat$ и $F4_prev_lon$ в кеше координат и состояний соответственно. Если эти значения присутствуют в кеше, алгоритм вычисляет значение интервала времени $elapsed$ между временем получения текущей координаты и $F4_Prev_t$. Значение сравнивается с переменной $P2_4_maxTime$, которая соответствует параметру фильтра $P_{maxTime}^4$. Далее вычисляется параметр d , который соответствует отношению расстояния между координатами к максимальному расстоянию, которое возможно преодолеть за интервал времени $elapsed$ со скоростью $P2_4_maxSpeed$, соответствующей параметру $P_{maxSpeed}^4$. Функцию $distance$, определяющая дистанцию между координатами с широтой lat и $F4_Prev_lat$ и долготой lon и $F4_Prev_lon$ соответственно, возможно реализовать по методу [50].

Параметр d сравнивается со значением переменной $P2_4_distK$, которая соответствует параметру модели фильтра P_{distK}^4 . Значения параметров координат сохраняются в кеше при условии неуспешного сравнения.

Оценка времени выполнения алгоритма в лучшем случае определяется одной операцией сравнения скорости с максимально возможной:

$$T_{2,4}^{min} = t_{base}. \quad (3.29)$$

Время выполнения алгоритма в худшем случае состоит из трех операций сравнения в первых условных блоках, одной операции вычитания, операции присваивания, алгоритма расчета значения переменной d , одной операций сравнения и трех операций присваивания в последнем блоке алгоритма. Помимо четырех арифметических операций и операции присваивания в расчете d , в худшем случае функция $distance$ выполняется за $2223t_{base}$. Итоговая верхняя оценка:

$$T_{2,4}^{max} = 2236t_{base}. \quad (3.30)$$

Помимо того, что фильтру необходима память для входных переменных t , lat , lon , spd , в алгоритме используются переменные $elapsed$ и d . Переменные spd , $elapsed$ и d следует определить типом $double$. Дополнительно для вычисления расстояния между двумя точками по их широте и долготе необходимо порядка 33 $double$ и 2 int чисел. Оценка памяти для фильтра:

$$E_{2,4} = e_{long} + 38e_{double} + 2e_{int}, \quad (3.31)$$

где e_{double} – объем памяти, необходимый для хранения переменной типа *double*.

Вычислительная сложность алгоритма соответствует:

$$C_{2,4} = O(1). \quad (3.32)$$

3.2.9. Алгоритм фильтрации по ускорению

Цель фильтрации – исключение координат с ускорением, больше максимально возможного. На Рисунке 3.12 представлен алгоритм работы фильтра.

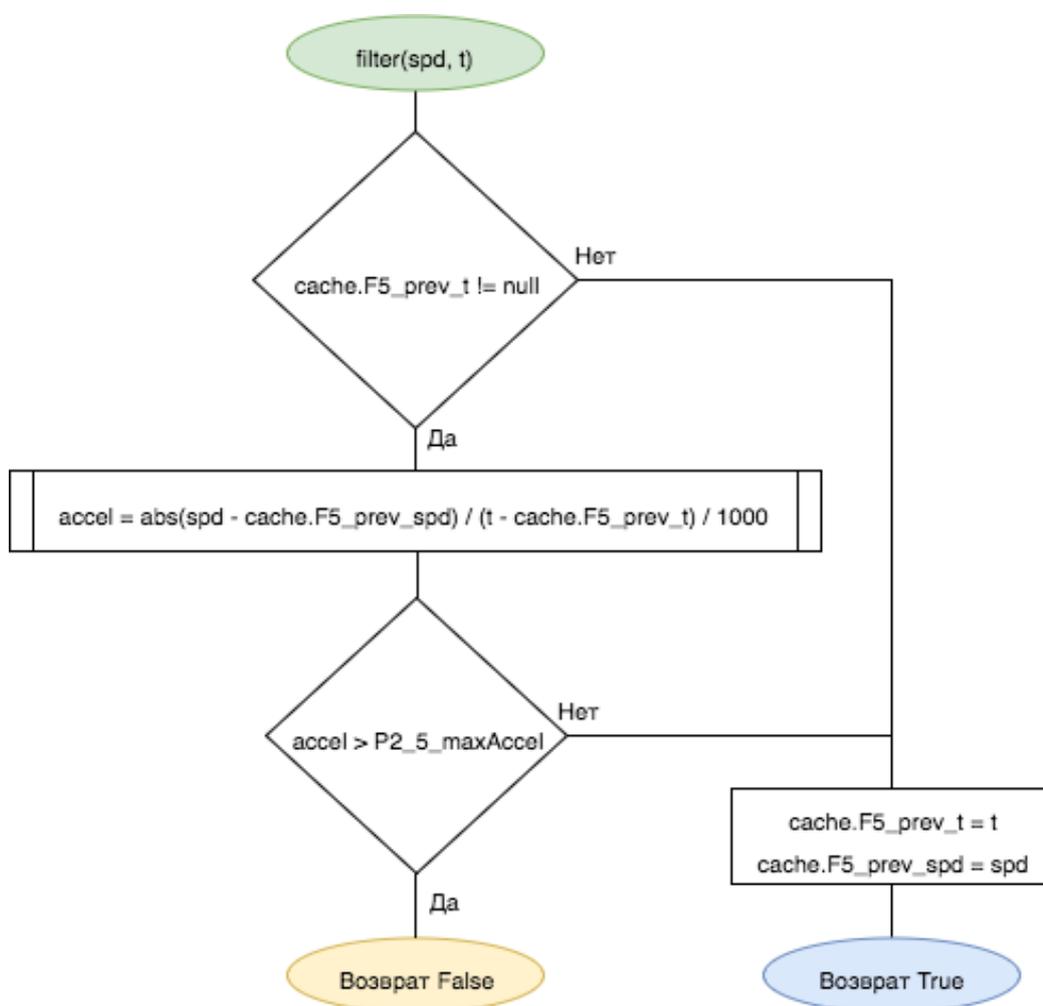


Рисунок 3.12. Схема алгоритма работы фильтра по ускорению

Метод фильтрации предполагает наличие у фильтра значений времени получения предыдущей координаты и ее скорости. Будем представлять эти параметры переменными $F5_prev_t$ и $F5_prev_spд$ в кеше координат и состояний соответственно. Если эти параметры присутствуют в кеше, алгоритм рассчитывает

ускорение $accel$ по скоростям и времени получения текущей и предыдущей координаты. Функция abs возвращает модуль числа. Полученное ускорение сравнивается с переменной $P2_5_maxAccel$, соответствующей параметру фильтра $P_{maxAccel}^5$. В случае успешного сравнения, параметры координаты сохраняются в кеш.

Оценка времени выполнения алгоритма в лучшем случае определяется одной операцией сравнения и записью двух значений в кеш:

$$T_{2,5}^{min} = 2t_{base}. \quad (3.33)$$

Время выполнения алгоритма в худшем случае соответствует времени выполнения всего алгоритма в лучшем случае плюс времени выполнения алгоритма вычисления и сравнения значения $accel$, который состоит из 4 арифметических операций, операции сравнения и операции взятия модуля, время которой примем за t_{base} . Итоговая верхняя оценка:

$$T_{2,5}^{max} = 8t_{base}. \quad (3.34)$$

Объем необходимой памяти определяется входными переменными t и spd и переменной $accel$. Для spd и $accel$ будем использовать $double$ числа. Необходимая память для фильтра:

$$E_{2,5} = e_{long} + 2e_{double}. \quad (3.35)$$

Вычислительная сложность алгоритма соответствует:

$$C_{2,5} = O(1). \quad (3.36)$$

3.2.10. Алгоритм фильтрации стоянок

Цель фильтрации – выявление состояний стоянок пользователя и исключение существенных выбросов и девиаций координат. На Рисунке 3.13 представлен алгоритм работы фильтра.

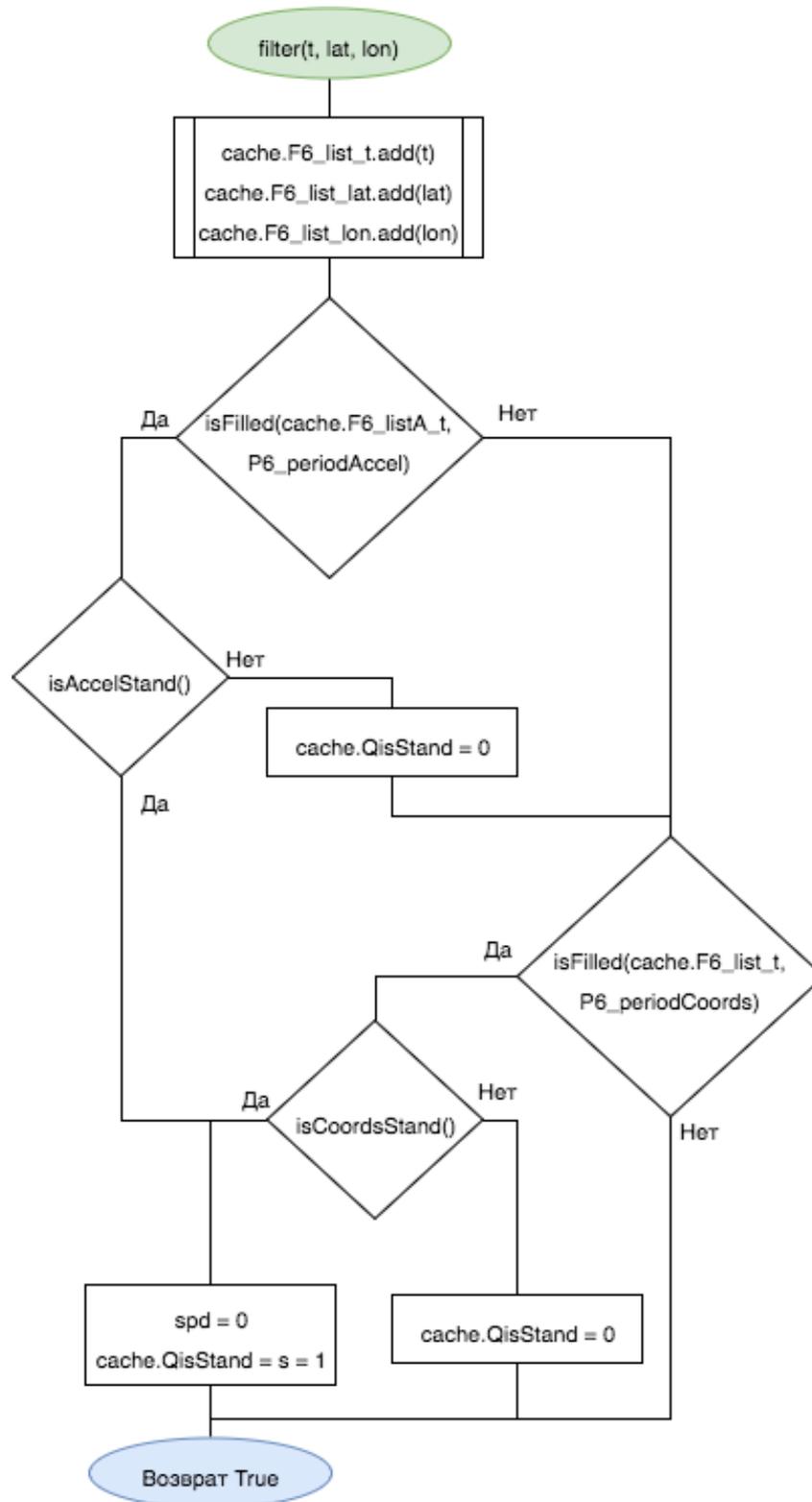


Рисунок 3.13. Схема алгоритма работы фильтра стоянок

Метод определяет признак стоянки для очередной координаты по дисперсиям скорости и ускорения. Для этого в первом блоке алгоритма параметры времени, долготы и широты текущей координаты сохраняются в соответствующие списки *F6_list_t*, *F6_list_lat* и *F6_list_lon* кеша координат и состояний. Данные

ускорений по осям x , y и z и времени их получения из акселерометра мобильного устройства сохраняются в соответствующих списках $F6_listA_ax$, $F6_listA_ay$, $F6_listA_az$, $F6_listA_t$ кеша асинхронно относительно выполнения функции `filter`.

Функция `isFilled` в первом блоке условий проверяет наличие ускорений в списках и возвращает `True`, если в списках гарантированно присутствуют данные об ускорении за период $P6_periodAccel$, который соответствует параметру $P6_periodAccel$ фильтра. Только в этом случае возможен дальнейший анализ дисперсии функцией `isAccelStand`, алгоритм которой показан на Рисунке 3.14.

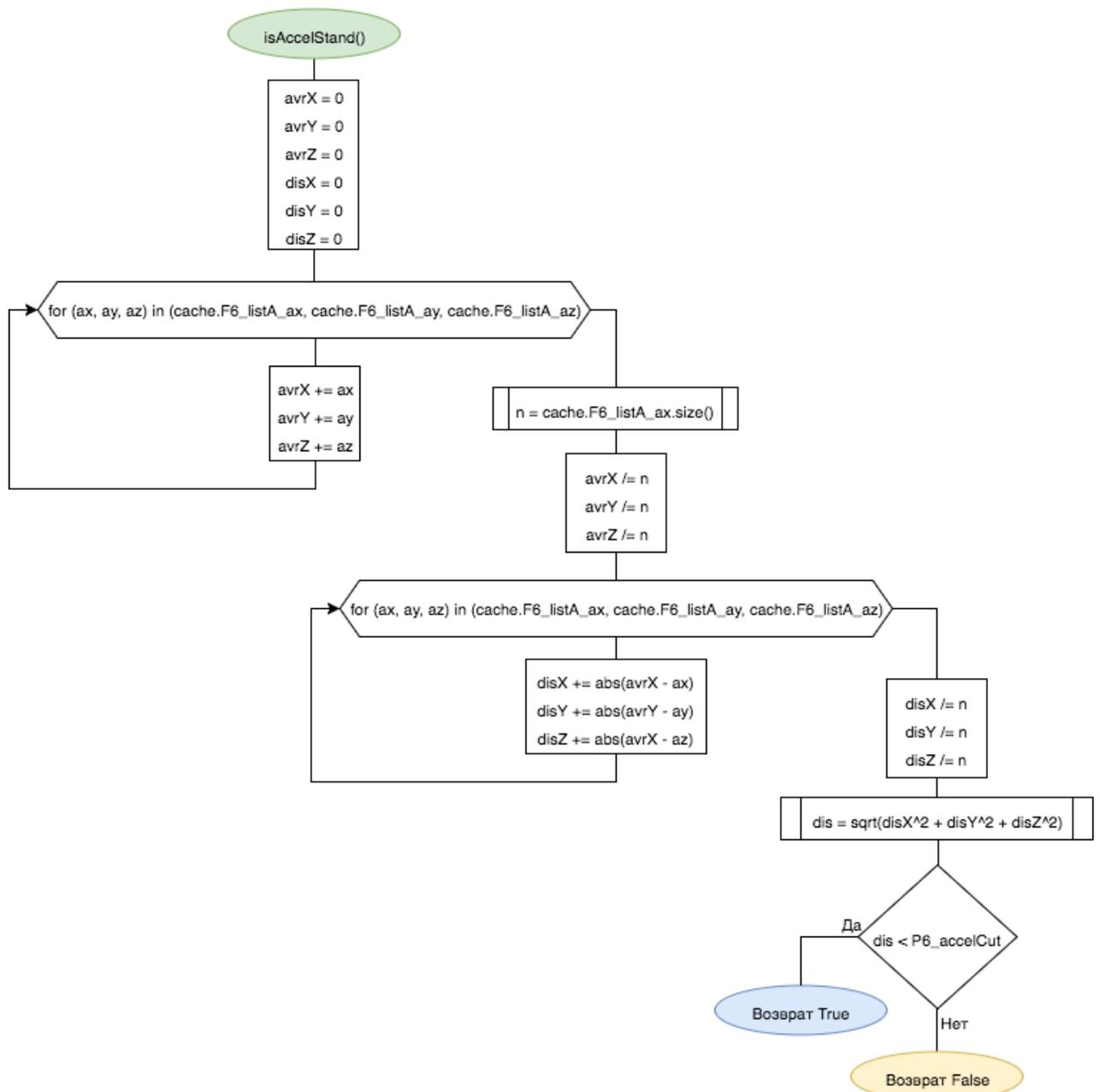


Рисунок 3.14. Схема алгоритма функции `isAccelStand`

В первом блоке алгоритма устанавливаются начальные значения параметров средних значений и дисперсий ускорений по всем осям. Вычисление значений происходит далее в циклах. В последних блоках алгоритма вычисляется дисперсия ускорения как квадратный корень из суммы квадратов дисперсий по каждой оси и сравнивается со значением переменной $P6_accelCut$, которое соответствует параметру фильтра $P6_{accelCut}$.

Если алгоритму не удалось определить стоянку по данным ускорения, выполняется аналогичный анализ дисперсии скорости по времени, долготе и широте координат в списках кеша с проверкой их полноты по времени $P6_periodCoords$, которое соответствует параметру $P6_{periodCoord}$. Алгоритм функции $isCoordsStand$, определяющей признак стоянки по дисперсии скорости, представлен на Рисунке 3.15.

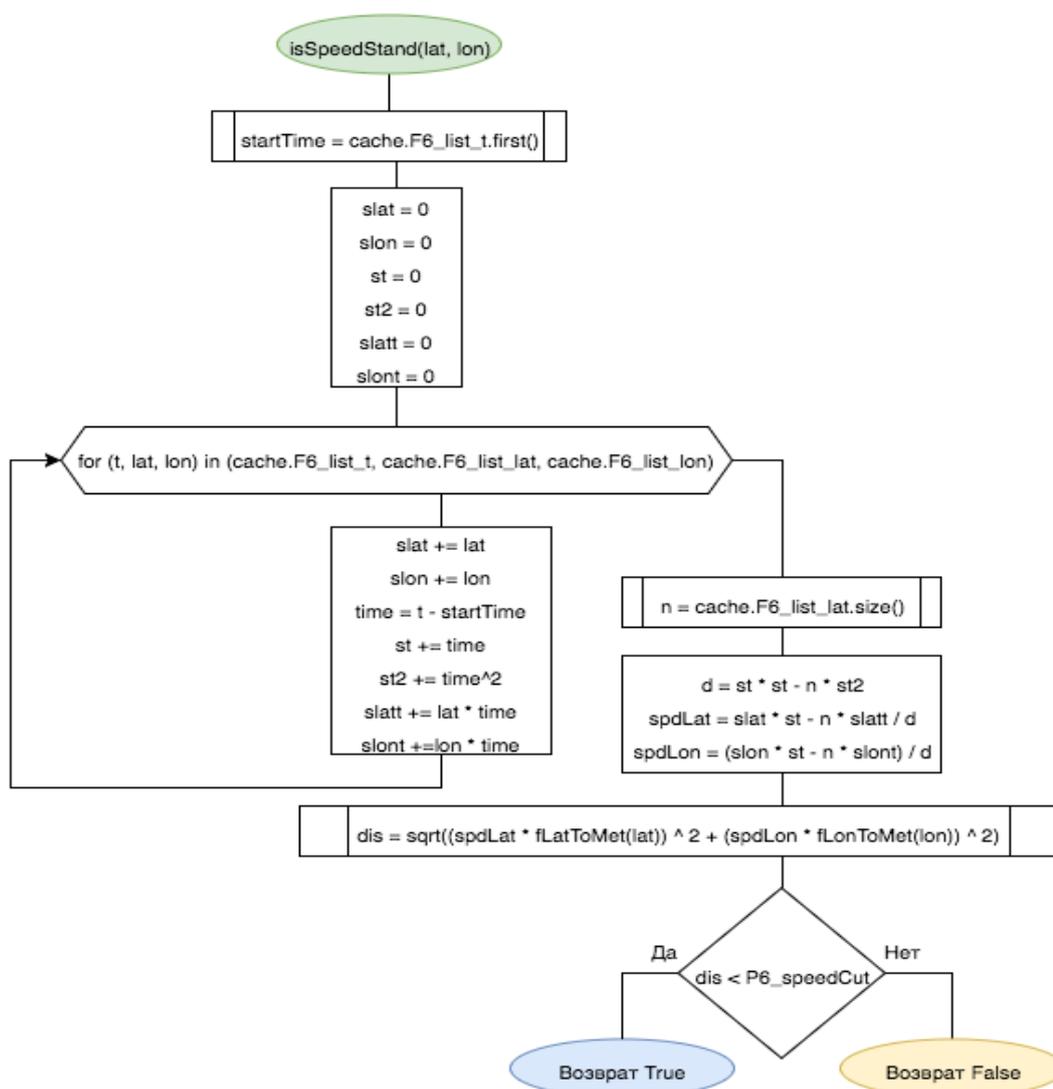


Рисунок 3.15. Схема алгоритма функции $isCoordsStand$

Алгоритм определяет дисперсию скорости аналогично вычислению дисперсии ускорения. Итоговое значение сравнивается с переменной $P6_speedCut$, которая соответствует параметру фильтра $P_{speedCut}^6$.

Итоговый признак стоянки формируется как логическое *ИЛИ* признака стоянки по дисперсии ускорения и признака стоянки по дисперсии скорости. Если очередная координата – стоянка, значения скорости и признака стоянки текущей координаты модифицируются соответствующим образом. Также стоит также отметить, что алгоритм вычисляет значение параметра $QisStand$ и возвращает *True* без учета результатов анализа дисперсий.

Оценка времени выполнения алгоритма в лучшем случае состоит из времени выполнения трех операций добавления параметров координат в список, которые примем за $3t_{base}$, двух функций $isFilled$, которые в случае отсутствия координат в списках, выполняются за $2t_{base}$. Итоговая оценка в лучшем случае:

$$T_{2,6}^{min} = 5t_{base}. \quad (3.37)$$

Время выполнения алгоритма в худшем случае состоит из времени выполнения трех операций добавления параметров в список, четырех операций присваивания, двух функций $isFilled$, которые выполняются за $10t_{base}$, функции $isAccelStand$ и $isCoordsStand$.

Функция $isAccelStand$ помимо циклических блоков состоит из 14 операций присваивания, 12 арифметических операций, одной операции сравнения и функции взятия числа элементов в списке, время выполнения которой примем за t_{base} . Первый блок цикла алгоритма содержит 3 операции сложения и 3 операции присваивания, второй цикл – 9 арифметических операций и 3 операции присваивания, количество итераций во всех случаях зависит от максимального числа элементов в списке, задаваемом параметром $P_{cacheSize}^{0'}$.

Функция $isSpeedStand$ помимо циклического блока состоит из 12 операций присваивания, 16 арифметических операций, одной операции сравнения, функции взятия первого элемента списка, время выполнения которой примем за t_{base} , функции взятия числа элементов в списке, время выполнения которой примем

также за t_{base} , двух функций $fLatToMet$ и $fLonToMet$, которые выполняются за $19t_{base}$ и $18t_{base}$ соответственно. Блок цикла алгоритма содержит 7 операций присваивания и 10 арифметических операций, количество итераций аналогично соответствует предыдущему случаю.

Итоговая оценка в худшем случае:

$$T_{2,6}^{max} = (113 + 35P_{cacheSize}^{0'})t_{base}. \quad (3.38)$$

Объем необходимой памяти определяется входными переменными t , lat , lon , а также переменными для расчета дисперсий ускорения и скорости. Необходимая память для фильтра:

$$E_{2,6} = 2e_{long} + 20e_{double} + 2e_{int}. \quad (3.39)$$

Вычислительная сложность алгоритма соответствует:

$$C_{2,6} = O(P_{cacheSize}^{0'}). \quad (3.40)$$

3.2.11. Алгоритм сглаживающей фильтрации

Цель блока обработки – сглаживание ломаной трека на основе экспоненциального скользящего среднего. На Рисунке 3.16 представлен алгоритм работы фильтра.

Метод фильтрации предполагает наличие у фильтра значений широты и долготы предыдущей координаты. Будем представлять эти параметры переменными $F7_prev_lat$ и $F7_prev_lon$ в кеше координат и состояний соответственно. Если эти параметры присутствуют в кеше, фильтр вычисляет расстояние между координатами функцией $distance$ и сохраняет его в переменную $dist$. Значение переменной сравнивается с максимальным расстоянием между точками $P2_7_maxSmoothDistance$ (параметр $P_{maxSmoothDistance}^7$). При успешном сравнении широта и долгота текущей координаты модифицируется в соответствии с предыдущими значениями и значением $P2_7_smoothBase$ (параметр $P_{smoothBase}^7$).

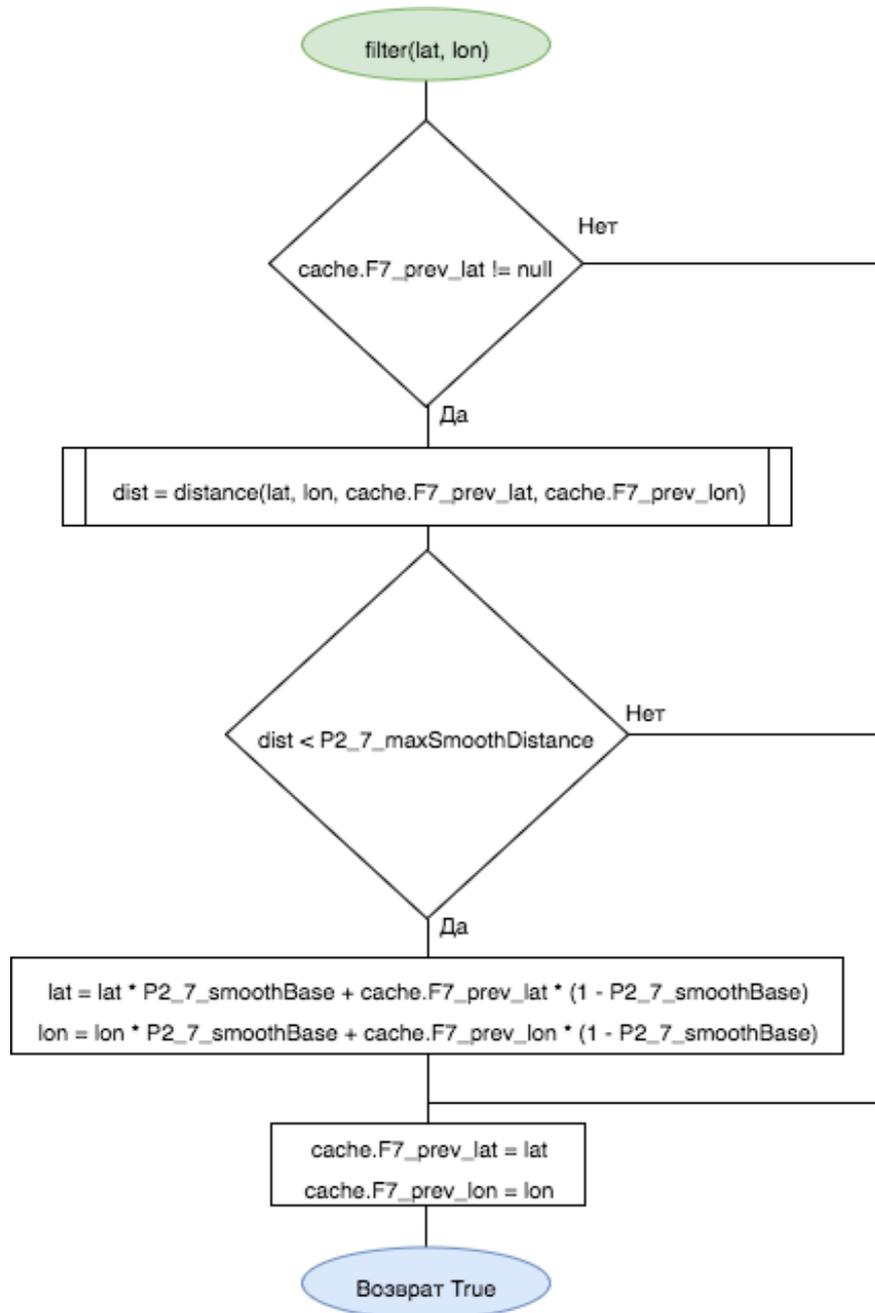


Рисунок 3.16. Схема алгоритма работы сглаживающего фильтра

Широта и долгота текущей координаты в последнем блоке записываются в кеш. Фильтр никогда не отбрасывает координату, всегда возвращая значение *True*.

Оценка времени выполнения алгоритма в лучшем случае определяется одной операцией сравнения записью двух значений в кеш:

$$T_{2,7}^{min} = 2t_{base}. \quad (3.41)$$

Время выполнения алгоритма в худшем случае есть время выполнения алгоритма в лучшем случае плюс расчет и присваивание расстояния между координатами, сравнение расстояния с параметром *P2_7_maxSmoothDistance* и

собственно сглаживание, состоящее из восьми арифметических операций и двух операций присваивания. Итоговая верхняя оценка:

$$T_{2,7}^{max} = 2236t_{base}. \quad (3.42)$$

Необходимая память для работы фильтра определяется входными переменными lat , lon , переменной $dist$ и другими переменными, необходимыми для расчета расстояния между двумя точками. Для $dist$ будем использовать число типа $double$. Оценка памяти:

$$E_{2,7} = 34e_{double} + 2e_{int}. \quad (3.43)$$

Вычислительная сложность алгоритма соответствует:

$$C_{2,7} = O(1). \quad (3.44)$$

3.2.12. Алгоритм угловой фильтрации

Цель фильтрации – уменьшение плотности точек трека, основанное на анализе угла между направлениями движения по трем координатам. На Рисунке 3.17 представлен алгоритм работы фильтра.

Метод фильтрации предполагает наличие у фильтра значений широты, долготы, скорости двух предыдущих координат и времени предыдущей координаты. Будем представлять эти параметры переменными $F8_prev_lat$, $F8_prev_lon$, $F8_prev_spd$, $F8_prev_prev_lat$, $F8_prev_prev_lon$, $F8_prev_prev_spd$ и $F8_prev_t$ в кеше координат и состояний соответственно. Первый условный блок проверяет существование параметров предыдущей координаты в кеше и сравнивает широту и долготу предыдущей координаты со значениями текущей. Если значения совпадают, то проводить дальнейший анализ угла предложенным методом не имеет смысла. Второй условный блок проверяет существование значений двух предыдущих координат одновременно в кеше, если координаты еще не получены, следует записать текущую координату кеш и вернуть управление со значением $True$, так как на данный момент нет достаточно данных для анализа углов между направлениями движений по трем координатам в последовательности. Иначе происходят вычисления:

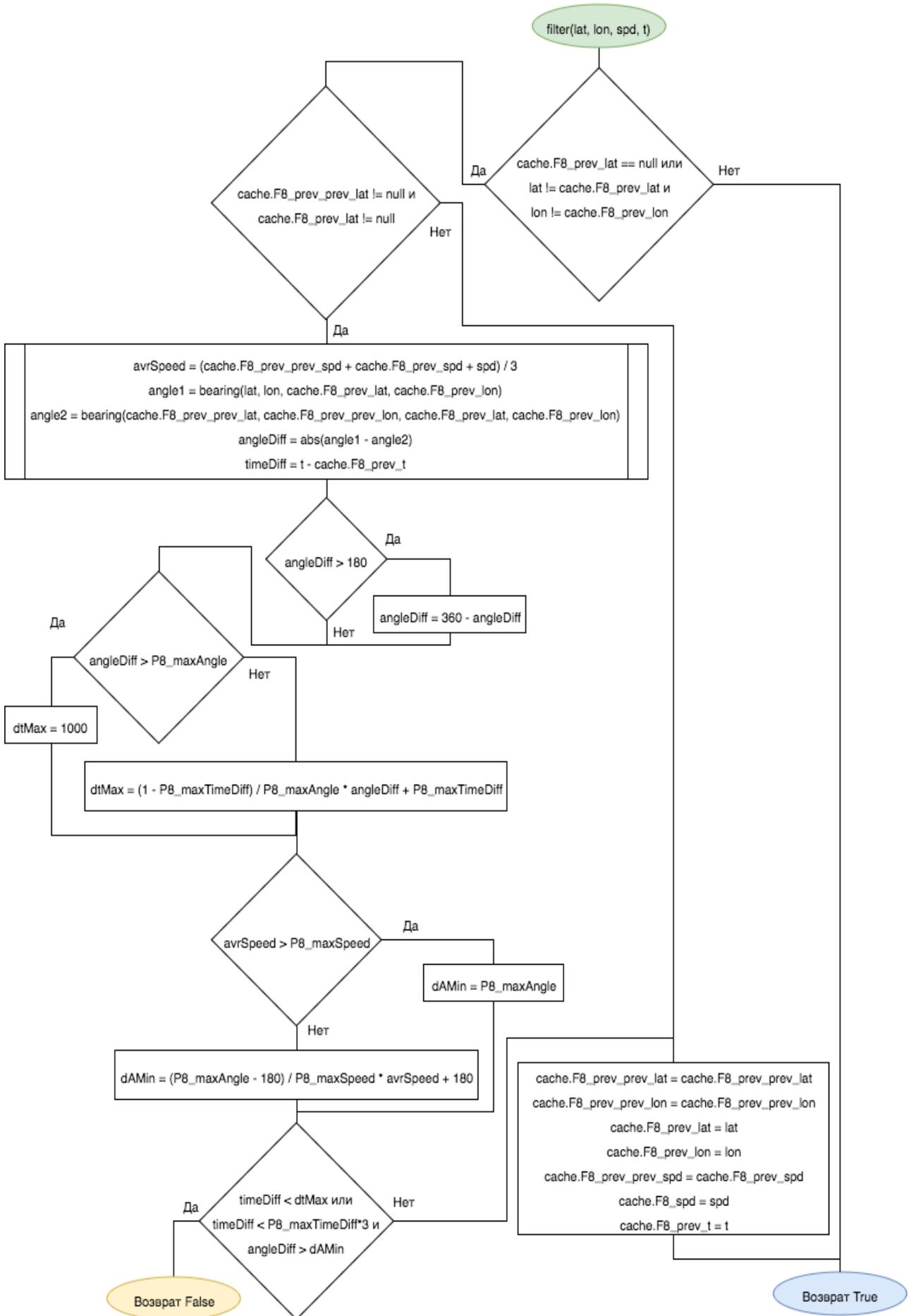


Рисунок 3.17. Схема алгоритма работы углового фильтра

- средней скорости $avrSpeed$ по трем координатам;
- углов направлений между первой и второй $angle1$, второй и третьей $angle2$ координатой. Здесь для вычисления используется функция $bearing$, принимающая широту и долготу двух координат в качестве входных параметров и возвращающая угол направления движения в градусах;
- угла $angleDiff$, соответствующего углу α – разнице между направлениями движения, значение переменной приводится к отрезку $[0; 180]$ градусов;
- интервала времени $timeDiff$ между текущей и предыдущей координатами.

Далее вычисляются значения $dtMax$ и $dAMin$ по формулам (2.20) и (2.23) с учетом параметров $P8_maxAngle$, $P8_maxTimeDiff$ и $P8_maxSpeed$, которые соответствуют параметрам $P8_{maxTimeDiff}^8$, $P8_{maxAngle}^8$ и $P8_{maxSpeed}^8$. Последний блок условия в алгоритме определяет критерий фильтрации.

Оценка времени выполнения алгоритма в лучшем случае состоит из времени выполнения двух операций сравнения в первом блоке условий алгоритма:

$$T_{2,8}^{min} = 2t_{base}. \quad (3.45)$$

Время выполнения алгоритма в худшем случае определяется:

- тремя операциями сравнения в первом блоке условий алгоритма;
- двумя – во втором;
- тремя арифметическими операциями и одной операцией присваивания при расчете значения переменной $avrSpeed$;
- временем выполнения алгоритмов функции $bearing$ вычисления угла по координатам, что соответствует $1317t_{base}$;
- одной операцией вычитания, взятия модуля и присваивания при расчете значения переменной $angleDiff$;
- операцией вычитания и присваивания при определении значения переменной $timeDiff$;
- операцией сравнения, операцией вычитания и присваивания при корректировке значения переменной $angleDiff$;

- двумя операциями сравнения, 8 арифметическими операциями и двумя операциями присваивания при расчете параметров $dtMax$ и $dAMin$;
- двумя операциями сравнения в последнем блоке условий;
- 7 операциями присваивания при сохранении параметров в кеш.

Итоговая верхняя оценка:

$$T_{2,8}^{max} = 3991t_{base}. \quad (3.46)$$

Необходимая память для работы фильтра определяется входными переменными $lat, lon, spd, t, avrSpeed, angle1, angle2, angleDiff, timeDiff, dtMax, dAMin$ и другими переменными, необходимыми для расчета угла направления движение по двум точкам. Для $avrSpeed, angle1, angle2, angleDiff, dAMin$ будем использовать числа типа $double$, для $timeDiff, dtMax$ – $long$. Оценка памяти:

$$E_{2,8} = 2e_{int} + 3e_{long} + 36e_{double}. \quad (3.47)$$

Вычислительная сложность алгоритма соответствует:

$$C_{2,8} = O(1). \quad (3.48)$$

3.2.13. Алгоритм диагностической фильтрации

Цель фильтрации – исключение отклонений координат в местах стоянок. На Рисунке 3.18 представлен алгоритм работы фильтра.

Метод диагностической фильтрации предполагает наличие у фильтра значений долготы, широты, скорости, точности, признака стоянки и типа источника. Будем представлять эти параметры переменными $F9_prev_lat, F9_prev_lon, F9_prev_spd, F9_prev_acc, F8_prev_s$ и $F8_prev_tp$ в кеше координат и состояний соответственно. Первый условный блок фильтра проверяет существование значений предыдущей координаты в кеше. Если значения не существует, дальнейший анализ проводить не имеет смысла. Иначе фильтр вычисляет значение переменной $dist$ по формулам (2.26) и (2.27), используя функции $facc$ и fk и переменные $P9_K$ и $P9_S$, соответствующие параметрам P_K^9 и P_S^9 .

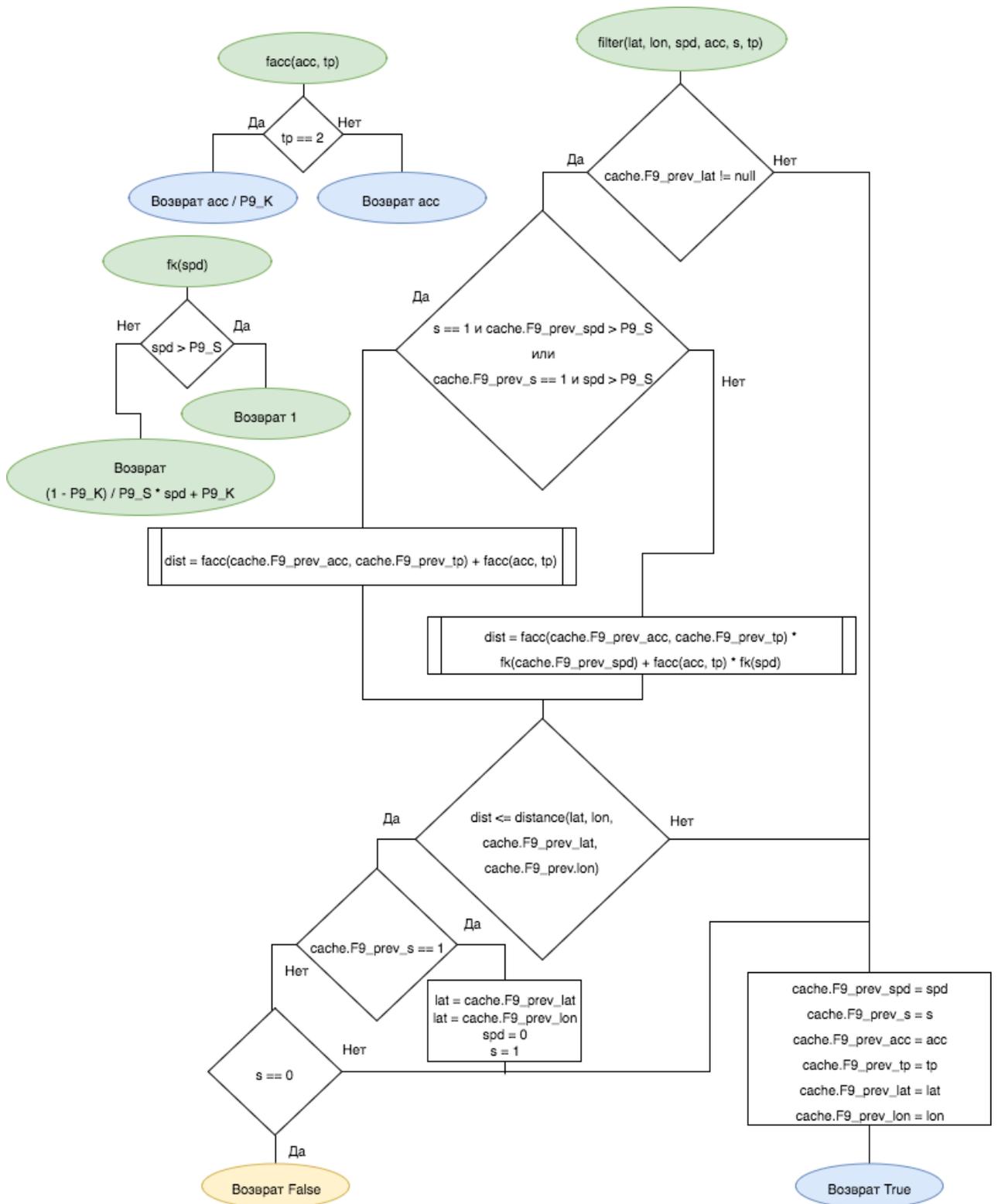


Рисунок 3.18. Схема алгоритма работы диагностического фильтра

Далее значение *dist* сравнивается с расстоянием между смежными точками, если условие выполняется, анализируются признаки стоянок текущей и предыдущей координаты. Если предыдущая координата – стоянка, то значения широты, долготы, скорости и признака стоянки текущей координаты модифицируются в

предположении, что текущая точка – продолжение стоянки. Иначе анализируется признак стоянки текущей координаты. Если текущая точка стоянка, фильтр сохраняет в кеш необходимые параметры текущей координаты и возвращает *True*. В противном случае координата исключается алгоритмом.

Оценка времени выполнения алгоритма в лучшем случае определяется одной операцией сравнения в первом блоке условий алгоритма и 6 операциями присваивания:

$$T_{2,9}^{min} = 7t_{base}. \quad (3.49)$$

Время выполнения алгоритма в худшем случае определяется одной операцией сравнения в первом блоке условий алгоритма, двумя операциями сравнения во втором блоке, 4 операциями сравнения, одной присваивания и 13 арифметическими операциями при расчете значения переменной *dist*, временем выполнения алгоритма нахождения расстояния между двумя точками, операцией сравнения найденного расстояния, операцией сравнения признака стоянки и 8 операциями присваивания. Итоговая верхняя оценка:

$$T_{2,9}^{max} = 2254t_{base}. \quad (3.50)$$

Необходимая память для работы фильтра определяется входными переменными *lat, lon, spd, acc, s, tp*, переменной *dist*, и другими переменными, необходимыми для расчета угла направления движение по двум точкам. Для переменной *dist* будем использовать число типа *double*, для *s* – *boolean*. Оценка памяти:

$$E_{2,9} = 3e_{int} + e_{boolean} + 37e_{double}. \quad (3.51)$$

Вычислительная сложность алгоритма соответствует:

$$C_{2,9} = O(1). \quad (3.52)$$

3.2.14. Алгоритм работы кеша координат и состояний

В соответствии с выбранным подходом реализации использования дополнительной информации связи алгоритм работы кеша координат и состояний

должен предусматривать сброс всех значений сохраняемых вычисленных значений параметров фильтров. Алгоритм сброса представлен на Рисунке 3.19.

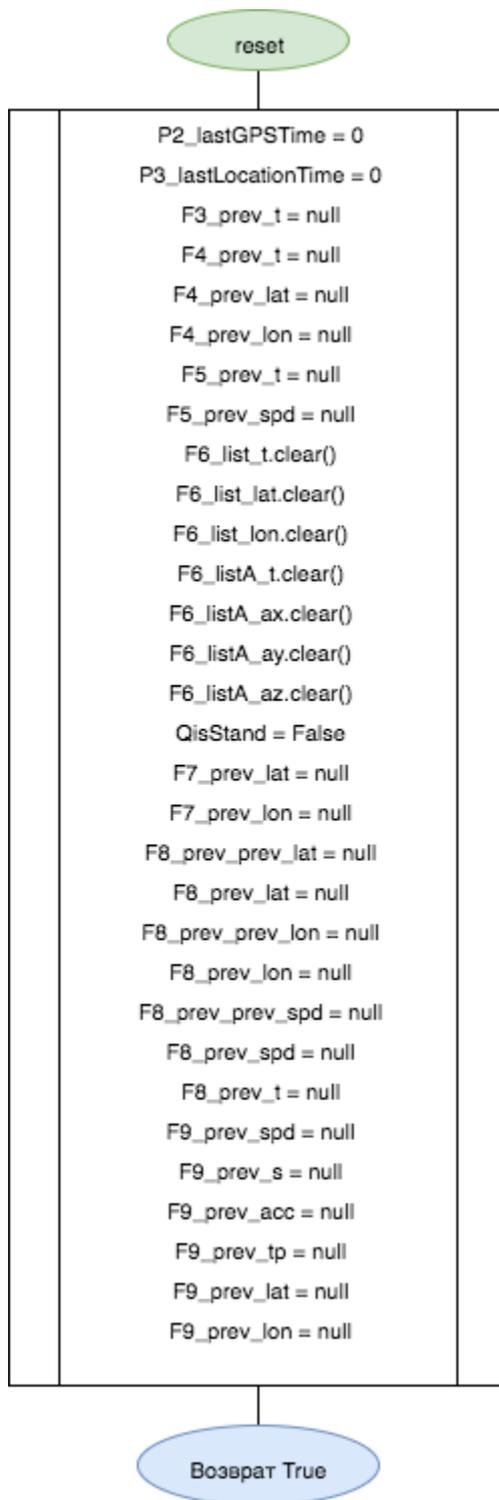


Рисунок 3.19. Схема алгоритма работы кеша координат и состояний
 Значения переменных сбрасываются в начальные значения *null* или 0. Для параметров *F6_list** предусмотрены функции *clear*, удаляющие все элементы из списков.

Для оценки необходимой памяти определим типы данных для хранения параметров координат и ускорений (см. Таблицу 8).

Таблица 8.

Типы данных параметров координат и ускорений

№	Параметр	Тип
1	<i>t</i>	<i>long</i>
2	<i>lat</i>	<i>double</i>
3	<i>lon</i>	<i>double</i>
4	<i>hdop</i>	<i>double</i>
5	<i>acc</i>	<i>double</i>
6	<i>spd</i>	<i>double</i>
7	<i>sat</i>	<i>int</i>
8	<i>tp</i>	<i>int</i>
9	<i>s</i>	<i>boolean</i>
10	<i>stype</i>	<i>int</i>
11	<i>isMocked</i>	<i>boolean</i>
12	<i>ax</i>	<i>double</i>
13	<i>ay</i>	<i>double</i>
14	<i>az</i>	<i>double</i>

В Таблице 9 представлен объем памяти для хранения значений одинаковых типов.

Таблица 9.

Типы данных параметров координат и ускорений

№	Параметр	Объем памяти
1	<i>P2_lastGPSTime</i>	$2e_{long}(1 + P_{cacheSize}^0)$
2	<i>P3_lastLocationTime</i>	

Таблица 10 (продолжение).

№	Параметр	Объем памяти	
3	$F3_{prev_t}$		
4	$F4_{prev_t}$		
5	$F5_{prev_t}$		
6	$F8_{prev_t}$		
7	$F6_{list_t}$		
8	$F6_{listA_t}$		
9	$F4_{prev_lat}$		$e_{double}(15 + 6P_{cacheSize}^{0'})$
10	$F4_{prev_lon}$		
11	$F5_{prev_spd}$		
12	$F7_{prev_lat}$		
13	$F7_{prev_lon}$		
14	$F8_{prev_prev_lat}$		
15	$F8_{prev_lat}$		
16	$F8_{prev_prev_lon}$		
17	$F8_{prev_lon}$		
18	$F8_{prev_prev_spd}$		
19	$F8_{prev_spd}$		
20	$F9_{prev_spd}$		
21	$F9_{prev_acc}$		
22	$F9_{prev_lat}$		
23	$F9_{prev_lon}$		
24	$F6_{list_lat}$		
25	$F6_{list_lon}$		
26	$F6_{listA_ax}$		
27	$F6_{listA_ay}$		

№	Параметр	Объем памяти
28	$F6_listA_az$	
29	$QisStand$	$2e_{boolean}$
30	$F9_prev_s$	
31	$F9_prev_tp$	e_{int}

Итоговый объем памяти:

$$E_{cache} = 2e_{long}(1 + P_{cacheSize}^{0'}) + e_{double}(15 + 6P_{cacheSize}^{0'}) + 2e_{boolean} + e_{int}. \quad (3.53)$$

3.3. Оценка быстродействия фильтрации

Сведем полученные оценки необходимой памяти, времени выполнения в худшем и лучшем случаях и вычислительной сложности в Таблицу 10.

Таблица 10.

Оценка эффективности фильтрации

№	Фильтр	Оценка			
		Необходимой памяти	Времени выполнения		Вычислительной сложности
			В лучшем случае	В худшем случае	
1	Типов источников	e_{int}	t_{base}	t_{base}	$O(1)$
2	Типов спутников	$2e_{int}$	t_{base}	$2t_{base}$	$O(1)$
3	Ложных координат	$e_{boolean}$	t_{base}	$2t_{base}$	$O(1)$
4	По точности	$2e_{int} + 2e_{double}$	t_{base}	$5t_{base}$	$O(1)$
5	Сетевых координат	$e_{long} + e_{int}$	$2t_{base}$	$6t_{base}$	$O(1)$
6	По времени	$e_{long} + e_{int}$	$3t_{base}$	$10t_{base}$	$O(1)$

Таблица 13 (продолжение).

№	Фильтр	Оценка			
		Необходимой памяти	Времени выполнения		Вычислительной сложности
			В лучшем случае	В худшем случае	
7	По скорости	$e_{long} + 38e_{double} + 2e_{int}$	t_{base}	$2236t_{base}$	$O(1)$
8	По ускорению	$e_{long} + 2e_{double}$	$2t_{base}$	$8t_{base}$	$O(1)$
9	Стоянок	$2e_{long} + 20e_{double} + 2e_{int}$	$5t_{base}$	$(113 + 35P'_{cacheSize})t_{base}$	$O(P'^{0}_{cacheSize})$
10	Сглаживающий	$34e_{double} + 2e_{int}$	$2t_{base}$	$2236t_{base}$	$O(1)$
11	Угловой	$2e_{int} + 3e_{long} + 36e_{double}$	$2t_{base}$	$3991t_{base}$	$O(1)$
12	Диагностический	$3e_{int} + e_{boolean} + 37e_{double}$	$7t_{base}$	$2254t_{base}$	$O(1)$

Из анализа таблицы следует, что:

- 1) угловой фильтр и фильтр по скорости требуют наибольшее количество памяти: $e_{long} + 38e_{double} + 2e_{int}$ и $2e_{int} + 3e_{long} + 36e_{double}$ соответственно, с учетом того, что число *double* представляется 8 байтами, *int* – 4, *long* – 8, *boolean* – 1;
- 2) фильтры по скорости, стоянок, сглаживающий, угловой и диагностический имеют на порядок большее значение оценки необходимого объема памяти и времени выполнения в худшем случае, так как в их реализации используются алгоритмы вычисления дисперсии скорости, ускорения, значения расстояния между двумя и угла направления между тремя точками по их широте и долготе;
- 3) оценки времени выполнения в лучшем случае у всех алгоритмов малы и сравнимы;
- 4) при $P'^{0}_{cacheSize} < 111$ максимальное время выполнения составляет $3991t_{base}$ у алгоритма фильтрации по углу; при $P'^{0}_{cacheSize} \geq 111$ оценка составляет $(113 + 35P'^{0}_{cacheSize})t_{base}$ у алгоритма работы фильтра стоянок;

5) максимальная оценка вычислительной сложности принимает значение $O(P_{cacheSize}^{0'})$ у алгоритма фильтрации стоянок, все остальные фильтры имеют вычислительную сложность $O(1)$.

По результатам анализа можно выделить 2 группы фильтров:

- с низкими вычислительными и емкостными затратами – фильтр по типу источников, типу спутников, фильтр ложных координат, по точности, сетевых координат, по времени и ускорению;
- с высокими вычислительными и емкостными затратами – фильтр по скорости, стоянок, сглаживающий, угловой и диагностический.

Оценка необходимой памяти для работы общего алгоритма фильтрации без учета алгоритмов препроцессинга данных представляет собой сумму емкостных оценок всех алгоритмов фильтрации и кеша координат и состояний:

$$E = 4e_{boolean} + 27e_{int} + (11 + 2P_{cacheSize}^{0'})e_{long} + (184 + 6P_{cacheSize}^{0'})e_{double}. \quad (3.54)$$

Общий алгоритм фильтрации предполагает обработку координаты, пришедшей из источника, последовательностью фильтров. При этом алгоритм определяется циклическим выполнением функции *filter* с необходимыми входными параметрами текущей координаты до тех пор, пока результат функции не примет значение *False*. Номер итерации здесь фактически соответствует номеру фильтра в последовательности. В этом случае координата не появляется на выходе системы. Если все функции вернули значение *True*, координата поступает на выход системы: считается, что координата имеет достаточную точность для дальнейшей обработки в конечном мобильном приложении.

Одной из ключевых особенностей предложенного алгоритма является то, что координаты появляются на выходе даже в случае стоянки пользователя. Рассмотрим случай, когда исходные данные представляют собой набор координат, которые удовлетворяют всем требованиям фильтров, при этом начальная и конечная координаты являются координатами в движении, а середине последовательности соответствует продолжительная стоянка (см. Рисунок 3.20).

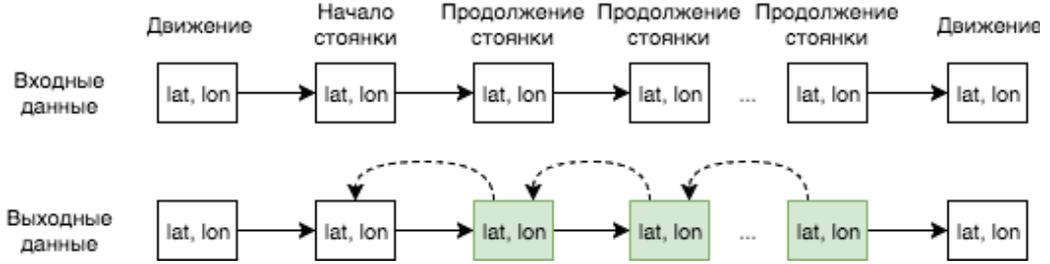


Рисунок 3.20. Особенность выдачи координат во время стоянки

Алгоритм фильтрации будет генерировать на выходе координату за координатой при поступлении данных на вход. При этом первая точка, продолжение стоянки, а также все дальнейшие до окончания стоянки по широте и долготе будут соответствовать координате начала стоянки, поскольку алгоритм последовательно модифицирует параметры широты и долготы. Такая особенность существенно уменьшает время отклика системы при отсутствии движения мобильного устройства.

Оценку времени выполнения общего алгоритма фильтрации в худшем и лучшем случаях можно найти, решая задачи оптимизации:

$$T_{max} = \sum_{j=1}^{p \leq n} T_j \rightarrow \max, \text{ при } F_{j-1}(\{x_i\}, \{P^{j-1}\}) = 1 \text{ и } T_{j-1} = T_{j-1}^+, \text{ если } j > 1; \quad (3.55)$$

$$T_{min} = \sum_{j=1}^{q \leq n} T_j \rightarrow \min, \text{ при } F_{j-1}(\{x_i\}, \{P^{j-1}\}) = 1 \text{ и } T_{j-1} = T_{j-1}^-, \text{ если } j > 1, \quad (3.56)$$

где T_{max} и T_{min} – оценка времени выполнения общего алгоритма фильтрации в худшем и лучшем случаях соответственно;

$n = 12$ – максимальное количество фильтров в последовательности;

$p \in \mathbb{N} > 0$ – количество фильтров в последовательности, оценка времени выполнения алгоритма фильтрации которой максимальна;

$q \in \mathbb{N} > 0$ – количество фильтров в последовательности, оценка времени выполнения алгоритма фильтрации которой минимальна;

T_j^+ – оценка времени выполнения алгоритма работы фильтра с порядковым номером j при $F_j(\{x_i\}, \{P^j\}) = 1$;

T_j^- – оценка времени выполнения алгоритма работы фильтра с порядковым номером j при $F_j(\{x_i\}, \{P^j\}) = 0$;

То есть оценка времени выполнения общего алгоритма фильтрации в худшем или лучшем случае будет являться максимумом или минимумом суммы

оценок времени выполнения алгоритмов в последовательности. При этом последовательность может не содержать все фильтры набора, она определяется только теми, при использовании которых функция достигает глобального максимума или минимума. Локальных оптимумов у функции T может быть много. Время выполнения каждого алгоритма в найденной последовательности кроме последнего должно быть рассчитано по позитивному (или негативному в случае определения T_{min}) сценарию, когда $F_j(\{x_i\}, \{P^j\}) = 1$, т.е. координата удовлетворяет требованиям фильтра. При этом значения T_j^+ в общем случае не равны временам выполнения алгоритмов в лучшем T_j^{min} или худшем T_j^{max} случаях.

Максимум функции T_{max} в предложенной алгоритме достигается в случае $p = 12$, при этом все оценки T_j^+ равны максимальному времени выполнения алгоритма T_j^{max} . Таким образом итоговая верхняя оценка времени выполнения будет соответствовать сумме всех T_j^{max} для максимального количества фильтров в последовательности:

$$T_{max} = \sum_{j=1}^{12} T_j^{max} = (10864 + 35P_{cacheSize}^{0'})t_{base}. \quad (3.57)$$

Оценка времени выполнения в лучшем случае у предложенного алгоритма будет ограничиваться набором всего одного первого фильтра в случае $F_1(\{x_i\}, \{P^1\}) = 0$:

$$T_{min} = \sum_{j=1}^1 T_j = T_1^- = t_{base}. \quad (3.58)$$

Рассмотрим пример, когда верхняя оценка не будет соответствовать сумме максимальных оценок. Допустим, алгоритм фильтра стоянок, который расположен в середине последовательности, способен отбрасывать координату в определенных условиях. После того как согласно алгоритму был вычислен признак того, что координату следует отбросить, определенные рассчитанные характеристики координаты и/или состояния фильтра сохраняются в кеш. При этом у алгоритмов расчетов настолько большая вычислительная сложность, что оценка времени их выполнения на порядок выше суммы всех верхних оценок алгоритмов последующих фильтров. В этом случае набор, максимизирующий целевую функцию, будет состоять из подмножества фильтров последовательности:

$$T_{max} = \sum_{j=1}^9 T_j = \sum_{j=1}^8 T_j^{max} + T_9^{min}. \quad (3.59)$$

Определение общей вычислительной сложности также основано на решении аналогичной задачи оптимизации, рассмотренной выше. Вычислительная сложность имеет максимум при количестве фильтров, равном 12, и соответствует:

$$C = O(P_{cacheSize}^{0'}). \quad (3.60)$$

Возможно также статистически оценить средние показатели времен выполнения каждого алгоритма работы фильтра в отдельности и фильтрации в целом, чтобы рассчитать требования к аппаратному обеспечению. Исходя из невысоких верхних оценок вычислительных сложностей алгоритмов и нетривиального характера задачи статистической обработки, в работе указанный анализ не выполнялся.

Выводы по главе 3

На основе результатов анализа источников геолокационных данных в мобильных операционных системах, специфики программной и аппаратной частей, с целью уточнения методов фильтрации и разработки алгоритма последовательной обработки потока геокоординат в реальном времени для исключения «выбросов», уменьшения плотности точек на прямых участках маршрута и сглаживания получаемого трека можно сделать следующие выводы:

- 1) необходимо использовать 3 источника данных: NMEA Receiver, Location API и Accelerometer API, которые позволят агрегировать информацию из источников, рассмотренных в главе 2;
- 2) следует дополнить методы фильтрации геолокационных данных блоком обработки ложных координат, типов источников и спутников. Предложенные методы существенно увеличат гибкость работы и настройки системы фильтрации на разных моделях мобильных устройств. Входные и выходные данные необходимо дополнить параметрами, позволяющими использовать уточняющие методы обработки;

- 3) использование дополнительной информации из конечного приложения в системе фильтрации необходимо организовать по типу триггера, что значительно уменьшит время отклика системы. Данный подход в сравнении с другими предложенными повысит точность фильтрации даже в случае, когда в конечном мобильном приложении нет достоверных сведений о местоположении пользователя;
- 4) в алгоритм фильтрации следует добавить блок кеша координат и состояний, который позволит системе восстанавливать свое состояние после перезагрузки устройства или отдельного приложения, а также настраивать параметры фильтров, например, число хранимых координат в фильтре стоянок, под размер общей памяти.

Рассмотренные подходы и методы фильтрации следует добавить в исходную модель отдельными блоками, не модифицируя базовые модули, предложенные в главе 2. В этом случае базовое ядро – комплексная модель потоковой фильтрации – будет самодостаточной подсистемой, ее можно будет использовать на разных платформах, в разном окружении, обеспечив необходимыми параметрами и исходными данными.

На основе уточненной расширенной модели был разработан алгоритм фильтрации геоданных в реальном времени, состоящей из 12 модулей обработки, эффективность которого будет оценена экспериментально в следующей главе. Были разработаны частные алгоритмы препроцессинга и фильтрации, определены оценки времени их выполнения в худшем и лучшем случаях, необходимой памяти для работы и вычислительной сложности. Разработан общий алгоритм фильтрации при поступлении на вход очередной координаты, выявлены его ключевые особенности. Проведен сравнительный анализ эффективности частных алгоритмов по рассчитанным показателям, из результатов которого следуют выводы:

- 1) угловой фильтр и фильтр по скорости требуют наибольшее количество памяти;

- 2) фильтры по скорости, стоянок, сглаживающий, угловой и диагностический имеют на порядок большее значение оценки необходимого объема памяти и времени выполнения в худшем случае, так как в их реализации используются алгоритмы вычисления дисперсии скорости, ускорения, значения расстояния между двумя и угла направления между тремя точками по их широте и долготе;
- 3) оценки времени выполнения в лучшем случае у всех алгоритмов малы и сравнимы;
- 4) при $P_{cacheSize}^{0'} < 111$ максимальное время выполнения составляет $3991t_{base}$ у алгоритма фильтрации по углу; при $P_{cacheSize}^{0'} \geq 111$ оценка составляет $(113 + 35P_{cacheSize}^{0'})t_{base}$ у алгоритма работы фильтра стоянок;
- 5) максимальная оценка вычислительной сложности принимает значение $O(P_{cacheSize}^{0'})$ у алгоритма фильтрации стоянок, все остальные фильтры имеют вычислительную сложность $O(1)$.

По результатам анализа можно выделить 2 группы фильтров:

- с низкими вычислительными и емкостными затратами – фильтр по типу источников, типу спутников, фильтр ложных координат, по точности, сетевых координат, по времени и ускорению;
- с высокими вычислительными и емкостными затратами – фильтр по скорости, стоянок, сглаживающий, угловой и диагностический.

На основе предложенного общего алгоритма фильтрации, частных алгоритмов и рассчитанных оценок эффективности была рассмотрена задача оптимизации, решение которой позволило определить оценки для общего алгоритма фильтрации при поступлении очередной координаты на вход:

- вычислительной сложности: $C = O(P_{cacheSize}^{0'})$;
- времени выполнения в лучшем случае: $T_{min} = t_{base}$;
- времени выполнения в худшем случае: $T_{max} = (10864 + 35P_{cacheSize}^{0'})t_{base}$.

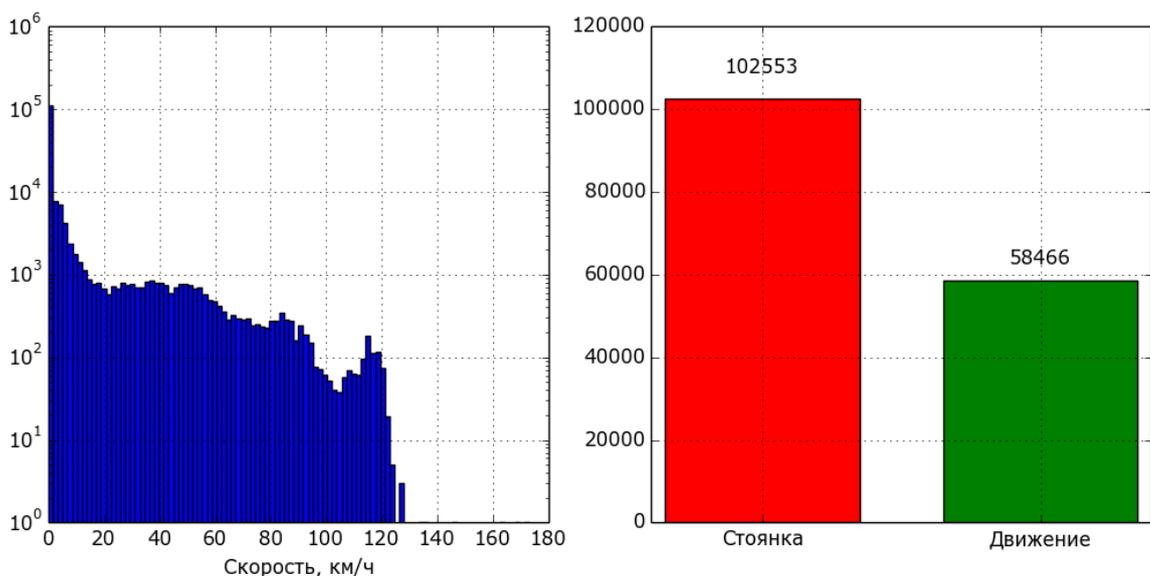
Глава 4. Разработка методов оценки эффективности фильтрации и экспериментальные исследования

Предложенные методы фильтрации и алгоритм обработки потока геоданных был реализована на языке Java. Базовые методы фильтрации, рассмотренные в главе 2, были имплементированы в рамках отдельной библиотеки, что позволило использовать их не только в Android-ориентированных программных продуктах. Общий алгоритм реализован библиотекой, включающей в себя частные алгоритмы фильтрации, препроцессинга и интерфейсы ввода и вывода геолокационной информации для применения в конечных Android приложениях.

4.1 Оценка эффективности алгоритма фильтрации потока геоданных

Для оценки качества результирующего трека, полученного в результате фильтрации исходных данных предложенным алгоритмом, с различных Android устройств были собраны треки пользователей, представляющие стоянки, движения пешком, на автомобиле, поезде, метро, велосипеде. Общее количество точек в треках порядка 1 млн. Из треков были выбраны те, в которых лучшим образом отражены проблемы выбросов и отклонений координат от реальных маршрутов движения. Общее количество треков в тестовом наборе данных – 24, число координат – 161019. Гистограмма распределения скорости и распределение стоянок и координат в движении представлены на Рисунке 4.1.

Среднее значение скорости $\mu = 8.647 \frac{\text{км}}{\text{ч}}$, $\sigma = 20.022 \frac{\text{км}}{\text{ч}}$, с учетом допуска в 3σ скорость распределена на отрезке $[0 \frac{\text{км}}{\text{ч}}; 68.714 \frac{\text{км}}{\text{ч}}]$. По рисунку видно, что существенную часть точек представляют собой координаты с незначительной моментальной скоростью. По количеству этих точек можно приближенно определить число координат, которые отражают стоянки пользователей. Принимая скорость в точке стоянки, равной нулю, количество координат-стоянок в наборе составляет 102553, точек в движении – 58466.



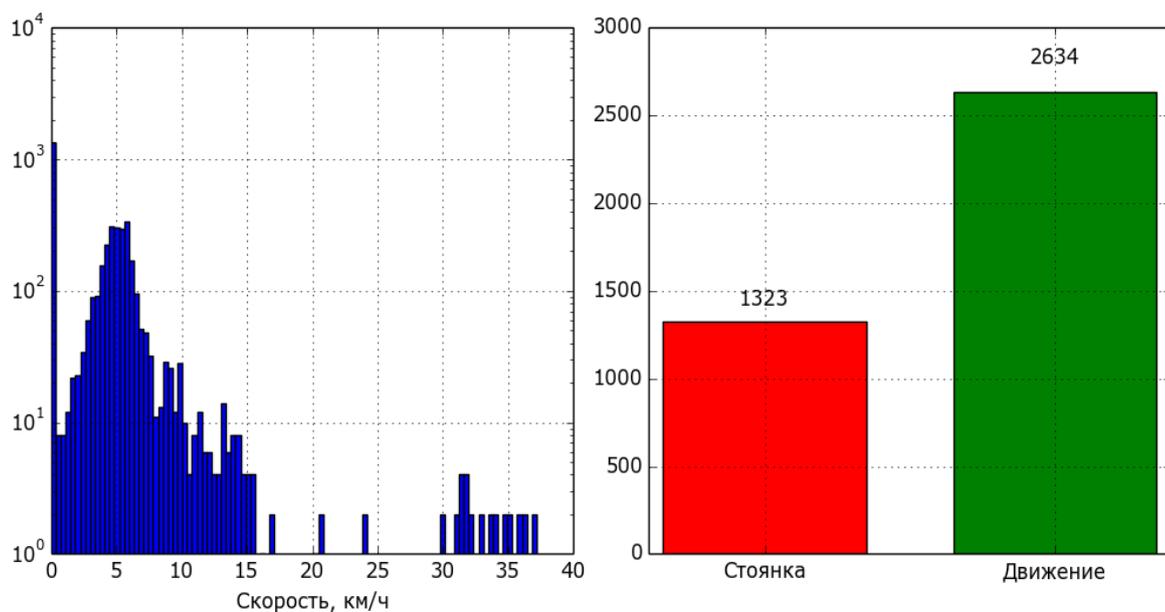
(а) распределение скорости

(б) распределение стоянок и координат
в движении

Рисунок 4.1. Распределение параметров координат в тестовом наборе данных

Эффективность предложенного алгоритма фильтрации будем оценивать в сравнении с существующими методами обработки геокоординат, рассмотренными в главе 1:

- 1) стандартным API геолокации операционной системы Android совместно с пороговым фильтром по точности *acc* без учета ограничений на параметры погрешности *hdop* и числа спутников *sat*. Пороговый параметр фильтрации P_{maxAcc}^1 примем равным 30 м. В качестве исходных данных будем использовать весь исходный тестовый набор данных. Результатом работы фактически будет являться подмножество исходных координат;
- 2) технологией Fused геолокации Android совместно с пороговым фильтром по точности, аналогичным фильтру в п. 1. Для анализа эффективности метода был собран отдельный набор данных, состоящий из 2 треков с общим количеством точек, равным 3957. Гистограмма распределения скорости и распределение стоянок и координат в движении представлены на Рисунке 4.2.



(а) распределение скорости (б) распределение стоянок и координат
в движении

Рисунок 4.2. Распределение параметров координат в тестовом наборе данных для оценки по технологии Fused геолокации

Среднее значение скорости $\mu = 3.831 \frac{\text{км}}{\text{ч}}$, $\sigma = 4.07 \frac{\text{км}}{\text{ч}}$, с учетом допуска в 3σ скорость распределена на отрезке $[0 \frac{\text{км}}{\text{ч}}; 16.04 \frac{\text{км}}{\text{ч}}]$. Принимая скорость в точке стоянки, равной нулю, количество координат-стоянок в наборе составляет 1323, точек в движении – 2634.

- 3) методом диагностической фильтрации [1, 2], который был расширен в главе 2 работы, совместно с фильтром по точности, аналогичным фильтру в п. 1. В качестве исходных данных будем использовать весь тестовый набор данных;
- 4) фильтром Калмана совместно с фильтром по точности, аналогичным фильтру в п. 1. В качестве исходных данных будем использовать весь тестовый набор данных.

В общем случае оценка качества отфильтрованного трека в сравнении с реальным маршрутом пользователя представляет собой нетривиальную задачу. Разработаем набор методов оценки эффективности на основе существующих и новых предложенных методов оценки качества обработки геоданных.

4.1.1 Оценка пройденного расстояния

Одной из самых достоверных оценок качества является разница между пройденным расстоянием по реальному маршруту пользователя и дистанции по треку после фильтрации в метрах:

$$E_d^i = D_{groundTruth}^i - D_{predicted}^i, \quad (4.1)$$

где i – порядковый номер трека;

$D_{groundTruth}^i$ – пройденное расстояние по реальному маршруту i в метрах;

$D_{predicted}^i$ – пройденное расстояние по треку i после фильтрации в метрах.

В качестве оценки эффективности по пройденному расстоянию будем рассматривать среднее значение, среднеквадратичное отклонение, минимальное и максимальное отношение в метрах:

$$\overline{E_d} = \frac{1}{n} \sum_{i=1}^n E_d^i, \quad (4.2)$$

$$\sigma_{E_d} = \sqrt{\frac{1}{n} \sum_{i=1}^n (E_d^i - \overline{E_d})^2}, \quad (4.3)$$

$$E_d^{min} = \min_{i=1}^n (E_d^i), \quad (4.4)$$

$$E_d^{max} = \max_{i=1}^n (E_d^i), \quad (4.5)$$

где n – количество треков в выборке;

Особенности оценки:

- значение $\overline{E_d}$ может быть как положительным, так и отрицательным. В первом случае алгоритм фильтрации преимущественно занижает пройденную дистанцию по отношению к реальному маршруту, во втором – завышает;
- если $\overline{E_d} = 0$, то есть среднее расстояние по результирующим трекам совпадает со средними дистанциями по реальным маршрутам, и при условии, что σ_{E_d} мало, можно говорить о высокой степени качества фильтрации.

4.1.2 Оценка плотности точек в треке

Одной из особенностей предложенного алгоритма является уменьшение плотности точек на прямых участках маршрута, что позволит существенно уменьшить объемы хранимых и передаваемых данных с мобильного устройства на серверную часть системы. Определим оценку плотности точек как:

$$N = \frac{N_{raw}}{N_{predicted}}, \quad (4.6)$$

где N_{raw} – количество точек в исходном тестовом наборе данных;

где $N_{predicted}$ – количество точек после фильтрации.

Значение N по определению находится на полуинтервале $[1; \infty)$, при этом оно отражает только эффективность разрежения трека, не сравнивает пространственное сходство результирующего трека с реальным маршрутом: значение, близкое к единице означает низкую эффективность уменьшения плотности, в этом случае практически все исходные координаты будут составлять трек после фильтрации. Чем больше значение, тем более разрежен трек.

Будем использовать также относительную оценку:

$$N_{\%} = \left(1 - \frac{N_{predicted}}{N_{raw}}\right) * 100, \quad (4.7)$$

определяющую уменьшение плотности точек треков в процентах. Чем ниже значение, тем менее разрежен трек.

4.1.3 Оценка времени обработки

Дополним набор оценок средним временем в секундах между координатами на выходе системы:

$$\overline{T_{out}} = \frac{1}{N_{predicted}-1} \sum_{i=1}^{N_{predicted}-1} T_{out}^{i,i+1}, \quad (4.8)$$

где $T_{out}^{i,i+1}$ – интервал времени в секундах между i и $i + 1$ координатами, полученными на выходе системы.

Будем также вычислять среднеквадратичное отклонение, минимальное и максимальное значение времени между координатами в секундах:

$$\sigma_{T_{out}} = \sqrt{\frac{1}{N_{predicted}-1} \sum_{i=1}^{N_{predicted}-1} (T_{out}^{i,i+1} - \overline{T_{out}})^2}, \quad (4.9)$$

$$T_{out}^{min} = \min_{i=1}^{N_{predicted}-1} (T_{out}^{i,i+1}), \quad (4.10)$$

$$T_{out}^{max} = \max_{i=1}^{N_{predicted}-1} (T_{out}^{i,i+1}). \quad (4.11)$$

Предложим оценку среднего времени получения координаты на выходе системы после получения данных на входе:

$$\overline{T_{in \rightarrow out}} = \frac{1}{N_{predicted}} \sum_{i=1}^{N_{predicted}} T_{in \rightarrow out}^i, \quad (4.12)$$

где $T_{in \rightarrow out}^i$ – интервал времени в секундах между получением данных на вход системы и генерации координаты i на выходе. Соответствующее среднеквадратичное отклонение, минимальное и максимальное значение времени в секундах:

$$\sigma_{T_{in \rightarrow out}} = \sqrt{\frac{1}{N_{predicted}} \sum_{i=1}^{N_{predicted}} (T_{in \rightarrow out}^i - \overline{T_{in \rightarrow out}})^2}, \quad (4.13)$$

$$T_{in \rightarrow out}^{min} = \min_{i=1}^{N_{predicted}} (T_{in \rightarrow out}^i), \quad (4.14)$$

$$T_{in \rightarrow out}^{max} = \max_{i=1}^{N_{predicted}} (T_{in \rightarrow out}^i). \quad (4.15)$$

Оценки времен будем использовать для определения времени отклика системы фильтрации: чем меньше значение оценок, тем меньше отклик.

4.1.4 Оценка быстродействия фильтрации

Для определения требований к аппаратному обеспечению и сравнения эффективности предложенного алгоритма с существующими решениями необходимо оценить быстродействие. Определим оценку быстродействия как среднее время выполнения общего алгоритма фильтрации в миллисекундах:

$$\overline{T_p} = \frac{1}{N_{raw}} \sum_{i=1}^{N_{raw}} T_p^i, \quad (4.16)$$

где T_p^i – время в миллисекундах от поступления на вход данных об очередной текущей координате, до генерации данных на выходе.

Будем также оценивать среднеквадратичное отклонение, минимальное и максимальное значение времени в миллисекундах:

$$\sigma_{T_p} = \sqrt{\frac{1}{N_{raw}} \sum_{i=1}^{N_{raw}} (T_p^i - \overline{T_p})^2}, \quad (4.17)$$

$$T_p^{min} = \min_{i=1}^{N_{raw}} (T_p^i). \quad (4.18)$$

$$T_p^{max} = \max_{i=1}^{N_{raw}} (T_p^i), \quad (4.19)$$

В качестве времени T_p^i будем брать усредненное время выполнения алгоритма фильтрации на устройствах: HUAWEI MediaPad 7 Lite (Android 4.0.3) и Lenovo Tablet 8704X (Android 7.1.1).

4.1.5 Оценка точности фильтрации

Алгоритм фильтрации, предложенный в работе, фактически представляет собой бинарный классификатор со способностью предсказывать один единственный класс очередной координаты, генерируя на выходе дискретные значения:

- 0 – невалидная точка, точка значительно отклоняется от реальной траектории движения;
- 1 – валидная точка, точка не имеет существенных отклонений от реальной траектории движения.

Особенность классификатора заключается в следующем. Если предсказанный класс точки равен единице, ее координаты в общем случае могут не соответствовать координатам исходной, полученной на входе, из-за методов сглаживания и/или определения продолжения стоянки, алгоритмы которых могут модифицировать широту и долготу. Единственная характеристика, по которой можно однозначно идентифицировать координаты на входе и выходе – время получения. В связи с этим для оценки точности фильтрации целесообразно использовать общие соотношения, которые повсеместно используются для определения эффективности работы классификаторов, – достоверность (англ. Accuracy), точность (англ. Precision) и полнота (англ. Recall) [56]. Для их расчета по

исходным трекам необходимо построить целевые маршруты (англ. Ground Truth), присвоив каждой координате i параметр-маркер $p_i^{groundTruth}$, значение которого – ноль или единица. Маршруты будут использоваться также для оценки эффективности по пройденному расстоянию, метод расчета которой был рассмотрена выше. Значению единица для маркера будут соответствовать координаты:

- в движении, например, на дорогах;
- первых точек в стоянках, например, на дорогах перед светофорами или в зданиях;
- последующих точек за первой точкой в стоянках. В этом случае будем маркировать единицей как точки, которые не имеют существенных отклонений от реального местоположения устройства, так и выбросы, и отклонения, в предположении, что предложенный алгоритм способен определить по выбросам в местах стоянок ситуацию продолжения стоянки, пропуская очередную точку на выход и модифицируя широту, долготу и признак стоянки.

По точкам на выходе фильтрации можно составить треки, каждая точка которых также имеет маркер $p_i^{predicted}$, значение которого также равно 0 или 1. В этом случае характеристики точности соответствуют:

$$N_{TP} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 1 \text{ и } p_i^{predicted} = 1, \\ 0, \text{ иначе} \end{cases}, \quad (4.20)$$

$$N_{TN} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 0 \text{ и } p_i^{predicted} = 0, \\ 0, \text{ иначе} \end{cases}, \quad (4.21)$$

$$N_{FP} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 0 \text{ и } p_i^{predicted} = 1, \\ 0, \text{ иначе} \end{cases}, \quad (4.22)$$

$$N_{FN} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 1 \text{ и } p_i^{predicted} = 0, \\ 0, \text{ иначе} \end{cases}, \quad (4.23)$$

где N_{TP} – истинно-положительное решение (англ. True Positive), число точек в наборе данных, реальное значение которых $p_i^{groundTruth} = 1$ и предсказанное

$p_i^{predicted} = 1$. С учетом правил маркировки координат значением $p_i^{groundTruth}$, данная оценка вполне достоверна;

N_{TN} – истинно-орицательное решение (англ. True Negative), число точек в наборе данных, реальное значение которых $p_i^{groundTruth} = 0$ и предсказанное $p_i^{predicted} = 0$. Аналогично значению N_{TP} данная оценка также вполне достоверна;

N_{FP} – ложно-положительное решение (англ. False Positive, Type-I Error), число точек в наборе данных, реальное значение которых $p_i^{groundTruth} = 0$ и предсказанное $p_i^{predicted} = 1$. Данная оценка также достоверна;

N_{FN} – ложно-отрицательное решение (англ. False Negative, Type-II Error), число точек в наборе данных, реальное значение которых $p_i^{groundTruth} = 1$ и предсказанное $p_i^{predicted} = 0$. Данное решение может означать, что точка валидная, однако

- 1) например, была отброшена только для уменьшения плотности трека, так как она лежит на прямом участке движения или совпадает с предыдущей валидной координатой. Это естественное предполагаемое поведение фильтрации, однако такие ситуации будут учитываться как ложно-отрицательные решения, что неверно;
- 2) была отброшена фильтрацией ошибочно. Такое поведение возможно как для точек в движении, так и в местах стоянок. Данная ситуация верно рассматривается как ложно-отрицательное решение;
- 3) точка соответствует продолжению стоянки, и классификатор принял ошибочное решение. При расчете значения N_{FN} такая ситуация учитывается верно.

В связи с особенностью п. 1 оценка N_{FN} недостоверна. Учитывать уменьшение плотности точек возможно путем ручного подбора целевого маршрута таким образом, чтобы он соответствовал разреженному треку, или путем изменения алгоритма фильтрации для определения причины неоявления координаты на выходе – выявление ситуации, когда точка отбрасывается, так как имеет

существенные погрешности параметров, или она исключается только для уменьшения плотности трека. Оба подхода представляют собой темы отдельных исследований и в данной работе рассматриваться не будут.

Проблема расчета значения N_{FN} определяет нетривиальный характер вычисления производных оценок точности предложенного алгоритма, который не соответствует существующим определениям классификаторов. Сумма N_{FP} состоит как из верных ложно-отрицательных решений, так и неверных (см. п. 1). Если значение N_{FN} принять за вклад в True Negative N_{TN} , то итоговая $N_{FN} = 0$, и оценка N_{TN} будет недостоверной, так как будет включать часть верных ложно-отрицательных значений (см. п. 2 и п. 3). Если не менять метод оценки, то N_{FN} будет недостоверной, так как будет включать часть неверных ложноотрицательных значений (см. п. 1). Итоговая матрица ошибок представлена в Таблице 11.

Таблица 11.

Матрица ошибок

Predicted \ Ground Truth	0 – Невайдная точка	1 – Валидная точка
0 – Невайдная точка	N_{TN}	N_{FN}
1 – Валидная точка	N_{FP}	N_{TP}

Для увеличения достоверности оценки N_{FN} предложим метод, заключающийся в разбиении точек со значением маркера, равным 1, на два дополнительных класса. Промаркируем каждую координату исходного трека значением $p_i^{groundTruth}$. При этом:

- если $p_i^{groundTruth} = 0$, то точка невайдная, она имеет большое отклонение от реального маршрута. Это соответствует ранее рассмотренному методу маркировки точек, где $p_i^{groundTruth} = 0$;
- если $p_i^{groundTruth} = 1$, то точка валидная, имеет малую погрешность определения широты и долготы, отражает координату в движении, допустим на дороге, или первую точку в стоянке, например, на дороге перед светофором или в здании;

- если $p_i^{groundTruth} = 2$, то точка валидная, и она отражает последующие координаты за первой точкой в стоянке.

Оба последних пункта соответствуют ранее рассмотренному методу маркировки, при $p_i^{groundTruth} = 1$. По точкам на выходе фильтрации составим треки, каждая точка которых имеет маркер $p_i^{predicted}$, значение которого также равно 0, 1 или 2. Рассчитаем истинно-отрицательные решения, в которые добавим неверные ложно-отрицательные, когда точка была отброшена фильтрацией только для уменьшения плотности трека (см. п. 1). В эту же сумму будут включаться верные ложно-отрицательные решения, что является недостатком предложенного метода. В связи с этим истинно-положительные решения соответствуют оценке по предыдущему методу, однако в этом случае учитывается дополнительное значение маркера:

$$N'_{TP} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 1 \text{ и } p_i^{predicted} = 1 \text{ или} \\ p_i^{groundTruth} = 2 \text{ и } p_i^{predicted} = 2 \\ 0, \text{ иначе} \end{cases} . \quad (4.24)$$

Истинно-отрицательные решения будут соответствовать:

$$N'_{TN} = N_{TN} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 0 \text{ и } p_i^{predicted} = 0 \text{ или} \\ p_i^{groundTruth} = 1 \text{ и } p_i^{predicted} = 0 \text{ или} \\ p_i^{groundTruth} = 2 \text{ и } p_i^{predicted} = 0 \\ 0, \text{ иначе} \end{cases} . \quad (4.25)$$

Ложно-положительные решения будут также соответствовать предыдущей оценке:

$$N'_{FP} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 0 \text{ и } p_i^{predicted} = 1 \text{ или} \\ p_i^{groundTruth} = 0 \text{ и } p_i^{predicted} = 2 \\ 0, \text{ иначе} \end{cases} . \quad (4.26)$$

Поскольку в истинно-отрицательные решения вошла часть ложноотрицательных, оценка ложно-отрицательных решений:

$$N'_{FN} = \sum_{i=1}^{N_{raw}} \begin{cases} 1, \text{ при } p_i^{groundTruth} = 1 \text{ и } p_i^{predicted} = 2 \text{ или} \\ p_i^{groundTruth} = 2 \text{ и } p_i^{predicted} = 1 \\ 0, \text{ иначе} \end{cases} , \quad (4.27)$$

в которой не присутствует часть верных и неверных ложно-отрицательных решений.

Итоговая матрица ошибок представлена в Таблице 12.

Таблица 12.

Матрица ошибок для предложенного метода оценки точности

Ground Truth \ Predicted	0 – Невайдная точка	1 – Валидная точка в движении или в начале стоянки	2 – Валидная точка продолжения стоянки
0 – Невайдная точка	N'_{TN}	$N'_{FN} \rightarrow N'_{TN}$	$N'_{FN} \rightarrow N'_{TN}$
1 – Валидная точка в движении или в начале стоянки	N'_{FP}	N'_{TP}	N'_{FN}
2 – Валидная точка продолжения стоянки	N'_{FP}	N'_{FN}	N'_{TP}

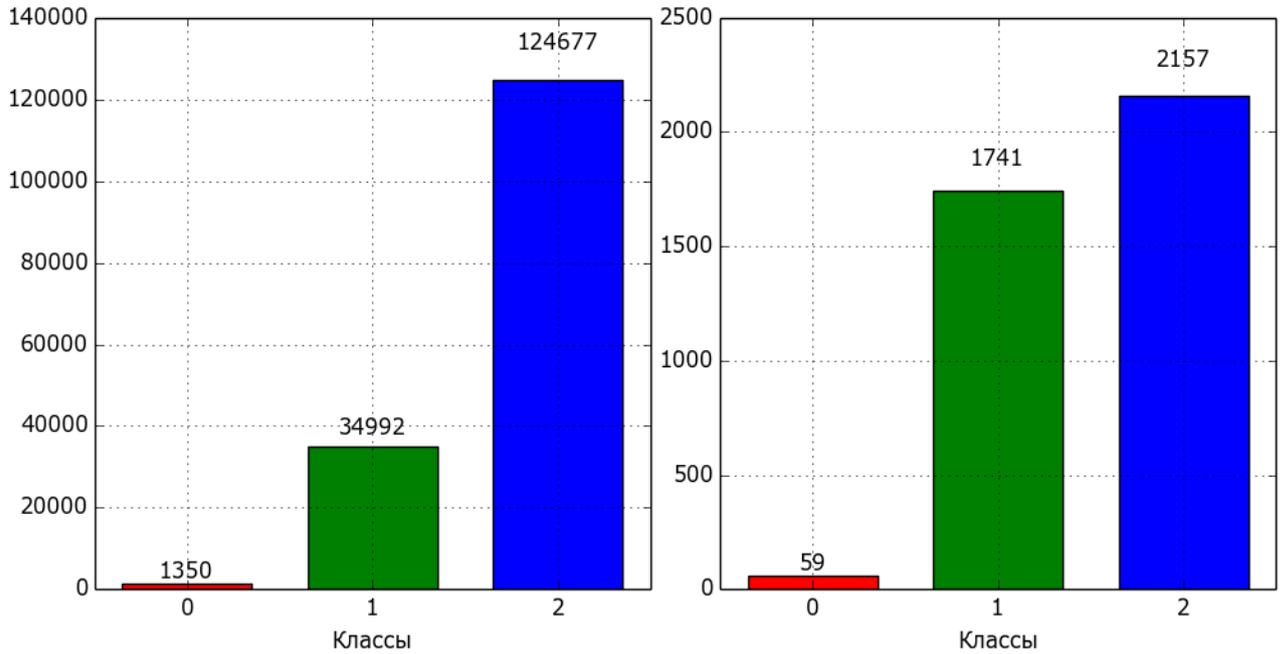
Разбиение классов по предложенному методу позволяет увеличить достоверность истинно-отрицательных и ложно-отрицательных оценок путем разбиения трех наборов точек ложноотрицательного класса, рассмотренного выше, на два дополнительных класса. В этом случае $N'_{FN} \neq 0$. Однако стоит отметить, что ни N'_{TN} , ни N'_{FN} не являются полностью адекватными оценками, так как в первую сумму добавляются верные ложно-отрицательные решения, которые должны присутствовать во второй.

На основе рассчитанных оценок определим достоверность фильтрации в процентах, как:

$$E_{acc\%} = \frac{N'_{TP} + N'_{TN}}{N_{raw}} * 100, \quad (4.28)$$

которая отражает количество точек, по которым классификатор принял верное решение. Оценка достоверности имеет следующую особенность. Она присваивает всем точкам одинаковый вес, и в случае, когда распределение маркеров в исходном наборе данных смещено в сторону тех или иных значений, оценка перестает быть адекватной. Это может привести к тому, что, например, достоверность $E_{acc\%} = 90\%$, однако в рамках какого-то отдельного маркера-класса

классификатор работает неточно. Рассмотрим распределение классов для двух наборов данных (см. Рисунок 4.3).



(а) распределение классов в тестовом наборе данных (б) распределение классов в тестовом наборе данных для оценки Fused геолокации

Рисунок 4.3. Распределение классов в наборах данных

По рисункам видно, что тестовые наборы данных сильно не сбалансированы, что повлечет за собой недостоверность оценки $E_{acc\%}$. Для нивелирования влияния распределения классов введем такие показатели, как точность и полнота соответственно:

$$E_{precision} = \frac{N'_{TP}}{N'_{TP} + N'_{FP}}, \quad (4.29)$$

$$E_{recall} = \frac{N'_{TP}}{N'_{TP} + N'_{FN}}. \quad (4.30)$$

Их оценки в процентах соответственно:

$$E_{precision\%} = E_{precision} * 100 = \frac{N'_{TP}}{N'_{TP} + N'_{FP}} * 100, \quad (4.31)$$

$$E_{recall\%} = E_{recall} * 100 = \frac{N'_{TP}}{N'_{TP} + N'_{FN}} * 100. \quad (4.32)$$

По определению точность $E_{precision}$ есть доля точек, действительно принадлежащих классу относительно всех координат, которые классификатор отнес к этому классу. Полнота E_{recall} – это доля найденных классификатором точек,

принадлежащих классу, относительно всех координат этого класса в исходной выборке.

Существует объединенная характеристика, которая учитывает оба показателя:

$$E_F = (\beta^2 + 1) \frac{E_{precision} E_{recall}}{\beta^2 E_{precision} + E_{recall}}, \quad (4.33)$$

где при $\beta \in (0; 1)$ оценка отдает приоритет точности, при $\beta > 1$ приоритет отдается полноте.

В общем случае можно оценивать точность классификатора только по оценке E_F смещая коэффициент β в ту или иную сторону. При $\beta = 1$ формула вырождается в сбалансированную меру F_1 , гармоническое среднее:

$$E_{F_1} = \frac{2E_{precision}E_{recall}}{E_{precision} + E_{recall}}, \quad (4.34)$$

которое в равной степени учитывает и точность $E_{precision}$, и полноту E_{recall} .

Для оценки точности алгоритмов фильтрации как классификаторов будем использовать оценки $E_{acc\%}$, $E_{precision\%}$, $E_{recall\%}$ и E_{F_1} , значения которых по определению прямо пропорциональны эффективности работы классификатора.

Часто используется оценка эффективности классификаторов по ROC-кривой (англ. Receiver Operating Characteristic). В зависимости от порога решающего правила строится график функции $N'_{TP} = f(N'_{FP})$, которую обычно называют кривой ошибок. Далее анализируется показатель AUC (англ. Area Under ROC Curve), соответствующий площади под кривой, и другие характеристики функции. Однако предложенный алгоритм фильтрации, как классификатор, способен определять принадлежность точки к классу только дискретными значениями 0 или 1 (0, 1 или 2), то есть порог решающего правила здесь отсутствует, кривая ошибок вырождается в точку, и ROC-анализ не применим.

4.1.6 Оценка ошибки фильтрации

Для оценки качества работы фильтров и эффективности предсказания классификаторов обычно также используют такую характеристику, как функцию

ошибки, которую конструируют исходя из оценки идентичности целевой функции (целевого маршрута, Ground Truth) и предсказаний классификатора или данных на выходе фильтрации, учитывая специфику предметной области, таким образом, чтобы ошибка была максимально достоверной для данной конкретной задачи [44]. Значение функции ошибки должно уменьшаться с увеличением качества работы модели. При этом решая задачу минимизации функции, возможно найти оптимальные параметры модели. В качестве функции ошибки наряду с евклидовой метрикой часто используют кросс-энтропию или функцию потерь Хьюбера. Разработаем функцию ошибки применительно к задаче фильтрации геокоординат.

Целевой маршрут фактически представляет собой ломаную линию, заданную вектором кортежей. Каждый кортеж представляет собой набор параметров: времени, широта и долгота, которые определяют точку в трехмерном пространстве:

$$L_{groundTruth} = (\dots, \langle t_i, lat_i, lon_i \rangle, \dots). \quad (4.35)$$

Будем составлять реальный маршрут $L_{groundTruth}$ из точек, у которых $p_i^{groundTruth} = 1$ или 2, при этом точки, где $p_i^{groundTruth} = 2$, сместим в начало стоянки – ближайшую предыдущую по времени точку, у которой $p_i^{groundTruth} = 1$. Координаты с параметром $p_i^{groundTruth} = 0$ не будут учитываться в маршруте.

Трек, полученный после фильтрации, также представляет собой ломаную:

$$L_{predicted} = (\dots, \langle t_j, lat_j, lon_j \rangle, \dots), \quad (4.36)$$

которая строится аналогично $L_{groundTruth}$.

Особенностью такого представления треков является то, то количество элементов в векторах $L_{groundTruth}$ и $L_{predicted}$ в общем случае может быть разным. Координаты широты $lat_i(lat_j)$ и долготы $lon_i(lon_j)$ могут быть также разными в обоих ломаных. Только по параметру $t_i(t_j)$ возможно однозначно определить две точки соответствующих одному и тому же времени. По этому параметру существует некая неравномерная дискретизация координат: каждой

единице времени t_k соответствует не менее одного кортежа $\langle t_k, lat_k, lon_k \rangle$ в векторах $L_{groundTruth}$ и $L_{predicted}$. То есть возможны случаи, когда:

- $\exists t_k: \exists! \langle t_k, lat_k, lon_k \rangle \in L_{groundTruth}, \nexists \langle t_k, lat_{k2}, lon_{k2} \rangle \in L_{predicted};$
- $\exists t_k: \nexists \langle t_k, lat_{k1}, lon_{k1} \rangle \in L_{groundTruth}, \exists \langle t_k, lat_k, lon_k \rangle \in L_{predicted};$
- $\exists t_k: \exists! \langle t_k, lat_{k1}, lon_{k1} \rangle \in L_{groundTruth}, \exists! \langle t_k, lat_{k2}, lon_{k2} \rangle \in L_{predicted}.$

Необходимо сравнить идентичность двух ломаных $L_{groundTruth}$ и $L_{predicted}$ функцией ошибки $E_{loss} \in \mathbb{R}$ как по пространственным координатам lat_i и lon_i , так и по временным t_i . Чем больше различаются ломаные, тем больше значение ошибки E_{loss} и наоборот. В общем случае $E_{loss} \in [0; \infty)$. При этом известно, что ломаные изначально так расположены в пространстве, что $E_{loss} = min$, так как всегда рассматривается одна и та же реальная траектория движения пользователя. Вектор $L_{predicted}$ в случае идеальной фильтрации должен совпадать с $L_{groundTruth}$, и $E_{loss} = 0$.

Идентичность ломаных в данном контексте следует понимать, как степень соответствия каждой точки $L_{predicted}$ с соответствующей точкой в $L_{groundTruth}$. Степень подобия в отличие от идентичности не будет адекватной оценкой, поскольку ломаные могут быть схожи с высоким показателем подобия, однако расположены далеко друг от друга, при этом значение ошибки будет мало, что неверно.

Задача конструирования E_{loss} усложняется тем, что координаты точек на ломаных lat_i, lon_i и lat_j, lon_j – это геокоординаты в замкнутом пространстве, эллипсоиде, а не точки на плоскости. Расстояние от точки до прямой здесь определяется дугой. Перевод геокоординат в прямоугольную систему координат может добавить существенную погрешность определения значения функции ошибки.

Рассмотрим возможные решения задачи нахождения максимально достоверной функции E_{loss} .

1. Ковариационный анализ. Метод предполагает составления матрицы ковариации между распределенными величинами. В нашем случае величинами являются координаты точек в двух ломаных $L_{predicted}$ и $L_{groundTruth}$.

После составления матрицы анализируются коэффициенты корреляции. По их значениям можно определить степень идентичности двух ломаных, однако метод применим только в случае одинаковой дискретизации точек на ломаных, что в нашем случае неверно.

2. Анализ Фурье и сравнение спектров. Проблема в данном случае аналогична первому пункту. Для ее разрешения возможно аппроксимировать дискретные функции $f_{t+dt} = (lat_i, lon_i)$ на интервалах времени $t + dt$ непрерывными функциями для каждой ломаной, выполнить передискретизацию по времени и сравнить непрерывные функции ковариационным анализом, анализом Фурье или простым вычислением интеграла разницы функций на интервалах $t + dt$. Однако определение значения dt для максимально достоверной оценки представляет собой нетривиальную задачу. Дело в том, что ломаные могут вести себя по-разному, в том числе описывать окружности на больших интервалах, и в таком случае аппроксимация не имеет смысла. На малых интервалах возможна большая погрешность вычисления ошибки E_{loss} .
3. Предложим три подхода, основанных на векторном представлении ломаных. Первый определяет E_{loss} как разницу сумм изменения углов направления по двум ломаным. Вторым подходом является перевод всех векторов в начало координат и подсчете разницы общих площадей под двумя ломаными. Однако в обоих случаях оценка может быть недостоверна, так как возможно построить два различных трека с нулевой ошибкой E_{loss} . Третий подход предполагает попарное сравнение векторов, но здесь появляется проблема дискретизации.

Все рассмотренные подходы с высокой степенью недостоверности вычисляют ошибку E_{loss} . Для декомпозиции задачи попробуем выделить частные случаи анализа двух ломаных.

Допустим моменту времени t_k соответствуют две точки на обоих ломаных, то есть: $\exists t_k: \exists! \langle t_k, lat_{k1}, lon_{k1} \rangle \in L_{groundTruth}, x_{k1} = (lat_{k1}, lon_{k1})$ и $\exists! \langle t_k, lat_{k2}, lon_{k2} \rangle \in L_{predicted}, x_{k2} = (lat_{k2}, lon_{k2})$. Ошибка

здесь очевидно пропорциональна расстоянию между двумя точками x_{k1} и x_{k2} в метрах:

$$E_{loss}^1 \sim D(x_{k1}, x_{k2}), \quad (4.37)$$

где D – функция расстояния между точками x_{k1} и x_{k2} .

Рассмотрим второй случай, когда $\exists t_k: \exists! \langle t_k, lat_k, lon_k \rangle \in L_{groundTruth}, x_k = (lat_k, lon_k), \nexists \langle t_k, lat_{k2}, lon_{k2} \rangle \in L_{predicted}$ или $\exists t_k: \nexists \langle t_k, lat_{k1}, lon_{k1} \rangle \in L_{groundTruth}, \exists \langle t_k, lat_k, lon_k \rangle \in L_{predicted}, x_k = (lat_k, lon_k)$, то есть для момента времени t_k существует только одна точка на одной из ломаных. Выполнение любого из условий сводится к одной задаче, поэтому далее будем рассматривать точку $\langle t_k, lat_k, lon_k \rangle$ на одной ломаной ($L_{groundTruth}$ или $L_{predicted}$), для которой нет соответствующей точки на другой ломаной ($L_{predicted}$ или $L_{groundTruth}$). Ошибка здесь пропорциональна расстоянию от точки x_k на первой ломаной до точки на второй ломаной \tilde{x}_l в метрах, которая на самом деле отсутствует:

$$E_{loss}^2 \sim D(x_k, \tilde{x}_l), \quad (4.38)$$

Эту точку целесообразно определить, как точку с местоположением пользователя, где бы он находился на второй ломаной в момент времени t_k .

За \tilde{x}_l можно взять существующую точку на второй ломаной, ближайшую к моменту времени t_k . Однако данный подход может существенно увеличить погрешность определения итоговой ошибки E_{loss} , так как ближайшая точка по времени может располагаться на значительном расстоянии от x_k . Для нивелирования погрешности следует учитывать разницу во времени, что представляет собой нетривиальную задачу.

Второй способ определения \tilde{x}_l заключается в нахождении на второй ломаной двух точек $x_l = (lat_l, lon_l)$ и $x_{l+1} = (lat_{l+1}, lon_{l+1})$, предыдущей и следующей по отношению к моменту времени t_k . Такой подход привнесет в определение ошибки E_{loss} не только оценку пространственной идентичности ломаных, но и временной. Следует учитывать, что в начале и/или в конце трека на одной из

ломанных может не быть точек x_l и/или x_{l+1} . В этом случае ошибка пропорциональна расстоянию до существующей точки x_l или x_{l+1} :

$$E_{loss}^2 \sim D(x_k, x_l), \quad (4.39)$$

$$E_{loss}^2 \sim D(x_k, x_{l+1}). \quad (4.40)$$

Если обе точки присутствуют на ломаной, имеем треугольник с геокоординатами x_l , x_{l+1} и x_k . Ошибка:

$$E_{loss}^2 \sim D(x_k, \tilde{x}_l) = D'(x_k, x_l, x_{l+1}), \quad (4.41)$$

где D' – функция расстояния между точкой x_k и дугой, составленной из точек x_l и x_{l+1} .

Рассмотрим подходы для определения функции D' .

1. Определение D' и \tilde{x}_l по скорости. В качестве \tilde{x}_l в данном подходе рассматривается точка, которая находится на отрезке (x_l, x_{l+1}) , на расстоянии $d_{l,k}$ от x_l . При этом $d_{l,k}$ – расстояние, которое может преодолеть пользователь за время $t_k - t_l$ со скоростью spd_l . Время t_l здесь соответствует кортежу $\langle t_l, lat_l, lon_l \rangle$, spd_l – моментальная скорость в точке x_l . Итоговая ошибка определяется расстоянием между x_k и \tilde{x}_l :

$$E_{loss}^2 \sim D(x_k, \tilde{x}_l) = D'(x_k, x_l, x_{l+1}) = D(x_k, \tilde{x}_l). \quad (4.42)$$

Однако скорость spd_l может определяться с большой погрешностью; движение в интервале времени от t_l до t_k может быть не равномерным, что предполагает учет ненулевого ускорения и/или градиентов ускорения, погрешность вычисления которых может лишь расти по отношению к spd_l .

В результате накопленная ошибка для E_{loss}^2 может быть значительна.

2. Определение D' как длины кратчайшего отрезка дуги от точки x_k до отрезка дуги (x_l, x_{l+1}) . При этом необходимо рассмотреть случаи, когда:

- точка \tilde{x}_l на дуге (x_l, x_{l+1}) , которая лежит на кратчайшем отрезке дуги, не лежит на отрезке дуги (x_l, x_{l+1}) . В этом случае в качестве E_{loss}^2 следует брать расстояние до ближайшей точки:

$$E_{loss}^2 \sim D(x_k, \tilde{x}_l) = D'(x_k, x_l, x_{l+1}) = \min(D(x_k, x_l), D(x_k, x_{l+1})). \quad (4.43)$$

- точка \tilde{x}_l на дуге (x_l, x_{l+1}) , которая лежит на кратчайшем отрезке дуги, лежит на отрезке дуги (x_l, x_{l+1}) . В этом случае ошибка пропорциональна кратчайшему расстоянию:

$$E_{loss}^2 \sim D(x_k, \tilde{x}_l) = D'(x_k, x_l, x_{l+1}) = D(x_k, \tilde{x}_l). \quad (4.44)$$

3. Определение D' как разницы расстояний. Данный подход предполагает расчет D' как разницы суммы расстояний от x_l до x_k , от x_k до x_{l+1} и от x_l до x_{l+1} :

$$E_{loss}^2 \sim D(x_k, \tilde{x}_l) = D'(x_k, x_l, x_{l+1}) = D(x_l, x_k) + D(x_k, x_{l+1}) - D(x_l, x_{l+1}). \quad (4.45)$$

Иллюстрация приведена на Рисунке 4.4.

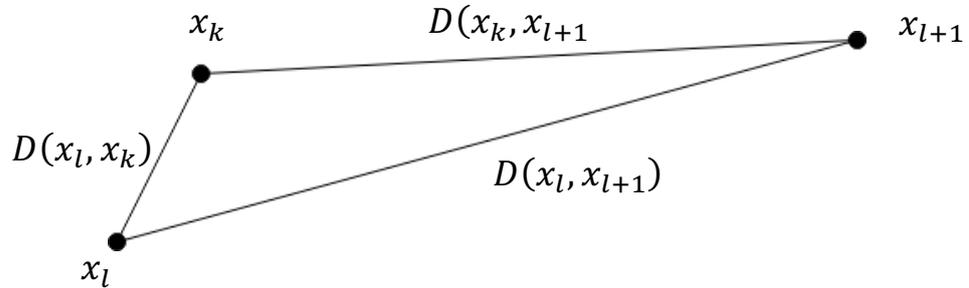


Рисунок 4.4. Иллюстрация расчета D' как разницы расстояний

Прямые между точками на рисунке эквивалентны дугам. Функция E_{loss}^2 отражает оценку расстояния между x_k и дугой (x_l, x_{l+1}) . Чем меньше E_{loss}^2 , тем ближе точка x_k к дуге (x_l, x_{l+1}) , чем больше – тем дальше. Если $E_{loss}^2 = 0$, точка x_k лежит на дуге.

Исходя из минимизации погрешности и трудоемкости определения функции E_{loss} был выбран подход, учитывающий расстояние между точками в случае присутствия обеих точек в определенный момент времени на двух ломаных $L_{predicted}$ и $L_{groundTruth}$ и разницу расстояний между точками (п. 3 выше) в противном случае.

Поскольку ошибка может быть неабсолютной величиной и не должна в общем случае отражать какие-либо параметры трека, например, сумму отклонений расстояний и др., знаки пропорциональности в формулах (4.37) и (4.45) заменим на знаки равенства. Общая ошибка E_{loss} будет являться квадратным корнем из среднеквадратической ошибки модели – отношения суммы квадратов

расстояний E_{loss}^1 или E_{loss}^2 в зависимости от наличия или отсутствия обеих точек на двух ломаных соответственно к количеству точек:

$$E_{loss} = \sqrt{\frac{\sum_{i=1}^{N_{max}} E_i^2}{N_{max}}}, \quad (4.46)$$

где $N_{max} = \max(N_{predicted}, N_{groundTruth})$, так как вычисление ошибки должно быть выполнено для каждой точки ломаных $L_{predicted}$ и $L_{groundTruth}$:

$E_i = D(x_{i,1}, x_{i,2})$, $x_{i,1} = (lat_{i,1}, lon_{i,1})$, $x_{i,2} = (lat_{i,2}, lon_{i,2})$, при условии, что $\langle t_i, lat_{i,1}, lon_{i,1} \rangle \in L_{groundTruth}$ и $\langle t_i, lat_{i,2}, lon_{i,2} \rangle \in L_{predicted}$. В противном случае, когда на одной из ломаных нет точки в момент t_i , т.е. $x_i = (lat_i, lon_i) = (lat_{i,1}, lon_{i,1})$, $\langle t_i, lat_{i,1}, lon_{i,1} \rangle \in L_{exist} = L_{groundTruth}$ и $\langle t_i, lat_{i,2}, lon_{i,2} \rangle \notin L_{notExist} = L_{predicted}$ или $\langle t_i, lat_{i,1}, lon_{i,1} \rangle \notin L_{notExist} = L_{groundTruth}$ и $x_i = (lat_i, lon_i) = (lat_{i,2}, lon_{i,2})$, $\langle t_i, lat_{i,2}, lon_{i,2} \rangle \in L_{exist} = L_{predicted}$:

- $E_i = D(x_i, x_l)$, при $x_l = (lat_l, lon_l)$, $\langle t_l, lat_l, lon_l \rangle \in L_{notExist}$, $t_l < t_i$, если $\forall t_{l+1} > t_i: \nexists \langle t_{l+1}, lat, lon \rangle \in L_{notExist}$;
- $E_i = D(x_i, x_{l+1})$, при $x_{l+1} = (lat_{l+1}, lon_{l+1})$, $\langle t_{l+1}, lat_{l+1}, lon_{l+1} \rangle \in L_{notExist}$, $t_{l+1} > t_i$, если $\forall t_{l-1} < t_i: \nexists \langle t_{l-1}, lat, lon \rangle \in L_{notExist}$;
- $E_i = D(x_l, x_i) + D(x_i, x_{l+1}) - D(x_l, x_{l+1})$, при $x_l = (lat_l, lon_l)$, $\langle t_l, lat_l, lon_l \rangle \in L_{notExist}$ и $x_{l+1} = (lat_{l+1}, lon_{l+1})$, $\langle t_{l+1}, lat_{l+1}, lon_{l+1} \rangle \in L_{notExist}$.

Будем использовать также максимальную и среднюю оценки ошибки:

$$E_{loss}^{max} = \max_{i=1}^{N_{max}} (E_i), \quad (4.47)$$

$$\overline{E}_{loss} = \frac{1}{N_{max}} \sum_{i=1}^{N_{max}} E_i. \quad (4.48)$$

Предложенный метод расчета значения функции ошибки использует как оценку пространственной идентичности ломаных, так и временной, учитывая специфику предметной области задачи. Заметим, что и предложенная оценка ошибка, и любые другие оценки эффективности, значения которых сильно зависят от геокоординат точек на треке, расстояний между ними, будут коррелироваться с ошибкой пройденного расстояния, рассмотренной ранее. Одним из преимуществ метода также является небольшая вычислительная сложность. В

качестве недостатков метода стоит отметить отсутствие учета метрики пространства в случае сравнения близости точки к отрезку дуги. Проведя кратчайшее расстояние или принимая во внимание скоростные и временные показатели движения, возможно сделать оценку более достоверной. Однако на основе сравнения экспертного анализа треков и корреляции экспертных оценок с функцией ошибки, можно сказать, что данный метод позволяет с достаточной степенью достоверности определить качество фильтрации, что с учетом оценок, рассмотренных в предыдущих разделах, позволит эффективно сравнить существующие методы фильтрации с предложенным алгоритмом обработки геокоординат.

4.2 Результаты экспериментов

Как было рассмотрено ранее, эффективность алгоритма фильтрации будем оценивать с существующими методами обработки геокоординат:

- 1) стандартным API геолокации операционной системы Android совместно с пороговым фильтром по точности *acc* без учета ограничений на параметры погрешности *hdop* и числа спутников *sat*. Пороговый параметр фильтрации P_{maxAcc}^1 примем равным 30 м. В качестве исходных данных будем использовать весь тестовый набор данных, состоящий из 24 треков с общим количеством точек, равным 161019;
- 2) технологией Fused геолокации Android совместно с пороговым фильтром по точности, аналогичным фильтру в п. 1. Здесь будем использовать отдельный тестовый набор данных из 2 треков с количеством точек 3957;
- 3) метод диагностической фильтрации [1, 2], совместно с фильтром по точности, аналогичным фильтру в п. 1. В качестве исходных данных будем использовать весь тестовый набор данных;
- 4) фильтр Калмана совместно с фильтром по точности, аналогичным фильтру в п. 1. В качестве исходных данных будем использовать весь тестовый набор данных.

Для оценки качества результирующих треков и эффективности работы фильтрации будем использовать предложенные рассмотренные:

1. среднее, минимальное, максимальное отклонение пройденного расстояния от реального и среднеквадратичное отклонение оценки: $\overline{E_d}, E_d^{min}, E_d^{max}, \sigma_{E_d}$ соответственно;
2. оценку плотности точек в треке и ее аналог в процентном соотношении: N и $N_{\%}$ соответственно;
3. среднее, минимальное, максимальное время между получением координат на выходе системы, между получением данных на входе и поступлении координаты на выход системы, их среднеквадратичные отклонения: $\overline{T_{out}}, T_{out}^{min}, T_{out}^{max}, \sigma_{T_{out}}, \overline{T_{in \rightarrow out}}, T_{in \rightarrow out}^{min}, T_{in \rightarrow out}^{max}$ и $\sigma_{T_{in \rightarrow out}}$ соответственно;
4. среднее, минимальное, максимальное время обработки одной координаты и среднеквадратичное отклонение оценки: $\overline{T_p}, T_p^{min}, T_p^{max}$ и σ_{T_p} соответственно;
5. оценки достоверности, точности, полноты и F_1 меры: $E_{acc\%}, E_{precision\%}, E_{recall\%}$ и E_{F_1} соответственно;
6. ошибку фильтрации, ее среднее и максимальное значение: $E_{loss}, \overline{E_{loss}}$ и E_{loss}^{max} соответственно.

Все параметры фильтров предложенного алгоритма были подобраны экспертным путем на основе визуального анализа исходных, результирующих треков и реальных маршрутов из сформированных тестовых наборов данных. Результаты эксперимента приведены в Таблице 13.

Проанализировав таблицу можно утверждать следующее:

- 1) предложенный алгоритм является лучшим из рассмотренных по оценке пройденного расстояния – одной из самых достоверных оценок качества фильтрации. Среднее значение отклонения составляет 943 метра, что 29.558 раз меньше полученного по технологии Fused геолокации

Результаты эксперимента

Таблица 13.

Оценка	Обозначение	API геолокации Android совместно с фильтром по точности	Fused геолокация Android совместно с фильтром по точности	Диагностическая фильтрация совместно с фильтром по точности	Фильтр Калмана совместно с фильтром по точности	Предложенный алгоритм
Пройденное расстояние, м	$\overline{E_d} \pm \sigma_{E_d}$	195511 ± 673195	27873 ± 23600	193769 ± 669576	195740 ± 673130	943 ± 2151
	$E_d^{min} \dots E_d^{max}$	2 ... 3067293	4272 ... 51473	0 ... 3047921	3 ... 3067293	0 ... 10353
Плотность точек	$N(N\%)$	2.195(54.447)	1.028(2.704)	1.875(46.667)	1.388(27.948)	2.934(65.916)
Время, с	$\overline{T_{out}} \pm \sigma_{T_{out}}$	12.218 ± 343.194	3.081 ± 75.397	10.596 ± 318.569	7.853 ± 274.127	16.686 ± 311.951
	$T_{out}^{min} \dots T_{out}^{max}$	0.002 ... 55077.293	0.001 ... 4112.530	0.001 ... 55077.293	0.001 ... 55077.293	1 ... 55074.407
	$\overline{T_{in \rightarrow out}} \pm \sigma_{T_{in \rightarrow out}}$	15.411 ± 497.692	8.535 ± 64.021	22.678 ± 536.987	41.042 ± 747.892	14.260 ± 442.365
	$T_{in \rightarrow out}^{min} \dots T_{in \rightarrow out}^{max}$	0.001 ... 55056.578	0.001 ... 569.480	0.001 ... 55056.578	0.001 ... 55056.578	0.001 ... 55053.692
Быстродействие, мс	$\overline{T_p} \pm \sigma_{T_p}$	0.519 ± 1.031	479.744 ± 448.5	0.695 ± 1.518	1.33 ± 2.479	3.444 ± 8.348
	$T_p^{min} \dots T_p^{max}$	0.001 ... 215.207	3 ... 999	0.002 ... 343.571	0.002 ... 449.414	0.002 ... 342.198
Точность	$E_{acc\%}$	70.202	59.464	97.774	51.571	99.164
	$E_{precision\%}$	99.051	97.822	99.725	98.941	99.836
	$E_{recall\%}$	34.7	59.105	96.081	32.901	97.704
	E_{F_1}	0.514	0.737	0.979	0.494	0.988
Ошибка, м	E_{loss}	4065.586	38.220	3566.108	3508.163	97.686
	$\overline{E_{loss}}$	72.024	13.788	47.825	59.68	13.108
	E_{loss}^{max}	951576.113	1104.327	951576.113	951576.113	11203.043

Android. Минимальное значение отклонения составляет 0 метров, что означает полное соответствие пройденной дистанции по результирующему треку с расстоянием по реальному маршруту пользователя в некоторых треках;

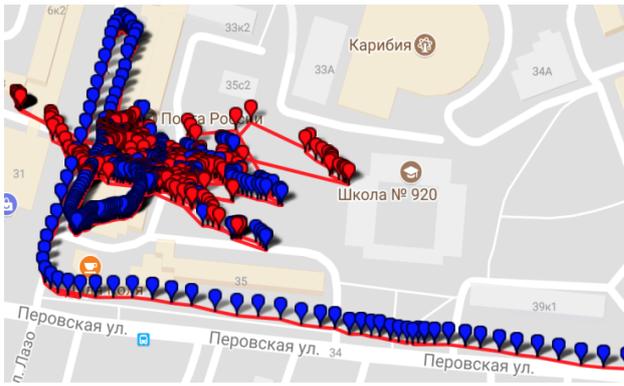
- 1) предложенный алгоритм снижает плотность точек в треке на 65.916 %, что на 11.469 % больше, чем стандартное API геолокации Android совместно с фильтром по точности;
- 2) оценки среднего времени между координатами на выходе системы и между получением данных на вход и поступлением координаты на выход принадлежат Fused геолокации совместно с фильтром по точности, значения показателей для предложенного алгоритма составляют 16.686 с и 14.260 с против 3.081 с и 8.535 с. Поскольку алгоритм, предназначен в том числе для уменьшения плотности точек в треке, значения оценок ожидаемо увеличились. Стоит отметить, что минимальный интервал времени между получением данных на вход и поступлением координаты на выход составляет 1 мс, что соответствует Fused геолокации, то есть в определенных ситуациях предложенный алгоритм не уменьшает плотность трека, например, в случае, когда следующая координата находится относительно далеко от предыдущей;
- 3) за счет обработки потока геокоординат набором фильтров время обработки ожидаемо увеличилось: 3.444 миллисекунды, против 0.519 у API геолокации, однако это совершенно несущественно для современных мобильных устройств;
- 4) показатели достоверности, точности, полноты и мера F_1 имеют наибольшие значения для предложенного алгоритма. Достоверность составляет 99.164 %, что на 1.39 % больше достоверности технологии диагностической фильтрации совместно с фильтром по точности. Точность равна 99.836 % (+0.111 %), полнота – 97.704 % (+1.623 %), F_1 мера принимает значение 0.988, против 0.979;

5) среднеквадратическая ошибка E_{loss} , как и максимальное значение E_{loss}^{max} минимальны для Fused технологии совместно с фильтром по точности и составляют 38.220 м и 1104.327 м, против 97.686 м и 11203.043 м у предложенного алгоритма, однако среднее значение ошибки минимально у рассмотренной системы фильтрации – 13.108 м, против 13.788 м.

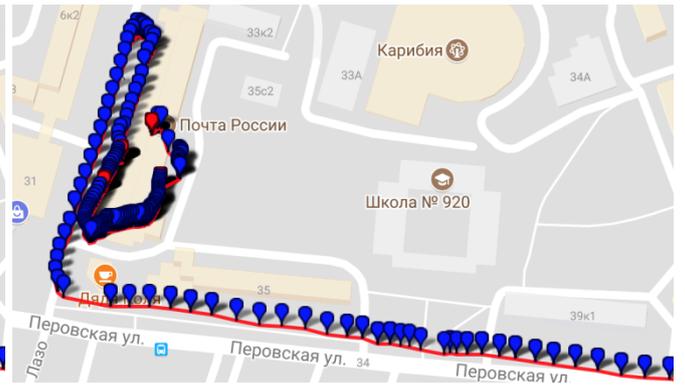
Количество точек в наборе данных, на котором вычислялись оценки эффективности Fused геолокации, составляет всего 3957, что мало в сравнении с числом точек в исходном тестовом наборе (161019). Более того, отношение количества координат в местах стоянок к числу всех точек во Fused наборе намного больше аналогичного показателя для исходного тестового набора. В связи с этим можно говорить о недостаточной достоверности всех оценок для технологии Fused геолокации, и, в частности, значения среднеквадратичной ошибки, так как именно в местах стоянок появляются продолжительные выбросы, что приводит к накапливанию ошибки.

Для доказательства эффективности фильтрации экспертной визуальной оценкой рассмотрим некоторые треки в движении и местах стоянок до и после фильтрации (см. Рисунок 4.5). Красным цветом отмечены маркеры, где скорость пользователя нулевая, синим – координаты в движении. Из анализа треков следует, что предложенный алгоритм наиболее эффективным образом позволил отфильтровать исходные данные, существенно уменьшив плотность трека в движении и верно определив все три стоянки пользователя рядом и внутри здания.

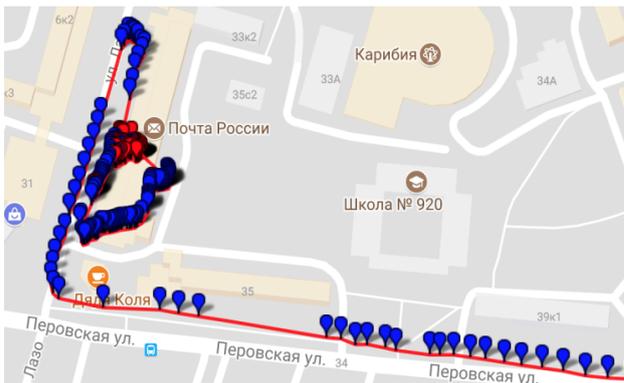
На Рисунке 4.6 представлены треки, состоящие из движения и продолжительной стоянки пользователя в здании. Предложенный алгоритм фильтрации позволил отфильтровать практически все выбросы в месте стоянки пользователя в здании, однако большая часть трека движения была исключена из результирующего набора координат по причине того, что все координаты в движении здесь имеют высокое значение параметра точности от 50 м и более, таким образом фильтр по точности, стоящий в начале каскада с порогом 50 м, не пропустил эти координаты.



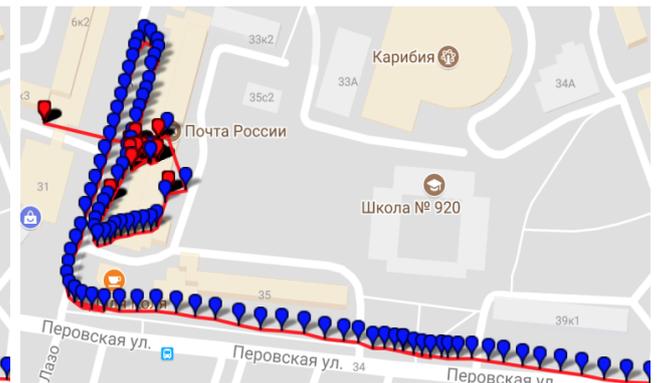
(а) исходные данные



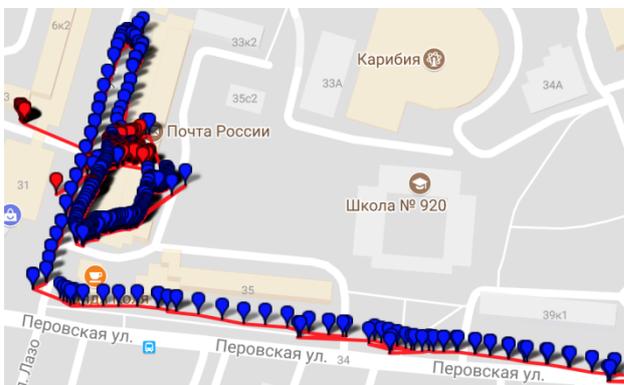
(б) реальный маршрут



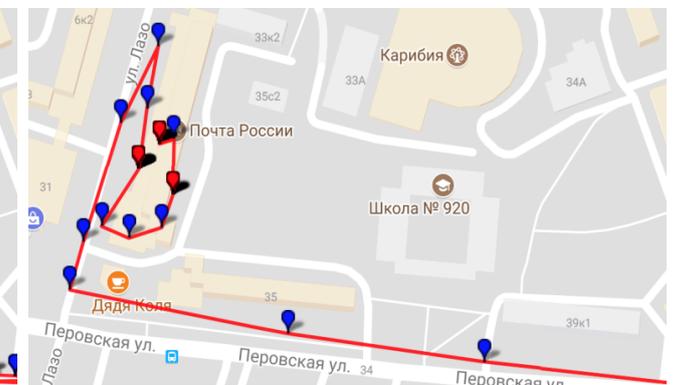
(в) API геолокация совместно с фильтром по точности



(г) диагностическая фильтрация совместно с фильтром по точности



(д) фильтр Калмана совместно с фильтром по точности

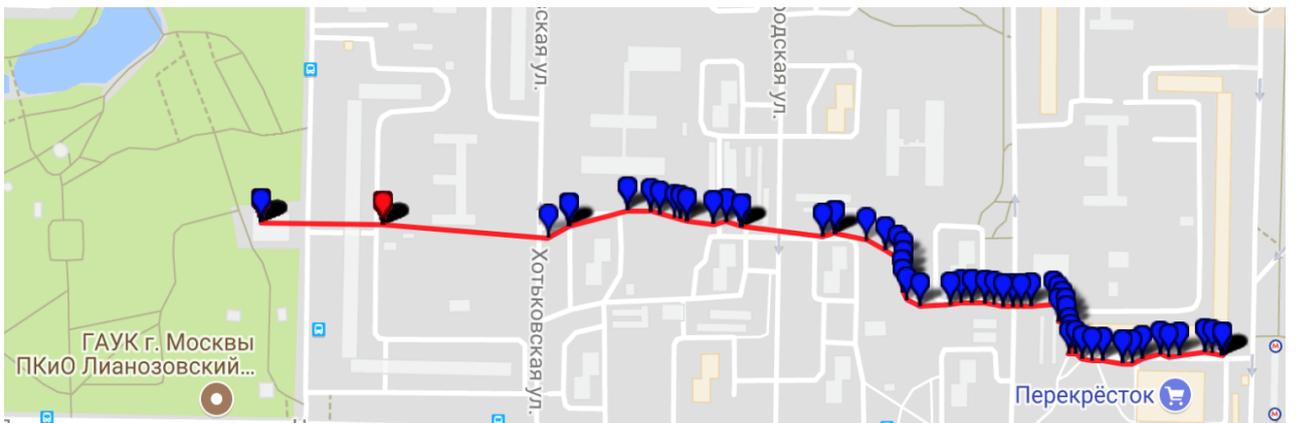


(е) предложенный алгоритм

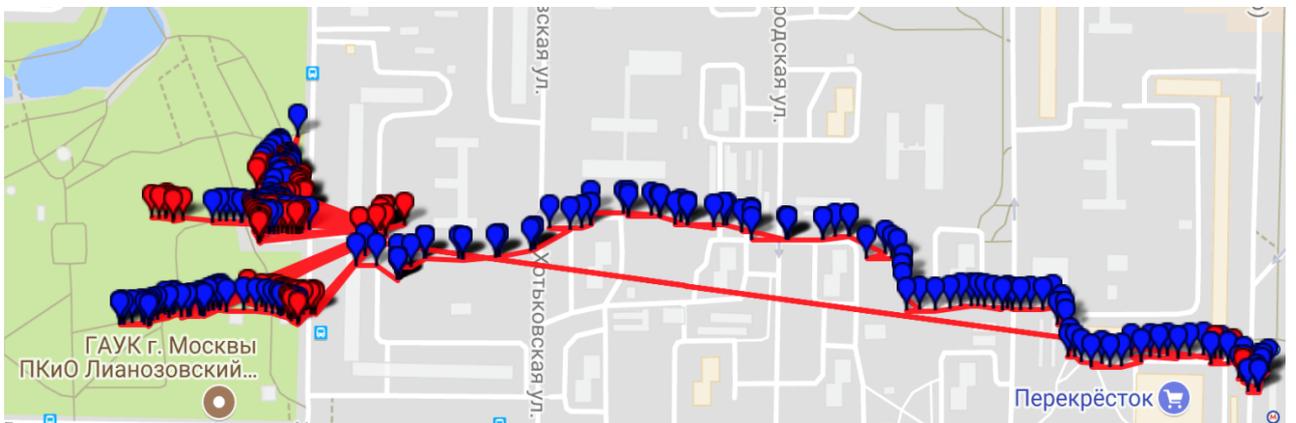
Рисунок 4.5. Экспериментальные результаты фильтрации трека (1)



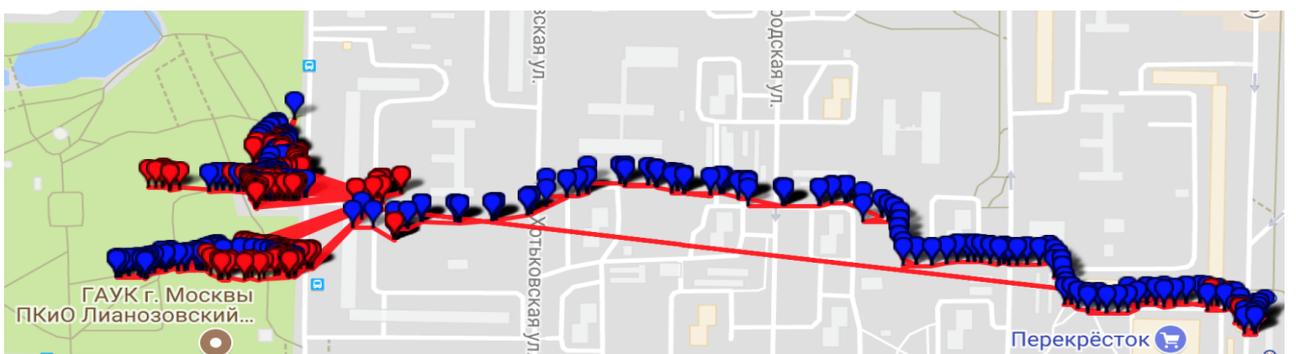
(а) исходные данные



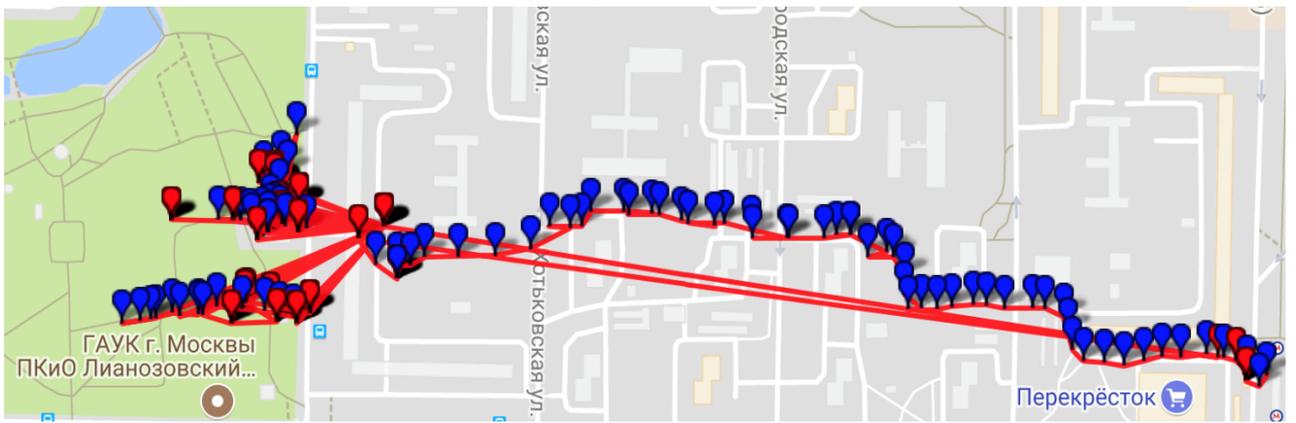
(б) реальный маршрут



(в) API геолокация совместно с фильтром по точности



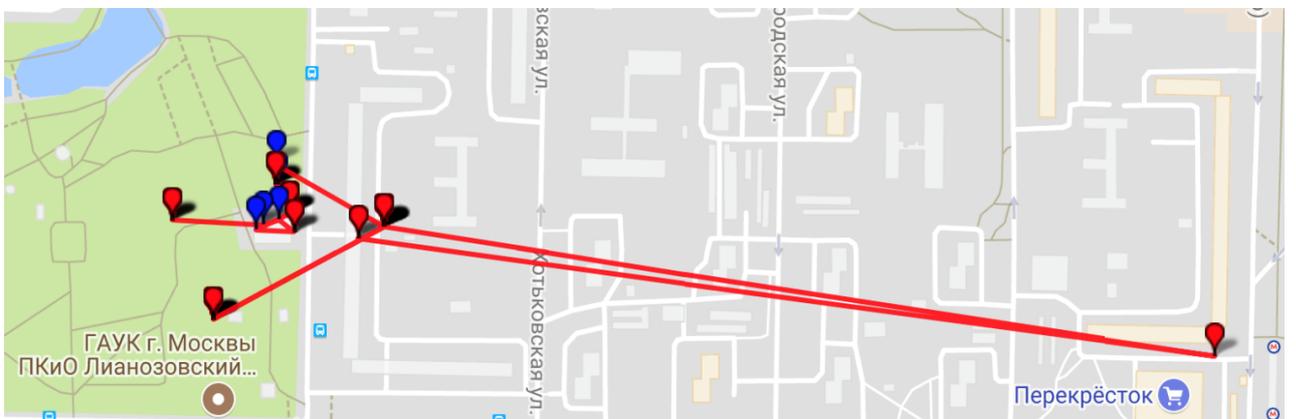
(г) Fused геолокация совместно с фильтром по точности



(д) диагностическая фильтрация совместно с фильтром по точности



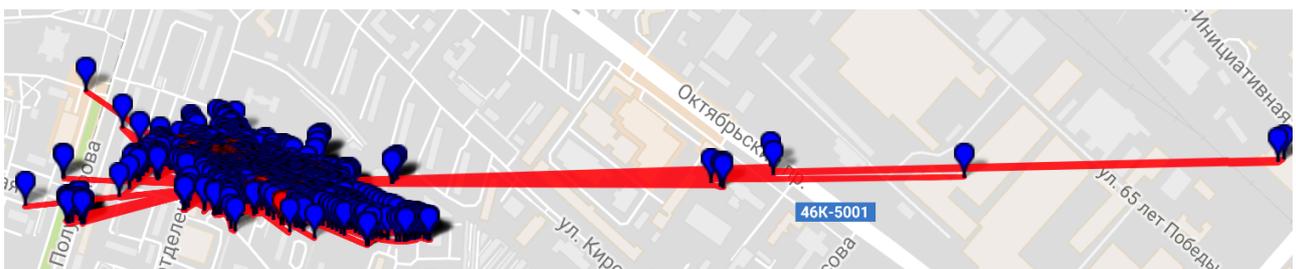
(е) фильтр Калмана совместно с фильтром по точности



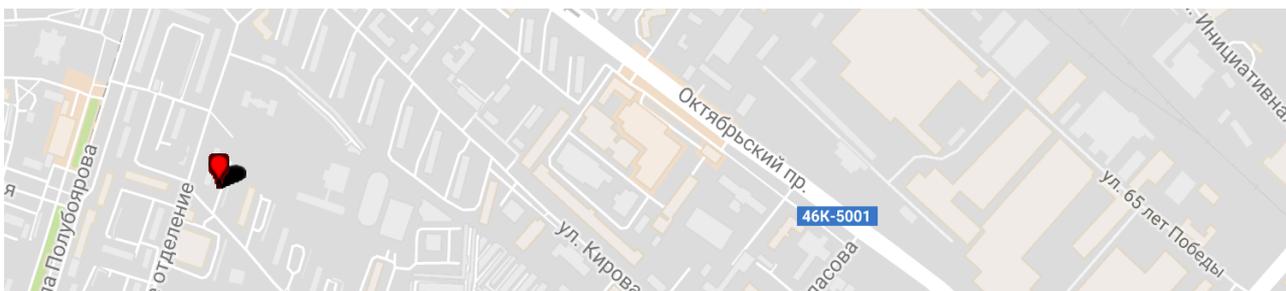
(ж) предложенный алгоритм

Рисунок 4.6. Экспериментальные результаты фильтрации трека (2)

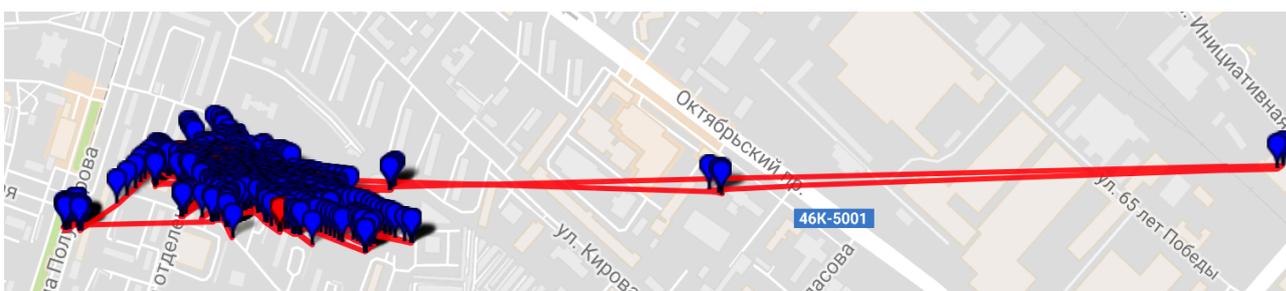
На Рисунке 4.7 представлена продолжительная стоянка в здании.



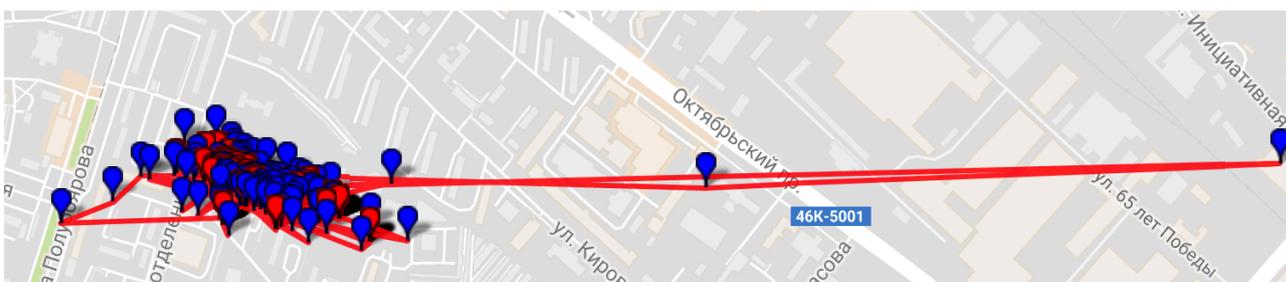
(а) исходные данные



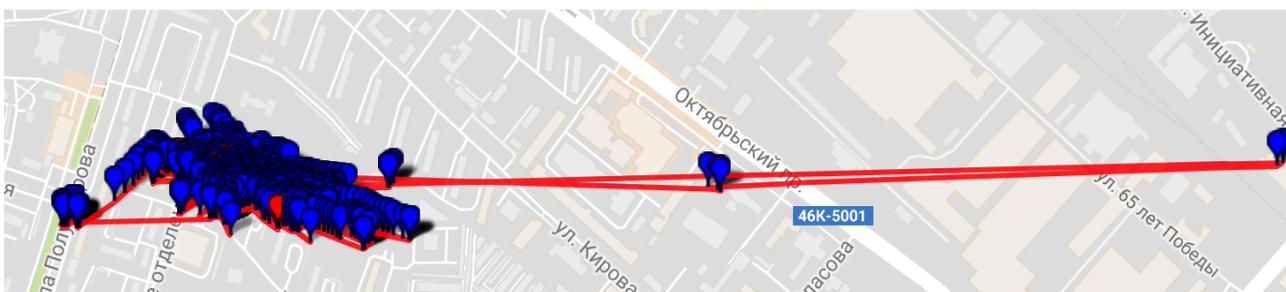
(б) реальный маршрут



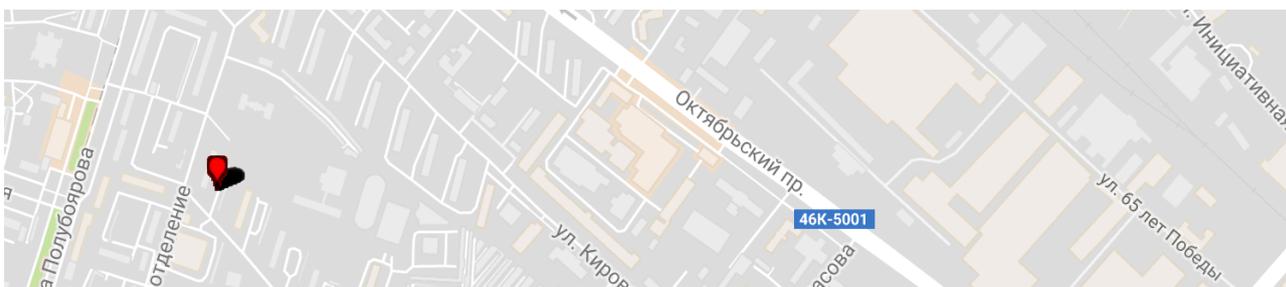
(в) API геолокация совместно с фильтром по точности



(г) диагностическая фильтрация совместно с фильтром по точности



(д) фильтр Калмана совместно с фильтром по точности

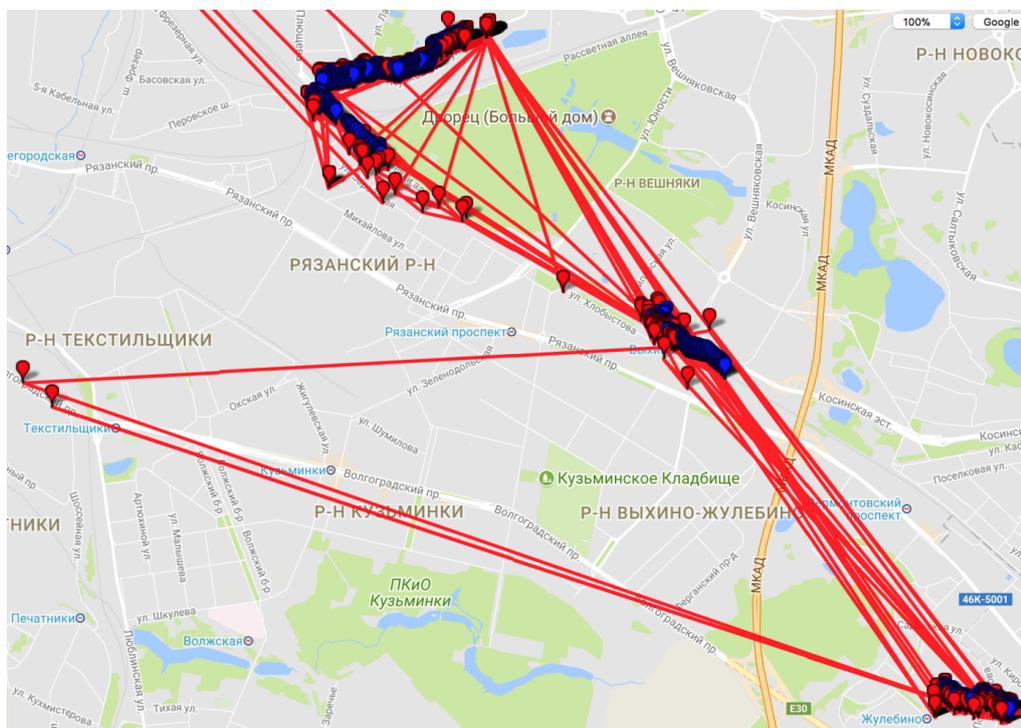


(е) предложенный алгоритм

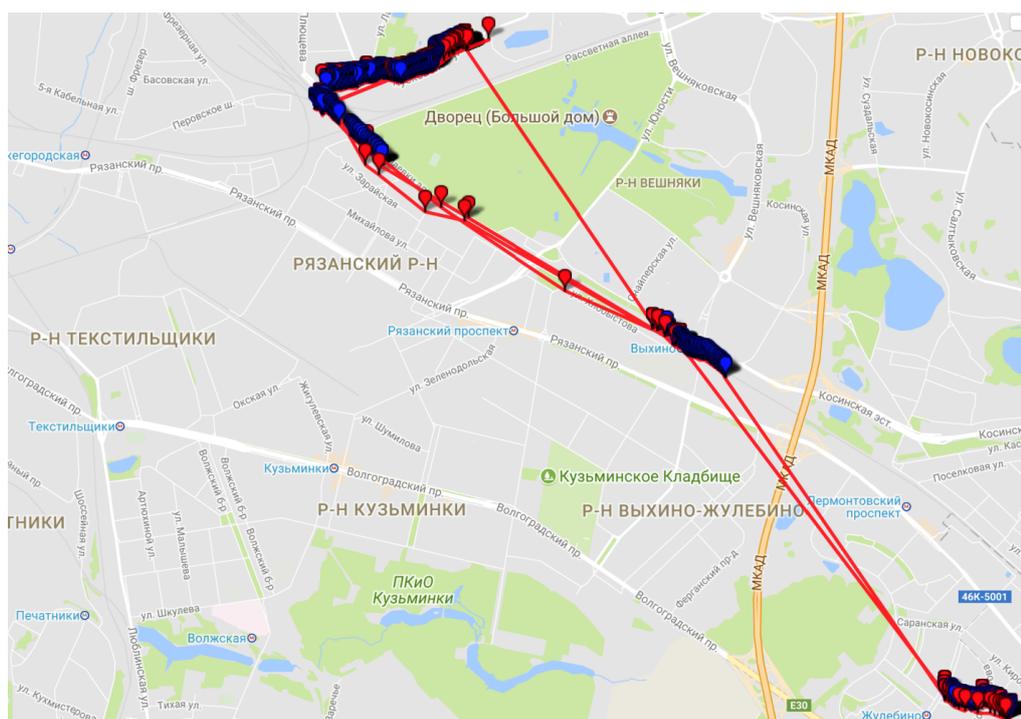
Рисунок 4.7. Экспериментальные результаты фильтрации трека (3)

Из визуального анализа треков следует, что предложенный алгоритм верно определил стоянку всего одной точкой в треке, полностью исключив все выбросы и отклонения геокоординат из исходных данных.

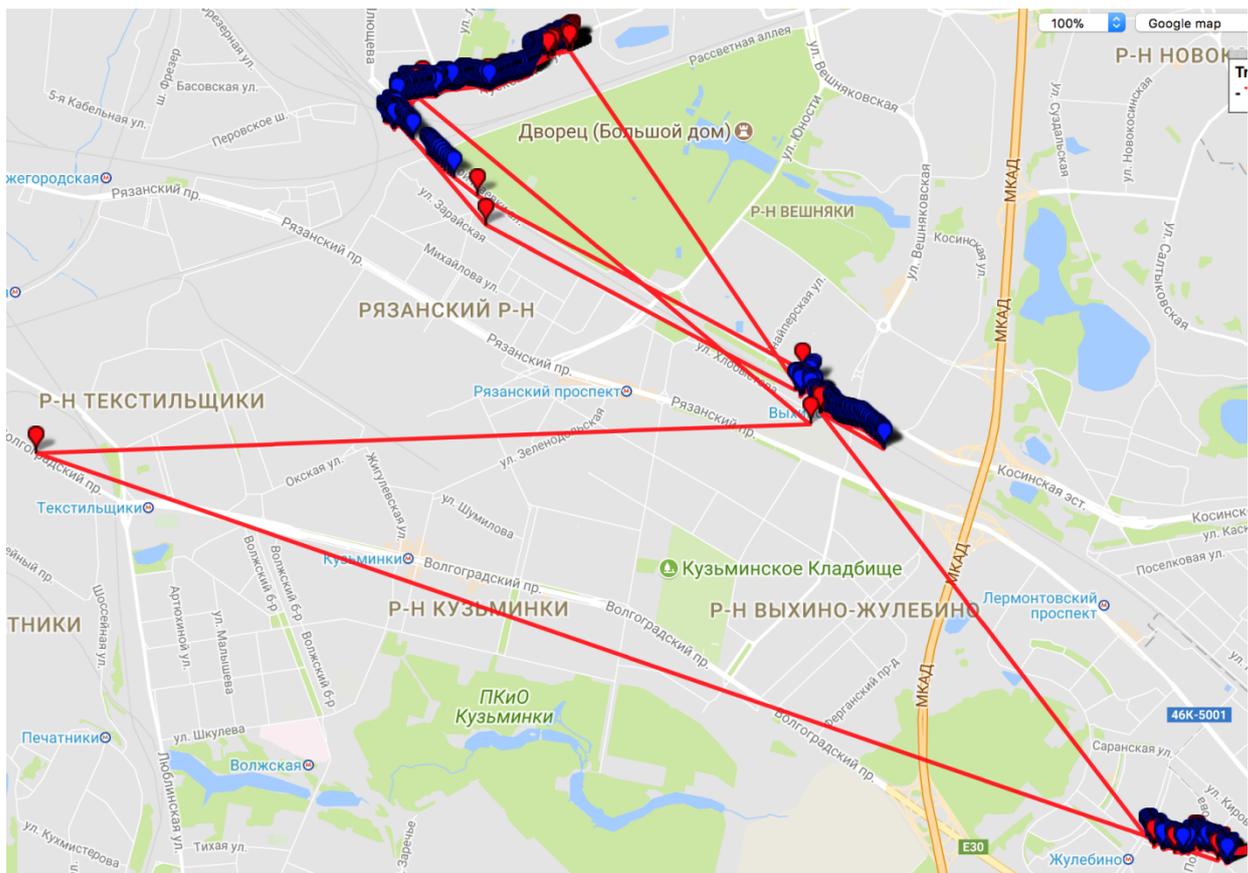
На Рисунке 4.8 показаны треки движения и стоянок пользователя со значительными выбросами геокоординат.



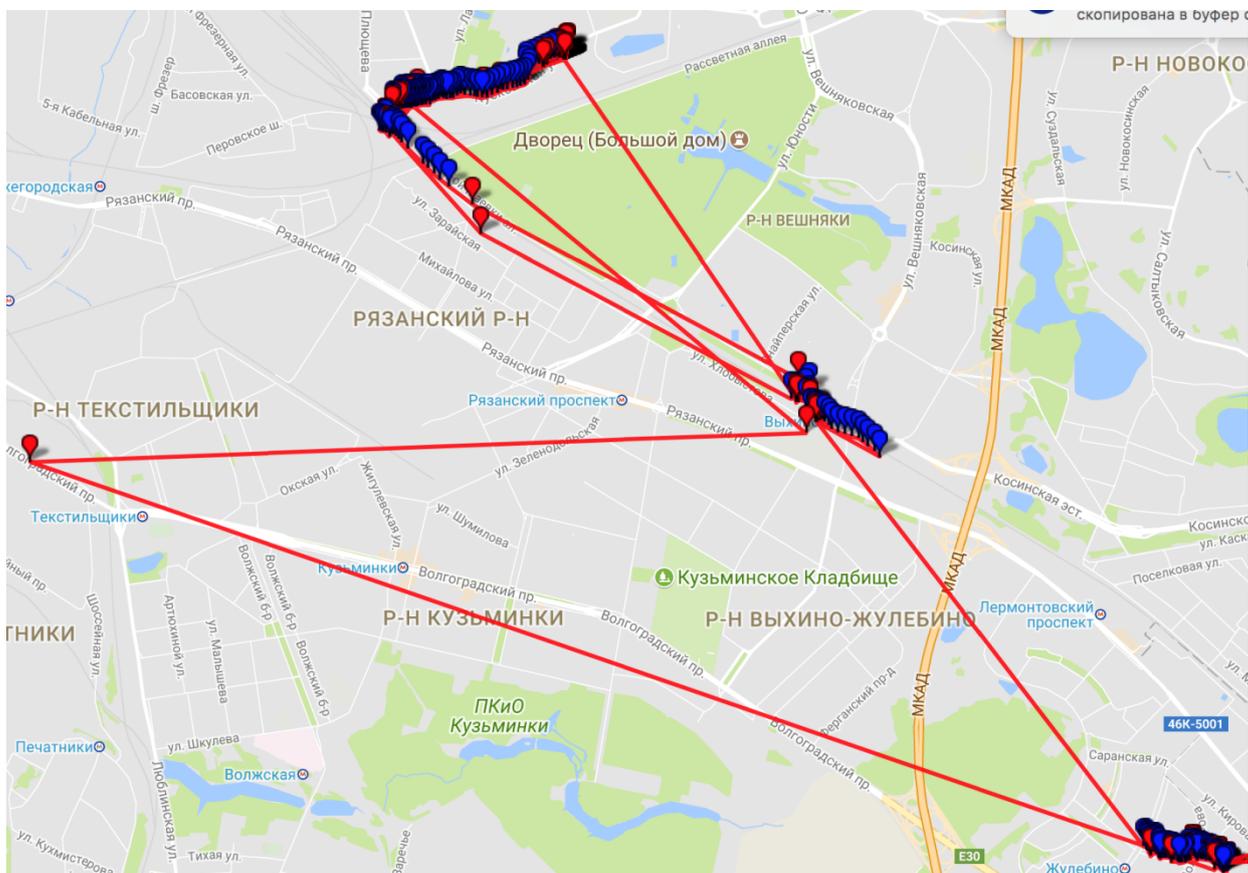
(а) исходные данные



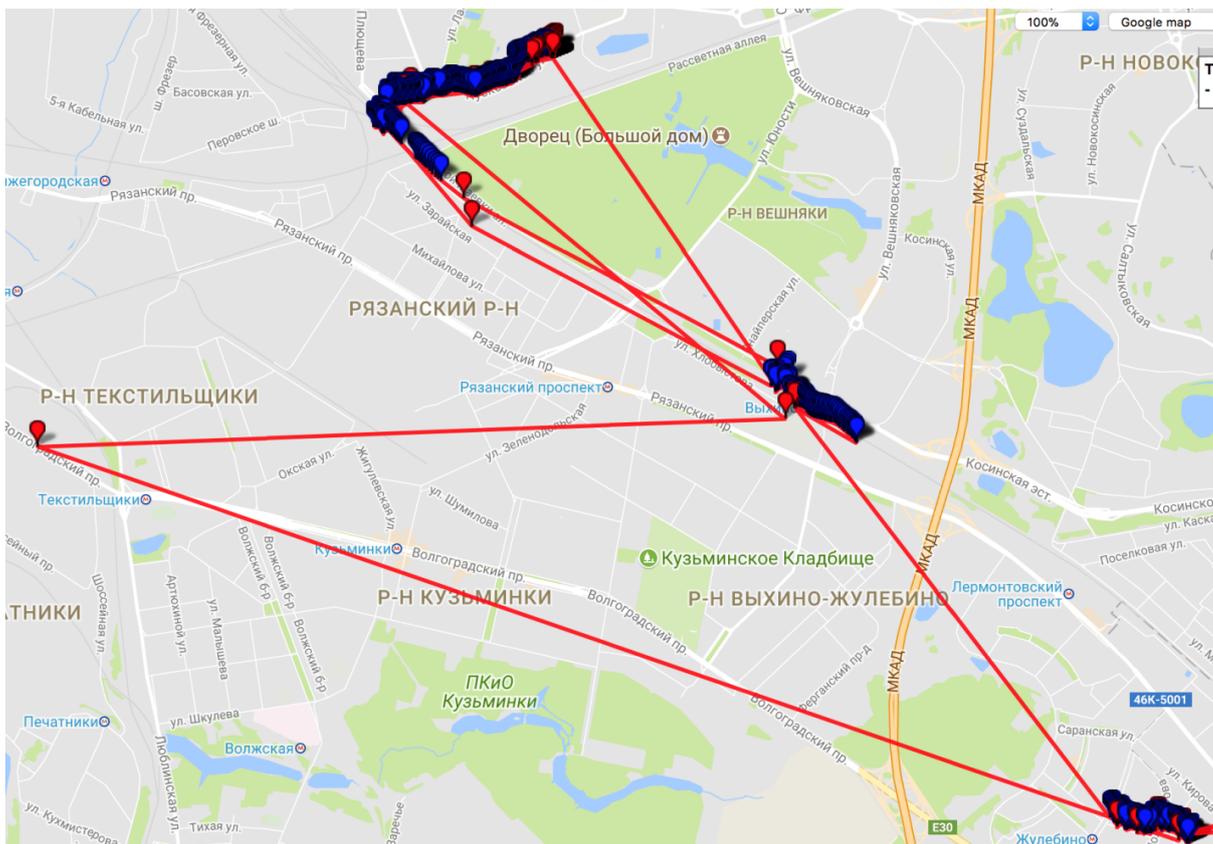
(б) реальный маршрут



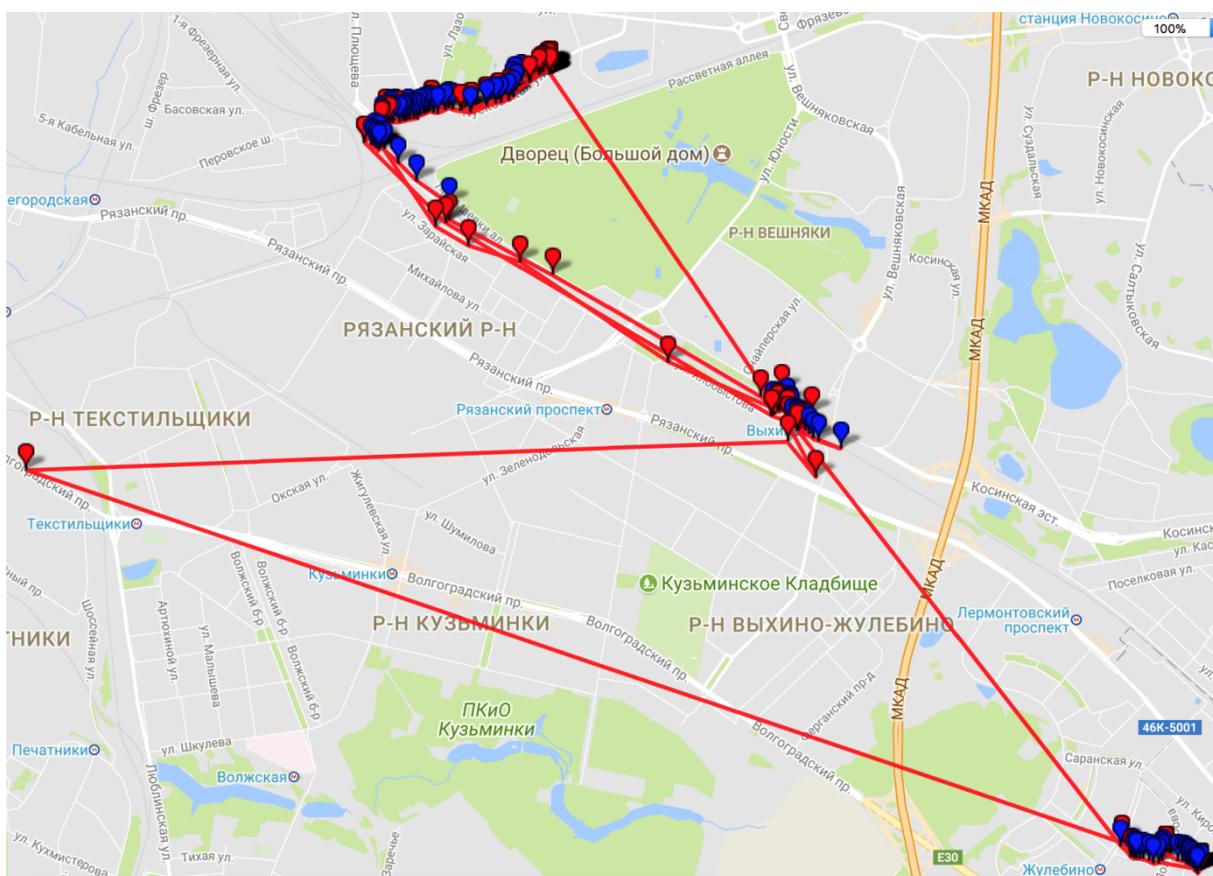
(в) API геолокация совместно с фильтром по точности



(г) диагностическая фильтрация совместно с фильтром по точности



(д) фильтр Калмана совместно с фильтром по точности

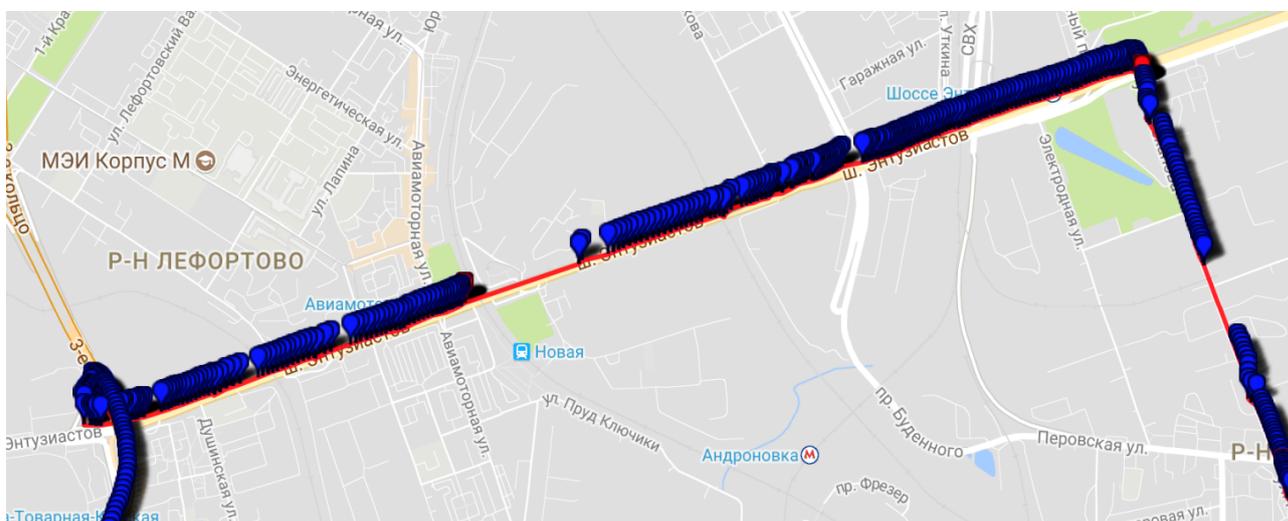


(е) предложенный алгоритм

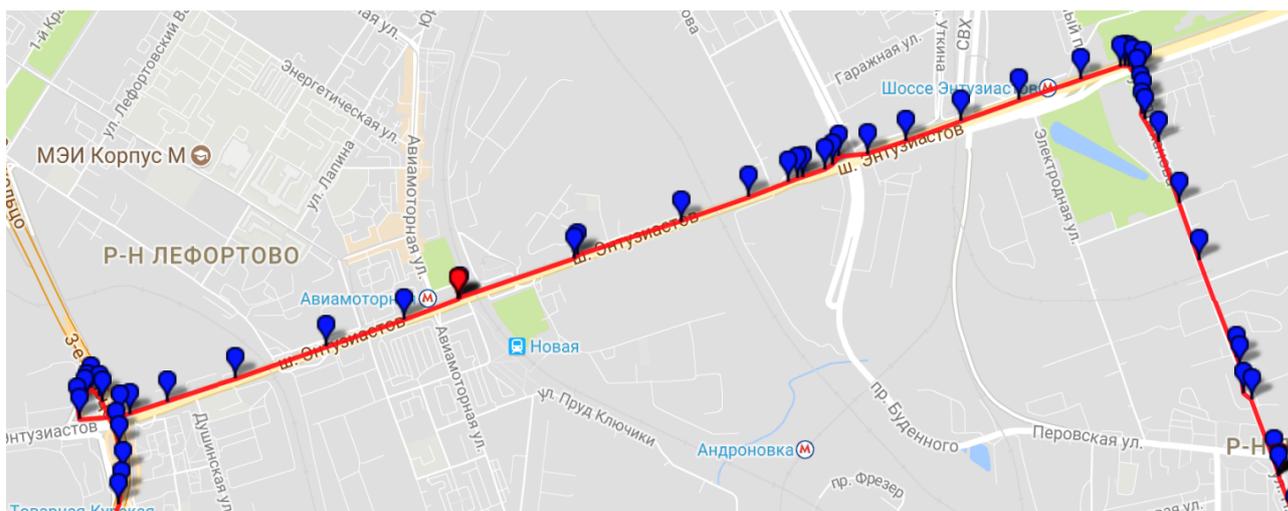
Рисунок 4.8. Экспериментальные результаты фильтрации трека (3)

По рисункам видно, что ни один метод не смог отфильтровать точку стоянки пользователя, являющуюся выбросом, в районе метро Текстильщики. Данная точка имеет параметр точности, равный 27м, и является сетевой координатой, полученной от сотовых вышек. Координата явно определена с большой погрешностью, однако из-за низкого значения параметра точности, определенного недостоверно, не представляется возможным ее исключить любыми существующими алгоритмами.

На Рисунках 4.9 показан реальный маршрут движения пользователя на автомобиле и трек после фильтрации.



(а) реальный маршрут



(б) предложенный алгоритм

Рисунок 4.9. Экспериментальные результаты фильтрации трека (4)

Исходя из анализа треков можно сделать вывод о значительном уменьшении плотности точек на прямых участках маршрута пользователя.

Выводы по главе 4

Предложенный алгоритм комплексной фильтрации потока геокоординат реализован в виде двух библиотек на языке Java, которые позволяют эффективно осуществлять фильтрацию геокоординат как в мобильных приложениях, так и в серверных модулях ГИС.

Для оценки эффективности разработанного алгоритма были собраны треки пользователей, состоящие из порядка 1 млн точек, представляющие стоянки, движения пешком, на автомобиле, поезде, метро, велосипеде. Из всех координат были сформированы тестовые наборы данных, в которых наилучшим образом отражены проблемы выбросов и отклонений координат от реальных маршрутов движения.

Разработан набор оценок качества фильтрации: по пройденному расстоянию, плотности точек в треке, времени обработки, быстродействию, точности и среднеквадратичной ошибке модели. На основе предложенного набора оценок была определена эффективность работы алгоритма фильтрации. По результату анализа показателей качества в сравнении с существующими методами обработки и экспертного визуального анализа треков можно сделать следующие выводы:

- 1) предложенный алгоритм является лучшим из рассмотренных по оценке пройденного расстояния, которая является одной из самых достоверных оценок качества фильтрации. Среднее значение отклонения составляет 943 метра, что 29.558 раз меньше наиболее эффективной технологии Fused геолокации Android. Минимальное значение отклонения составляет 0 метров, что означает полное соответствие пройденной дистанции по результирующему треку с расстоянием по реальному маршруту пользователя в некоторых треках;

- 2) предложенный алгоритм снижает плотность точек в треке на 65.916 %, что на 11.469 % больше, чем стандартное API геолокации Android совместно с фильтром по точности;
- 3) значения показателей среднего времени между координатами на выходе системы и между получением данных на вход и поступлением координаты на выход для предложенного алгоритма составляют 16.686 с и 14.260 с против 3.081 с и 8.535 с у технологии Fused геолокации совместно с фильтром по точности;
- 4) среднее время обработки в предложенном алгоритме сравнимо с фильтром Калмана, и составляет 3.444 мс, но за счет каскада фильтров стандартное отклонение ожидаемо увеличилось примерно в 3.4 раза, однако это совершенно несущественно для современных мобильных устройств. Минимальное значение времени обработки принимает значение 1 мкс;
- 5) показатели достоверности, точности, полноты и мера F_1 имеют наибольшие значения для предложенного алгоритма. Достоверность составляет 99.164 %, что на 1.39 % больше достоверности технологии диагностической фильтрации совместно с фильтром по точности. Точность равна 99.836 % (+0.111 %), полнота – 97.704 % (+1.623 %), F_1 мера принимает значение 0.988, против 0.979;
- 6) среднеквадратическая ошибка, как и максимальное значение минимальны для технологии Fused геолокации совместно с фильтром по точности и составляют 38.220 м и 1104.327 м соответственно, против 97.686 м и 11203.043 м у предложенного алгоритма, однако среднее значение ошибки минимально у рассмотренной системы фильтрации – 13.108 м, против 13.788 м.

Общие выводы и заключение

В процессе выполнения диссертации получены новые научные результаты в области увеличения точности геопозиционирования на мобильных устройствах в реальном времени:

- 1) на основе результатов анализа треков выявлены причины и характер их отклонений от реальных маршрутов движения, исследованы и классифицированы различные типы «выбросов» геокоординат, определены признаки отклонений геокоординат различных источников;
- 2) разработана формальная постановка задачи увеличения степени соответствия трека реальному маршруту движения, что позволило систематизировать исходные данные и выбрать пути решения;
- 3) по результатам анализа источников разработан метод фильтрации и модель описания входных и выходных данных системы обработки геолокационных данных, выполнен статистический анализ собранных исходных геолокационных данных, выявлены зависимости между параметрами геокоординат, в результате получены расчетные соотношения для исключения «выбросов» геокоординат при построении трека;
- 4) разработан алгоритм последовательной обработки потока геоданных, ориентированный на мобильные операционные системы, с целью исключения координат с высокой погрешностью определения, сглаживания и уменьшения плотности точек треков в реальном масштабе времени; рассчитаны оценки времен выполнения, необходимой памяти и вычислительной сложности, проведен сравнительный анализ быстродействия ключевых частей алгоритма;
- 5) разработанный алгоритм реализован в виде двух библиотек на языке Java, которые позволяют эффективно осуществлять фильтрацию геокоординат как в мобильных приложениях, так и в серверных модулях ГИС;
- 6) разработан набор критериев оценки качества фильтрации: по расстоянию, плотности точек, времени обработки, быстродействию, точности и

среднеквадратичной ошибке;

- 7) с учетом разработанных критериев на основе сформированных репрезентативных тестовых наборов данных экспериментально определено, что предложенный алгоритм фильтрации обладает достоверностью 99.164 %, что на 1.39 % больше показателя для существующих рассмотренных технологий фильтрации геоданных. Точность классификации составляет 99.836 % (+0.111 %), полнота – 97.704 % (+1.623 %), F_1 мера принимает значение 0.988, против 0.979. Среднее значение отклонения пройденного расстояния составляет 943 метра, что 29.558 раз меньше наиболее эффективной технологии Fused геолокации Android, при этом минимальное значение отклонения составляет 0 метров, что означает полное соответствие пройденной дистанции по результирующему треку с расстоянием по реальному маршруту пользователя в некоторых треках. Предложенный алгоритм снижает плотность точек в треке на 65.916 %, что на 11.469 % больше существующих рассмотренных систем фильтрации. Значения показателей среднего времени между координатами на выходе системы и между получением данных на вход и поступлением координаты на выход для предложенного алгоритма составляют 16.686 с и 14.260 с против 3.081 с и 8.535 с у технологии Fused геолокации совместно с фильтром по точности. Среднее время обработки сравнимо с фильтром Калмана, и составляет 3.444 мс, однако за счет каскада фильтров стандартное отклонение ожидаемо увеличилось примерно в 3.4 раза, что несущественно для современных мобильных устройств. Минимальное значение времени обработки принимает значение 1 мкс. Среднее значение ошибки минимально у рассмотренной системы фильтрации – 13.108 м, против 13.788 м у лучшего из существующих рассмотренных методов.

Разработанными в диссертации моделями, методами и алгоритмами обработки геоданных удалось решить часть задач увеличения точности геопозиционирования на мобильных устройствах, оставшиеся являются перспективными направлениями дальнейших исследований.

Список сокращений и условных обозначений

ГИС – Гео Информационная Система

ОС – Операционная система

ТОиР – Техническое Обслуживание и Ремонт

API (англ. Application Programming Interface) – интерфейс прикладного программирования

FMCG (англ. Fast Moving Consumer Goods) – быстрооборачиваемые потребительские товары

ESN (англ. Echo State Network) – эхо-сеть, разновидность нейронной сети

FIFO – (англ. First In First Out) – очередь «первым пришёл — первым ушёл»

Galileo – совместный проект спутниковой системы навигации Европейского союза и Европейского космического агентства

GLONASS (рус. ГЛОНАСС) – ГЛОбальная НАвигационная Спутниковая Система, российская спутниковая система навигации

GNSS (англ. Global Navigation Satellite Systems) – глобальная навигационная спутниковая система

GPS (англ. Global Positioning System) – система глобального позиционирования

GRU (англ. Gated Recurrent Unit) - англ. Gated Recurrent Units, разновидность нейронной сети

GSM (англ. Global System for Mobile Communications) глобальный стандарт цифровой мобильной сотовой связи

HDOP (hdop) (англ. Horizontal Dilution of Precision) – горизонтальный фактор потери точности

LSTM (англ. Long Short-Term Memory) – долгая краткосрочная память, разновидность нейронной сети

NMEA (англ. National Marine Electronics Association) – стандарт, текстовый протокол связи навигационного оборудования

NTM (англ. Neural Turing Machine) – нейромашин Тьюринга, разновидность нейронной сети

RNN (англ. Recurrent Neural Network) – рекуррентная нейронная сеть, разновидность нейронной сети

WFPS (англ. Wi-Fi Protected Setup) – стандарт создания беспроводной сети Wi-Fi

WiPS (англ. Wireless Intrusion Prevention System) – технология предотвращения угроз в беспроводных сетях

WLAN (англ. Wireless Local Area Network) – беспроводная локальная сеть

Список литературы

1. Головков А.А., Иванова Г.С. Адаптивная фильтрация потока геолокационных данных в реальном времени // Наука и Образование. МГТУ им. Н.Э. Баумана. № 4. 2016. С. 156-169. URL. <http://technomag.bmstu.ru/doc/835995.html> (дата обращения 07.11.2016).
2. Хруль С.А., Сонькин Д. М. Адаптивный алгоритм обработки потока навигационных данных на основе метода диагностической фильтрации // Известия Томского политехнического университета. № 5. 2012. С. 217-222. URL. <http://cyberleninka.ru/article/n/adaptivnyy-algoritm-obrabotki-potoka-navigatsionnyh-dannyh-na-osnove-metoda-diagnosticheskoy-filtratsii> (дата обращения 07.11.2016).
3. Сетевой ресурс. Спецификация NMEA. <http://www.gpsinformation.org/dale/nmea.htm>.
4. Листеренко Р. Р. Применение фильтра Калмана для обработки последовательности GPS-координат // Молодежный научно-технический вестник. № 9. 2015. 7с. URL. <http://sntbul.bmstu.ru/doc/802893.html> (дата обращения 26.01.2016).
5. Забегаев А.Н., Павловский В.Е. Адаптация фильтра Калмана для использования с локальной и глобальной системой навигации // Препринты ИПМ им. М.В.Келдыша. № 82. 2010. 24с. URL. <http://library.keldysh.ru/preprint.asp?id=2010-82> (дата обращения 26.01.2016).
6. Deriving Personal Trip Data from GPS Data: A Literature Review on the Existing Methodologies / Lei Gong [et al.] // Procedia-Social and Behavioral Sciences. № 138. 2014. P. 557-565. URL. <http://www.sciencedirect.com/science/article/pii/S1877042814041597> (accessed 26.01.2016).
7. Algorithms for GPS operation indoors and downtown / Nainesh Agarwal [et al.] // GPS Solutions. № 6-3. 2002. P. 149-160. URL. <http://link.springer.com/article/10.1007%2Fs10291-002-0028-0> (accessed 26.01.2016).

8. Mining Personally Important Places from GPS Tracks / Changqing Zhou [et al.] // Data Engineering Workshop, 2007 IEEE 23rd International Conference on. 17-20 April 2007. P. 517-526. URL. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=4401037&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxppls%2Fabs_all.jsp%3Farnumber%3D4401037 (accessed 26.01.2016).
9. Великанова Е.П., Ворошилин Е.П. Адаптивная фильтрация координат маневрирующего объекта при изменениях условий передачи в радиолокационном канале // Доклады Томского государственного университета систем управления и радиоэлектроники. № 2-1(26). 2012. С. 29-35. URL. <http://www.tusur.ru/filearchive/reports-magazine/2012-26-1/029.pdf> (дата обращения 26.01.2016).
10. Макаренко Г.К., Алешечкин А.М. Исследование алгоритма фильтрации при определении координат объекта по сигналам спутниковых радионавигационных систем // Доклады Томского государственного университета систем управления и радиоэлектроники. № 2-2(26). 2012. С. 15-18. URL. <http://cyberleninka.ru/article/n/issledovanie-algoritma-filtratsii-pri-opredelenii-koordinat-obekta-po-signalam-sputnikovyh-radionavigatsionnyh-sistem> (дата обращения 26.01.2016).
11. Прохорцов А. В., Савельев В. В. Методы определения координат и скорости подвижных объектов с помощью спутниковых радионавигационных систем // Известия Тульского государственного университета. Технические науки. № 2. 2011. С. 264-274. URL. <http://cyberleninka.ru/article/n/metody-opredeleniya-koordinat-i-skorosti-podvizhnyh-obektov-s-pomoschyu-sputnikovyh-radionavigatsionnyh-sistem> (дата обращения 26.01.2016).
12. Пудловский В.Б. Методы и алгоритмы навигационных определений с использованием ретранслированных сигналов спутниковых радионавигационных систем: дисс... канд.тех.наук. М. 2009. 263с.

13. A Kalman Filter Implementation for Precision Improvement in Low-Cost GPS Positioning of Tractors / Jaime Gomez-Gil [et al.] // Sensors. № 13. 2013. P. 15307-15323. URL. www.mdpi.com/1424-8220/13/11/15307/pdf (accessed 26.01.2016).
14. Садковский Б.П., Садковская Н.Е., Трошкин Д.О. Взаимодействие систем глобального позиционирования ГЛОНАСС/GPS и наземных вычислительных центров // Инженерный журнал: наука и инновации. № 6(30). 2014. 6с. URL. <http://engjournal.ru/catalog/it/nav/1284.html> (дата обращения 26.03.2016).
15. Салычев О.С. MEMS/GPS — малогабаритная интегрированная навигационная система // Геопрофи. № 3. 2013. С. 16-17. URL. http://www.geoprofi.ru/technology/article_697_10.aspx (дата обращения 26.03.2016).
16. Miao Lin, Wen-Jing Hsu. Mining GPS data for mobility patterns: A survey // Pervasive and Mobile Computing. 2014. P. 1-16. URL. https://www.researchgate.net/publication/262527747_Mining_GPS_data_for_mobility_patterns_A_survey (accessed 11.12.2016).
17. Mining GPS Data for Trajectory Recommendation / Peifeng Yin [et al.] // Advances in Knowledge Discovery and Data Mining. vol 8444. 2014. P. 50-61. URL. http://link.springer.com/chapter/10.1007/978-3-319-06605-9_5 (accessed 11.12.2016).
18. Mining regular routes from GPS data for ridesharing recommendations / Wen He [et al.] // UrbComp '12 Proceedings of the ACM SIGKDD International Workshop on Urban Computing. 2012. P. 79-86. URL. <http://dl.acm.org/citation.cfm?id=2346510> (accessed 11.12.2016).
19. T. Kitasuka, T. Nakanishi, A. Fukuda. Wireless LAN based indoor positioning system WiPS and its simulation // IEEE Pacific Rim Conference on Communications Computers and Signal Processing. 2003. URL. <http://ieeexplore.ieee.org/document/1235770/> (accessed 11.12.2016).
20. T. Kitasuka, T. Nakanishi, A. Fukuda. Design of WiPS: WLAN-Based Indoor Positioning System // Korea Multimedia Society. 2003. № 7(4). P. 15-29. URL.

- https://scholar.google.com/citations?view_op=view_citation&hl=en&user=POQSBf4AAAAJ&citation_for_view=POQSBf4AAAAJ:u-x6o8ySG0sC (accessed 11.12.2016).
21. Сетевой ресурс. iBeacon. <https://developer.apple.com/ibeacon/>
 22. Фальков Е. В., Романов А. Ю. Применение маячков beacon и технологии bluetooth low energy для построения систем навигации в зданиях // Новые информационные технологии в автоматизированных системах. 2015. № 18. С. 62-65. URL. <http://cyberleninka.ru/article/n/primenenie-mayachkov-beacon-i-tehnologii-bluetooth-low-energy-dlya-postroeniya-sistem-navigatsii-v-zdaniyah> (дата обращения 11.12.2016).
 23. Сетевой ресурс. MyBeacon. <http://my-beacon.ru>
 24. R. Faragher, R. Harle. An Analysis of the Accuracy of Bluetooth Low Energy for Indoor Positioning Applications // Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation. 2014. P. 201-210. URL. <https://www.ion.org/publications/abstract.cfm?jp=p&articleID=12411> (accessed 11.12.2016).
 25. Harm Hermsen. Using GPS and Accelerometer Data for Rowing Race Tracking. Master Thesis. Dept of Artificial Intelligence, University of Groningen, The Netherlands. 2013. P. 53.
 26. Moustafa Youssef, Mohamed Amir Yosef, Mohamed El-Derini. GAC: Energy-Efficient Hybrid GPS-Accelerometer-Compass GSM Localization // Proceedings of the Global Telecommunications Conference (GLOBECOM). 2010. URL. <http://ieeexplore.ieee.org/document/5684304/> (accessed 11.12.2016).
 27. Weiliang Zeng, Tomio Miwa, Takayuki Morikawa. Exploring Trip Fuel Consumption by machine Learning from GPS and CAN Bus Data // Journal of the Eastern Asia Society for Transportation Studies. №13. 2015. P. 15307-15323. URL. https://www.jstage.jst.go.jp/article/easts/11/0/11_906/_article (accessed 11.12.2016).
 28. Christopher M. Bishop. Pattern Recognition and Machine Learning. Springer. 2006. 738p.

29. Huiqin Li, Gang Wu. Map Matching for Taxi GPS Data with Extreme Learning Machine // Advanced Data Mining and Applications: 10th International Conference proceedings. 2014. P. 447-460. URL. http://link.springer.com/chapter/10.1007%2F978-3-319-14717-8_35 (accessed 11.12.2016).
30. Machine Learning Techniques Applied to the Assessment of GPS Accuracy under the Forest Canopy / Celestino Ordóñez [et al.] // Journal of Surveying Engineering. № 4. 2011. P. 140-149. URL. [http://ascelibrary.org/doi/pdf/10.1061/\(ASCE\)SU.1943-5428.0000049](http://ascelibrary.org/doi/pdf/10.1061/(ASCE)SU.1943-5428.0000049) (accessed 11.12.2016).
31. Identifying active travel behaviours in challenging environments using GPS, accelerometers, and machine learning algorithms / Katherine Ellis [et al.] // Frontiers in Public Health. №2. 2014. P. 1-8. URL. https://www.researchgate.net/publication/262055313_Identifying_Active_Travel_Behaviors_in_Challenging_Environments_Using_GPS_Accelerometers_and_Machine_Learning_Algorithms (accessed 11.12.2016).
32. Christian Manasseh, Raja Sengupta. Predicting driver destination using machine learning techniques. 16th International IEEE Conference on Intelligent Transportation Systems. 2013. P. 142-147. URL. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6728224> (accessed 11.12.2016).
33. Maja Stella, Mladen Russo, Dinko Begusic. RF Localization in Indoor Environment // Radioengineering. № 21(2). 2012. P. 557-567. URL. https://www.researchgate.net/publication/289759911_RF_Localization_in_Indoor_Environment (accessed 11.12.2016).
34. Abhishek Goswami, Luis E. Ortiz, Samir R. Das. WiGEM: a learning-based approach for indoor localization // CoNEXT '11 Proceedings of the Seventh Conference on emerging Networking EXperiments and Technologies. 2011. URL. <http://dl.acm.org/citation.cfm?id=2079299> (accessed 11.12.2016).
35. An iBeacon primer for indoor localization: demo abstract / Paul Martin [et al.] // BuildSys '14 Proceedings of the 1st ACM Conference on Embedded Systems

- for Energy-Efficient Buildings. 2014. P. 190-191. URL. <http://dl.acm.org/citation.cfm?id=2675028> (accessed 11.12.2016).
36. Berk Kapicioglu. Applications of Machine Learning to Location Data. PhD Thesis. Princeton University. 2013. P. 80.
37. Гаврилов А.В. Использование фильтра Калмана для решения задач уточнения координат БПЛА // Современные проблемы науки и образования. № 1. 2015. 7с. URL. <http://www.science-education.ru/ru/article/view?id=19453> (дата обращения 11.12.2016).
38. Jeffrey L. Elman. Finding Structure in Time // Cognitive Science. № 14. 1990. P. 179-211. URL. <https://crl.ucsd.edu/~elman/Papers/fsit.pdf> (accessed 11.12.2016).
39. Hochreiter Sepp, Jürgen Schmidhuber. Long short-term memory // Neural Computation. № 9(8). 1997. P. 1737-1780. URL. http://deeplearning.cs.cmu.edu/pdfs/Hochreiter97_lstm.pdf (accessed 11.12.2016).
40. Empirical evaluation of gated recurrent neural networks on sequence modeling / Junyoung Chung [et al.] // NIPS 2014 Deep Learning and Representation Learning Workshop. 2014. 9p. URL. <https://arxiv.org/pdf/1412.3555v1.pdf> (accessed 11.12.2016).
41. Herbert Jaeger, Harald Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication // Science. №30. 2004. P. 78-80. URL. <https://pdfs.semanticscholar.org/8922/17bb82c11e6e2263178ed20ac23db6279c7a.pdf> (accessed 11.12.2016).
42. Alex Graves, Greg Wayne, Ivo Danihelka. Neural Turing Machines // arXiv preprint arXiv:1410.5401. 2014. 29p. URL. <https://arxiv.org/pdf/1410.5401v2.pdf> (accessed 11.12.2016).
43. Сетевой ресурс. Использование FusedLocationProviderApi в Android ОС. <https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi>

44. Головков А.А., Иванова Г.С. Метод оценки эффективности фильтрации геокоординат // Современные наукоемкие технологии. № 5. 2018. С. 51-55.
45. Головков А.А. Методика обработки геоданных в мобильных устройствах // Динамика сложных систем - XXI век. № 2. 2018. С. 4-13.
46. Головков А.А. Источники геоданных в мобильных устройствах // Динамика сложных систем - XXI век. № 4. 2017. С. 94-101.
47. Головков А.А. Особенности обработки данных местоположения в геоинформационных системах // Научные исследования в области технических наук. Сборник научных трудов по итогам международной научно-практической конференции. № 3. 2018. С. 66-69.
48. Головков А.А., Иванова Г.С. Обработка геолокационной информации как практическая задача машинного обучения. Управление качеством инженерного образования. Возможности ВУЗов и потребности промышленности»: Тезисы докладов второй международной научно-практической конференции. М.: НУК ИУ МГТУ им. Н.Э. Баумана, Москва, 2016. С. 314.
49. Сетевой ресурс. WGS 84. https://geostart.ru/wgs_84
50. Сетевой ресурс. Haversine formula. <https://www.movable-type.co.uk/scripts/latlong.html>
51. Хействер А.А., Костюк В.П. Метод фильтрации геоданных в системе выбора рационального маршрута движения транспортом // Математические методы в технике и технологиях - ММТТ. 2014. № 4 (63). С. 95-97.
52. Костюк В.П., Хействер А.А. Разработка и исследование моделей и методов маршрутизации движения пассажиров общественным транспортом // Вестник Саратовского государственного технического университета. 2015. Т. 2. № 1 (79). С. 103-109.
53. Дементьев В.Е., Кондратьев Д.С. Разработка алгоритмов навигации автономных летательных аппаратов на основе комплексирования разнородных навигационных данных // В сборнике: Четырнадцатая национальная конференция по искусственному интеллекту с международным участием. КИИ-2014 Труды конференции. 2014. С. 256-264.

54. Бейлина Н.В. Блочно-временной алгоритм фильтрации геолокационных данных // Вестник Самарского университета. Естественнонаучная серия. 2013. № 9-1 (110). С. 212-215.
55. Гитис В.Г. Геоинформационные технологии для научных исследований // Вестник Российского фонда фундаментальных исследований. 2011. № 2-3 (70-71). С. 13-32.
56. Хенрик Бринк, Джозеф Ричардс, Марк Феверолф. Машинное обучение. Питер. 2017. 336с.
57. Зайнуллин Р.И., Гузаиров М.Б. Новые подходы к хранению и обработке больших массивов геоданных // Инновации и инвестиции. 2014. № 12. С. 164-166.
58. Сидоров И.Ю., Армяков А.О., Байтин А.А., Гунин А.В. Хранилище точечных геобъектов // В сборнике: Минцевские чтения Вторая Всероссийская научно-техническая конференция молодых конструкторов и инженеров, посвящена 120-летию со дня рождения академика А.Л. Минца и 60 летию аспирантуры Радиотехнического института. 2015. С. 234-247.
59. Семенова Л.Л. Современные методы навигации беспилотных летательных аппаратов // Наука и образование сегодня. 2018. № 4 (27). С. 6-8.
60. Маркелов В.М. Добыча данных и геоданных // Образовательные ресурсы и технологии. 2015. №2 (10). С. 126-131.
61. Дышленко С.Г. Анализ и разработка характеристик качества геоданных // ПНиО. 2016. №2 (20). С. 23-27.
62. Краснобородько А.В. Исследование и разработка структур баз геоданных информационно-телекоммуникационных систем: дисс...канд.тех.наук. М. 2007. 102с.
63. Степанов О.А., Торопов А.Б., Носов А.С. Алгоритмы фильтрации в задаче навигации подвижных объектов по геофизическим полям // Материалы XXX конференции памяти выдающегося конструктора гироскопических приборов Н.Н. Острякова. 2016. С. 421-432.

64. Степанов О.А., Торопов А.Б. Методы нелинейной фильтрации в задаче навигации по геофизическим полям. ч.2. Современные тенденции // Гироскопия и навигация. 2016. № 1. С. 147.
65. Моторин А.В., Торопов А.Б., Степанов О.А. Многоальтернативная фильтрация применительно к задаче оценивания модели погрешностей датчиков // В сборнике: Навигация и управление движением. Материалы XVII конференции молодых ученых. 2015. С. 267-274.
66. Розенберг И.Н., Дулин С.К. Интеллектуализация формирования сетевых ресурсов геоинформационного портала // Материалы 10-й Всероссийской мультikonференции по проблемам управления (МКПУ-2017). 2017. С. 54-57.
67. Автоматизация сбора и обработки пространственных данных железнодорожной инфраструктуры / Уманский В.И. [и др.] // Железнодорожный транспорт. № 10. 2017. С. 46-49.
68. Дулин С.К., Дулина Н.Г. Анализ неопределенности геоинформации при визуализации объектов арктической зоны, допускающей наличие активного картографического фона // Системы высокой доступности. Т. 13. № 1. 2017. С. 52-64.
69. Дулин С.К., Розенберг И.Н., Уманский В.И. Структуризация проблемы улучшения пространственной согласованности баз геоданных арктической зоны // Системы высокой доступности. Т. 13. № 3. 2017. С. 3-14.
70. Макет интероперабельной информационно-аналитической системы для обеспечения пространственного и семантического поиска и анализа геоданных / Дулин С.К. [и др.] // Системы и средства информатики. № 1. 2017. С. 46-59.
71. Дулин С.К., Дулина Н.Г., Якушев Д.А. Геоинформационный синтез пространственных данных инфраструктуры железнодорожного транспорта // Системы и средства информатики. Т. 27. № 3. 2017. С. 126-144.
72. Дулин С.К., Якушев Д.А. Точность геопро пространственных данных железнодорожной инфраструктуры, полученных методом мобильного лазерного

- сканирования // Наука и технологии железных дорог. Т. 1. № 4 (4). 2017. С. 54-62.
73. Мазуров Н.В. Разработка и исследование принципов построения центра высокопроизводительной обработки геопространственных данных: дисс...канд.тех.наук. М. 2005. 117с.
74. Семенов В.Ю. Методы определения местоположения пользователя в информационных радиосистемах в условиях многолучевого канала с угловой дисперсией: дисс...канд.физ.-мат..наук. Нижний Новгород. 2012. 129с.
75. Камалов Ю.Б. Разработка и моделирование алгоритмов определения местоположения абонента в сетях мобильной связи: дисс...канд.тех.наук. Ульяновск. 2011. 132с.
76. Ассур О.С., Филаретов Г.Ф. Разработка комплексного метода позиционирования объектов по данным беспроводных сетей Wi-Fi и устройств BLE (Bluetooth Low Energy) // Известия Института инженерной физики. Т. 2. № 36. 2015. С. 2-10.
77. Навигационный алгоритм с использованием планов зданий и данных автономных датчиков / Дэвидсон П. [и др.] // Гироскопия и навигация. № 1 (88). 2015. С. 29-42.
78. Амелин К.С. Метод ориентирования сверхлегкого бпла при редком обновлении данных о его местоположении // Стохастическая оптимизация в информатике. Т. 10. № 2. 2014. С. 3-14.
79. Егоров И.В., Гайворонский Д.В. Разработка прототипа системы высокоточного позиционирования объектов // В сборнике: Современные проблемы радиоэлектроники и телекоммуникаций "РТ - 2016" Материалы 12-ой международной молодежной научно-технической конференции. Севастопольский государственный университет. 2016. С. 33.
80. Идентификация и определение местоположения источников радиоизлучения сети WIMAX в многопозиционной радиотехнической системе разностно-дальномерным способом / Бердинских Л.Н. [и др.] // Телекоммуникации. № 7. 2016. С. 19-24.

81. Болотин Ю.В., Фатехрад М. Навигация пешехода с использованием бинс с коррекцией по нулевой скорости // В сборнике: Сборник материалов XXII Санкт-Петербургской международной конференции по интегрированным навигационным системам Главный редактор В. Г. Пешехонов. 2015. С. 49-53.
82. Болотин Ю.В., Фатехрад М. Навигация пешехода с использованием бесплатформенной инерциальной навигационной системы (БИНС), установленной на стопе // Российский журнал биомеханики. Т. 19. № 1. 2015. С. 25-36.
83. Беспроводные технологии на автомобильном транспорте. глобальная навигация и определение местоположения транспортных средств / Власов В.М. [и др.] // Учебное пособие / Москва, 2017. Сер. Высшее образование: Бакалавриат. 184с.
84. Решетников В.Н., Болодурина И.П., Нугуманова А.А. Фильтр калмана как метод вторичной обработки информации с системы ГЛОНАСС // Программные продукты и системы. № 4. 2015. С. 116-120.
85. Алгоритм навигации наземных транспортно-технологических средств на основе данных интегрированной инерциально-спутниковой системы навигации и одометра / Новиков П.В. [и др.] // Известия Московского государственного технического университета МАМИ. № 2 (32). 2017. С. 31-39.
86. Салех Х.М., Фролов И.А., Матафонов Д.И. Мобильное приложение для определения местоположения объектов внутри здания посредством Wi-Fi // В сборнике: Теоретические и прикладные вопросы науки и образования сборник научных трудов по материалам Международной научно-практической конференции: в 16 частях. 2015. С. 110-112.
87. Кротова Е.И. Метод повышения точности определения местоположения абонента мобильной связи // В сборнике: Проблемы техники и технологий телекоммуникаций ПТиТТ-2014; Оптические технологии в телекоммуникациях ОТТ-2014 Материалы Международных научно-технических конференций. 2014. С. 113-115.

88. Агафонов С.Ю., Волгушев Д.Б. Влияние количества используемых точек доступа на точность определения местоположения мобильного объекта // Информационные технологии и телекоммуникации. № 2 (6). 2014. С. 27-33.
89. Камалов Ю.Б., Служивый М.Н. Об ошибках определения местоположения мобильного абонента для гиперболической системы // REDS: Телекоммуникационные устройства и системы. Т. 4. № 3. 2014. С. 257-259.
90. Маркин Д.О., Макеев С.М. Модель системы определения местоположения мобильного устройства на основе метода статистических испытаний // Известия Тульского государственного университета. Технические науки. № 2. 2016. С. 150-165.

Акт

о внедрении результатов

диссертационной работы аспиранта кафедры «Компьютерные системы и сети»

МГТУ им. Н.Э. Баумана

Александра Александровича Головкова

Группа компаний CDC использовала в ряде программных продуктов для коммерческих организаций теоретические и практические результаты диссертационной работы Головкова А.А. в виде методов потоковой фильтрации геокоординат на мобильных устройствах и их программных реализаций. Результаты диссертации были внедрены в следующие системы:

- ОПТИМУМ АСУМТ (автоматизация мобильной торговли);
- ОПТИМУМ ГИС (автоматизация транспортной логистики);

ОПТИМУМ ММС (автоматизация технического обслуживания, ремонтов оборудования, управление ТОиР).

Применение указанных методов и программных средств позволило существенно повысить качество обработки геолокационной информации, поступающей из разнородных источников, в системах мониторинга мобильных сотрудников, достоверность определения местоположения была увеличена на 1.39 %.

Генеральный директор УК Джeneral Партнерс
Управляющая компания ГК СидиСи



Антошин Р.Б.

Акт
о внедрении результатов
диссертационной работы аспиранта кафедры «Компьютерные системы и сети»
МГТУ им. Н.Э. Баумана
Александра Александровича Головкина

Настоящий акт подтверждает то, что группа компаний СНС использует решение ОПТИМУМ ГИС, в которое входит подсистема определения местоположения сотрудников и построения маршрутов, разработанная на основе результатов диссертационной работы Головкина А.А. Решение применяется для автоматизации и мониторинга деятельности торговых представителей. С момента внедрения система позволила повысить количество визитов торговых представителей на 7% и соответственно количество продуктивных визитов на 3%.

Директор Департамента Информационных технологий
Группа компаний СНС, Сушинский А.П.



ОТЗЫВ

научного руководителя д.т.н., профессора Ивановой Г.С. об аспиранте Головкове Александре Александровиче, представившем диссертацию на тему «Алгоритм фильтрации геоданных для мобильных устройств в реальном времени» на соискание ученой степени кандидата технических наук по специальности 05.13.01 – Системный анализ, управление и обработка информации (в технических системах)

В настоящее время получили широкое распространение геоинформационные системы (ГИС), предназначенные для сбора, хранения и обработки пространственных данных. В особый класс можно выделить ГИС, использующие получаемые от навигационных спутников геоданные о маршрутах движения объектов. В качестве таких объектов в работе Головкова А.А. рассматриваются пешие и передвигающиеся на автотранспорте полевые сотрудники различных фирм, аварийные ремонтные бригады, а также беспилотная летающая техника. Объекты снабжены мобильными средствами вычислительной техники (планшетами, смартфонами), оснащенными разного рода навигационными приемниками. Управление объектами предполагает наличие достаточно точной информации о местоположении рассматриваемых объектов в реальном времени, а современные ГИС при использовании разных источников геоданных обеспечивают построение маршрутов движения объектов с ошибкой около 1000 метров, что недопустимо с точки зрения решения оперативных задач.

Существенную проблему также составляло то, что разрабатываемая ГИС для разгрузки сервера должна обеспечивать построение трека движения объекта на мобильных устройствах. А значит, обработка геоданных должна иметь сравнительно небольшую вычислительную сложность.

Тема диссертационного исследования А.А. Головкова, связанная с разработкой алгоритмов фильтрации геоданных на мобильных устройствах, соответствует приоритетному направлению развития науки, технологий и тех-

ники РФ «Информационно-телекоммуникационные системы». Основными научными результатами работы являются формальная постановка решаемой задачи, модель геоданных, получаемых от разных источников, классификация случаев отклонения трека от реального маршрута. Полученные результаты позволили использовать дополнительную информацию, которая присутствует в геоданных, получаемых от различных источников, для построения системы фильтров, обеспечивающих исключение невалидных геоданных и корректировку геоданных низкой точности.

К научным результатам также следует отнести формирование набора критериев оценки качества фильтрации и получение расчетных соотношений для их вычисления по имеющимся данным.

Указанные научные результаты послужили основой для создания программного обеспечения ГИС, экспериментальные исследования с применением которой подтвердили полученные научные результаты.

Научной работой Головков А.А. начал заниматься в 2013 г, обучаясь в университете. Первые его работы были посвящены проблематике распараллеливания сложных вычислительных процессов.

Разработкой ГИС Головков А.А. занимается с 2014 г. Собственный опыт, полученный при создании сложных информационных систем, в том числе для мобильных устройств, позволил ему четко определить требования к разрабатываемому алгоритму фильтрации геоданных и наметить пути их реализации.

В настоящее время Головков А.А. является руководителем отдела мобильных решений ООО «Си Ди Си Рус», учится в аспирантуре кафедры Компьютерные системы и сети МГТУ им. Н.Э. Баумана и по совместительству работает преподавателем на той же кафедре. Он ведет дисциплину «Параллельные вычисления» и руководит группой магистров кафедры в рамках научно-исследовательской работы по машинному зрению.

Значимость научных разработок Головкова А.А. подтверждается публикациями. Всего он имеет 12 работ, в том числе 6 статей в научных журна-

лах из списка ВАК РФ. Головков А.А. принял участие во многих конференциях как Всероссийских, так и Международных, где были апробированы основные результаты научной деятельности.

Как руководитель я особенно хотела бы отметить увлеченность, целеустремленность, большую работоспособность, широкий круг интересов и высокий уровень профессионализма диссертанта, а также тщательность проработки темы и ответственность за полученные результаты. С моей точки зрения соискатель является сложившимся научным работником, выполненная им диссертация на актуальную тему носит законченный характер.

Считаю, что диссертация удовлетворяет критериям, установленным п. 9 «Положения о присуждении ученых степеней» № 842, предъявляемым к кандидатским диссертациям, а Головков А.А. заслуживает присуждения ему ученой степени кандидата технических наук по специальности 05.13.01 – Системный анализ, управление и обработка информации (в технических системах).

Научный руководитель,
Лауреат Премии правительства
в области образования 2009 г.,
профессор кафедры «Компьютерные
системы и сети» ФГБОУ ВО
«Московский государственный
технический университет им. Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана),
доктор технических наук, профессор 
Тел.: 8 (499) 263-60-89
E-mail: gsivanova@bmstu.ru

Галина Сергеевна Иванова

Подпись профессора Г.С. Ивановой заверяю

МГТУ им. Н.Э. Баумана
1005005, Москва, 2-я Бауманская ул., д. 5, стр. 1
Телефон (499) 263-6391
Факс (499) 267-4844
Эл. почта bauman@bmstu.ru



А. Г. МАТВЕЕВ
НАЧ. УПРАВЛЕНИЯ КАДРОВ

ТЕЛ. 8499-263-67-69