

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт математики, механики и компьютерных наук
имени И. И. Воровича

Направление подготовки
02.04.02 — Фундаментальная информатика
и информационные технологии

Д. С. Людовских
Система мониторинга качества дорожного покрытия с
помощью мобильных устройств
Магистерская диссертация

Научный руководитель:
доц. к.т.н. М. И. Чердынцева
Рецензент:
к.ф.м.н. А. А. Зеленина

Ростов-на-Дону
2016

Содержание:

| | |
|---|----|
| 1. Введение | 3 |
| 2. Постановка задачи | 6 |
| 3. Анализ параметров движения с помощью мобильных устройств | 7 |
| 4. Системы мониторинга качества дорожного покрытия | 11 |
| 5. Проект системы мониторинга качества дорожного покрытия | 16 |
| 6. Выбор операционной системы для мобильных устройств | 27 |
| 7. Выбор картографического сервиса | 30 |
| 8. Аспекты реализации | 32 |
| 9. Тестирование реализованной системы | 41 |
| 10. Заключение | 43 |
| 11. Список литературы | 44 |

1. Введение

Дорожные службы по всему миру тратят миллионы долларов на контроль состояния, обслуживание и ремонт дорог. Несмотря на все эти инвестиции большинство людей недовольны качеством дорог, где они живут и работают. Причины этого заключаются в том, что плохие дороги увеличивают расход топлива и обслуживание транспорта, иногда опасны для водителей и пешеходов, и как минимум раздражают как водителей автомобилей, так и водителей мотоциклов и велосипедов. Они также являются причиной дорогостоящих судебных исков, связанных с требованием о возмещении ущерба.

Сохранение дорог в хорошем состоянии является сложной задачей, так как суровые дорожные условия, неожиданные нагрузки трафика, и нормальный износ неизбежно разрушают даже хорошо продуманные дороги в течении относительно короткого периода времени (от нескольких недель до нескольких месяцев). И поскольку, как правило, бюджеты дорожных служб ограничены, задача фиксирования дорог, которые нуждаются в ремонте, является важной. Кроме того, информирование водителей об опасных участках дороги, с точки зрения качества дорожного покрытия, была бы полезной функцией для навигационных систем, особенно в ночное время или при плохом освещении. Эта работа направлена на реализацию этих потребностей.

Мониторинг состояния дорожного покрытия это задача, которая не может быть легко решена путем установки статических датчиков на дороге. Огромные размеры дорожно-транспортной сети делают размещение статических датчиков сложной задачей с точки зрения стоимости и рабочей силы. Кроме того, естественным выглядит анализировать качество поверхности дороги при помощи измерений вибраций и импульсов во время езды. К счастью, так как состояние дороги ухудшается постепенно в течении недель, то достаточно проверять качество дорожного покрытия

несколько раз в день или даже реже. Добавление недорогого и простого в установке оборудования по сбору данных, делает мониторинг качества дорожного покрытия побочным эффектом нормальной эксплуатации автотранспорта.

В качестве оборудования по сбору данных было решено взять мобильные устройства (смартфоны, планшеты), так как большинство современных мобильных девайсов содержит много различных датчиков, включая акселерометр, GPS-контроллер, гироскоп, магнитометр и другие. Использование смартфонов в качестве инструментов для сбора данных является перспективной альтернативой из-за своей относительно малой стоимости, простоты в использовании, а также потому что почти у каждого человека есть такое устройство. Еще одним большим плюсом мобильных устройств является то, что они могут быть легко установлены в кабине транспорта для сбора данных, а после поездки сразу же могут использоваться в обычных целях.

Таким образом, в данной работе рассматривается задача применения современных мобильных устройств для мониторинга качества дорожного покрытия. Была поставлена задача рассмотреть алгоритмы оценки качества дорожного покрытия и маркировки участков дорог по качеству, реализовать приложение, позволяющее в режиме реального времени обнаруживать дефекты на поверхности дороги и особым образом помечать такие участки дороги на карте.

Задача оценки качества дорожного покрытия и маркировки участков дорог по качеству заключается в установлении наличия на участке дороги дефектов какого-либо типа (неровностей, ям), обладающих некоторыми определенными характеристиками. И если такие дефекты присутствуют на участке дороги – помечать этот участок дороги на карте. Дополнительные сложности связаны как с наличием большого количества событий на дороге (например, торможение и ускорение транспорта, перестроения и резкие

повороты, езда в горку или с горки, лежащие полицейские, люки, рельсы), так и с тем, что одна и та же неровность проявляет себя каждый раз по-разному при её прохождении.

В качестве результата работы предполагается приложение, обнаруживающее в режиме реального времени дефекты дорожного покрытия и выделяющее их на карте некоторым образом – например, цветом.

2. Постановка задачи

В рамках магистерской диссертации необходимо решить следующие задачи:

1. Изучение справочной информации (статей, демонстрационных проектов) по принципам разработки систем мониторинга качества дорожного покрытия.
2. Изучение технологий и библиотек, необходимых для реализации мобильного приложения.
3. Разработать поэтапный алгоритм решения как композицию стандартных методов, выполнить подбор параметров.
4. Реализация системы с соответствующими функциями: оценка качества дорожного покрытия, маркировка участков дорог по качеству, визуализация результатов.
5. Провести тестирование реализованной системы, провести анализ устойчивости и эффективности реализованной системы.

3. Анализ параметров движения с помощью мобильных устройств

Система мониторинга качества дорожного движения должна работать в режиме «онлайн», что накладывает определенные ограничения. Прежде всего, отметим, что подразумевается работа с двумя потоками данных:

- первый поток – последовательность троек вещественных чисел, получаемая в реальном времени с трехосевого акселерометра, где каждое число из тройки показывает ускорение в соответствующей ему оси;
- второй поток – последовательность наборов данных, в который входит местоположение на карте (широта, долгота, высота), точное время, скорость и угол, на который текущая траектория движения отклоняется от траектории на север, получаемая в реальном времени с GPS-контроллера.

В основной своей массе мобильные устройства, которые могут быть использованы в системе мониторинга качества дорожного покрытия для получения и обработки потоков данных от трехосевого акселерометра и GPS-контроллера, отличаются как довольно посредственными характеристиками трехосевого акселерометра (на рисунке 1 можно увидеть, что случайное блуждание показаний трехосевого акселерометра, то есть отклонение от показания прибора при нулевом кажущемся ускорении, довольно велико) и GPS-контроллера (максимальная точность измерения бытовых GPS-приемников всегда ограничена погрешностью часов и бит-таймом – временем обработки сигнала в приемнике, остальная погрешность набирается при прохождении сигналом атмосферы, то есть зависит от облачности и погоды, от различных препятствий, – лес, здания, тело самого владельца прибора и прочее; на рисунках 2 и 3 видно, что точность GPS-контроллера, применяемого в мобильных устройствах, не высока), так и относительно небольшой вычислительной мощностью. Наиболее распространенными

устройствами, подходящими для решения данной задачи, являются смартфоны, коммуникаторы и планшетные компьютеры, что и будет учитываться в данной работе.

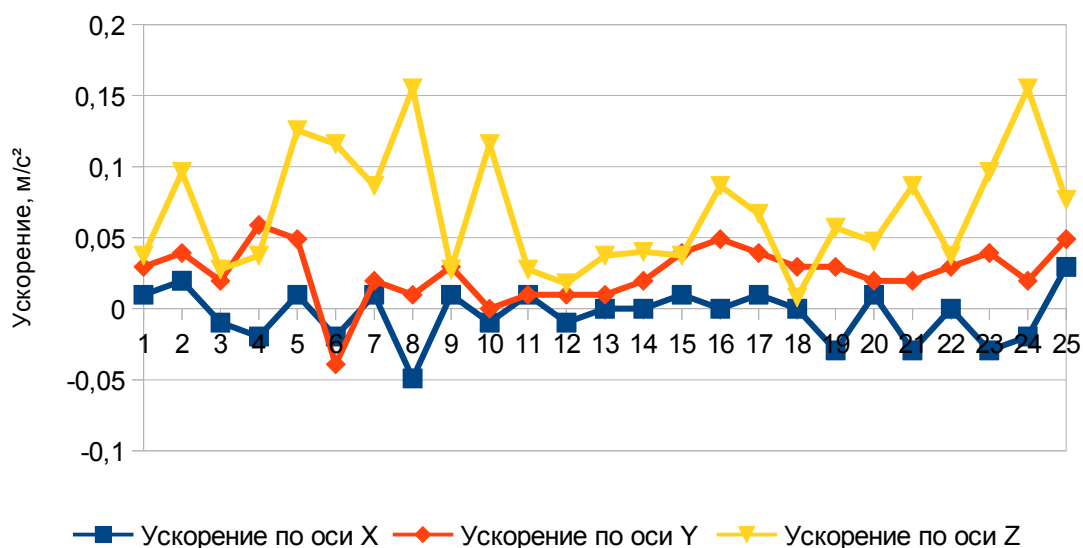


Рисунок 1: Показания трехосевого акселерометра мобильного устройства в неподвижном состоянии

Пример работы GPS-контроллера (см. рис. 2):

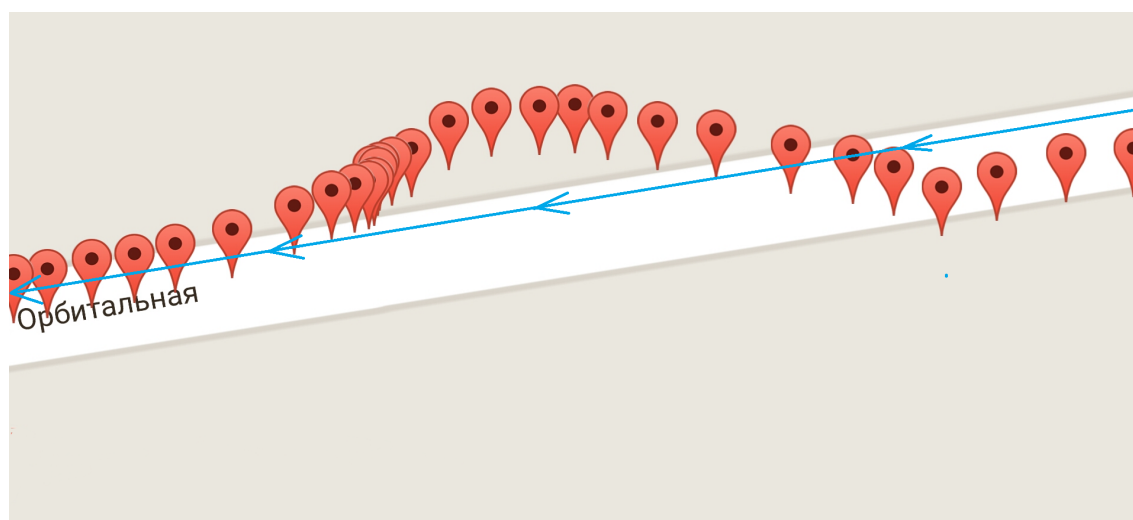


Рисунок 2: Показания GPS-контроллера (красные маркеры) и реальный маршрут автомобиля (синяя линия)

Более сложный пример работы GPS-контроллера (см. рис. 3):

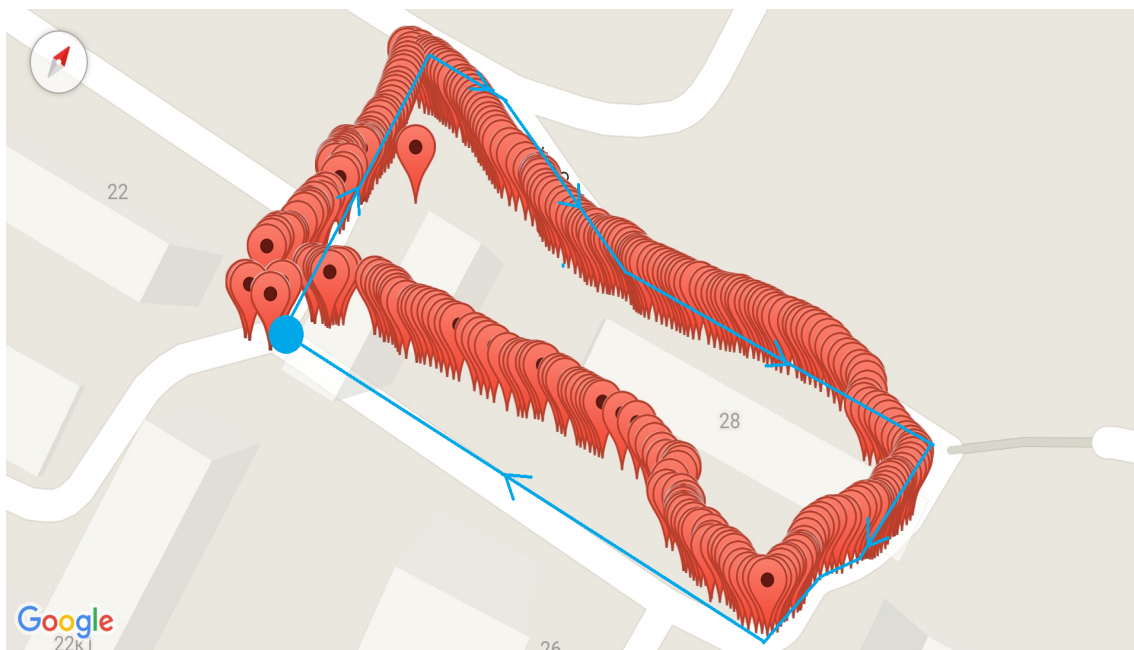


Рисунок 3: Показания GPS-контроллера (красные маркеры) и реальный маршрут автомобиля (синяя линия), где синий круг – начало и завершение маршрута

Итак, предполагается работа с акселерометром мобильного устройства, что налагает следующие ограничения (связанные с типичными характеристиками таких устройств):

- Количество осей: 3
- Пределы измерений: $\pm 2g$, $\pm 4g$, $\pm 8g$ м/с²
- Разрешение: $0.0015g$ м/с²
- Частота выборки: 100-800 Гц
- Погрешность нуля: $0.025g$ м/с²
- Приведенная погрешность нуля: до 3%
- Боковая чувствительность: до 5%

Из общего обзора существующих мобильных акселерометров можно вынести следующие закономерности:

- Рассматриваемые акселерометры имеют довольно существенную начальную погрешность нуля

- Имеется начальная погрешность чувствительности
- Мобильные акселерометры имеют боковую чувствительность. Это значит, что при движении только по оси X, мы также увидим небольшое движение по оси Y

На GPS-контроллер мобильного устройства налагаются следующие ограничения (связанные с типичными характеристиками таких устройств):

- Холодный старт: 30 с
- Теплый старт: 1 с
- Время определения позиций: < 1 с
- Точность определения скорости: 0,1 м/с
- Точность определения времени: 1 мкс
- Точность определения местоположения: 5 м

Данные трехосевого акселерометра и GPS-контроллера получены с помощью мобильного устройства, установленного внутри автомобиля. Изначально рассматриваем базовую задачу – обнаружение дефектов какого-либо типа на одном небольшом участке дороги, предположительно протяженностью от 1 до 10 метров, полученного на основе двух подряд идущих координат с GPS-контроллера. Предполагается, что именно эта задача будет являться наиболее сложной. Обработка последовательности таких участков дорог никакого принципиального отличия не имеет. Дефект дорожного покрытия должен быть установлен довольно быстро (не более секунды). Следующей важной задачей является задача оценки качества этого участка дороги на основе обнаруженных дефектов и маркировка на карте особым образом. Решение этих задач является основой для системы мониторинга качества дорожного покрытия.

4. Системы мониторинга качества дорожного покрытия

На данный момент статьи на тему систем мониторинга качества дорожного покрытия – большая редкость. При этом большинство статей оценивает качество дорожного покрытия с помощью дорогих передвижных лабораторий. В русскоязычном сегменте альтернативные способы оценки состояния дороги представлены только методами оценки с помощью видеоустройств. А вот методов оценки с помощью датчиков типа акселерометр/гироскоп не удалось найти. Поэтому все статьи по данной теме брались из англоязычных источников.

Ознакомиться с основными алгоритмами обнаружения дефектов дорожного покрытия с помощью акселерометра можно в статье [1]. В этой статье описываются следующие алгоритмы:

Z-THRESH – это первый и самый простой алгоритм обнаружения дефектов дорожного покрытия, основанный на показаниях акселерометра по оси Z. Функция, которая классифицирует значения акселерометра по оси Z, является пороговым фильтром по некоторому значению (см. рис. 4). На основе этого порогового значения определяется тип дефекта, например, яма, трещина или чередование на покрытии впадин и возвышений. Алгоритм предполагает, что информация о положении оси Z акселерометра известна (дополнительная виртуальная переориентация акселерометра возможна).



Рисунок 4: Алгоритм обнаружения дефектов дорожного покрытия Z-THRESH

Z-DIFF – это более продвинутый алгоритм обнаружения дефектов дорожного покрытия. Этот алгоритм также основан на показаниях акселерометра по оси Z. Функция-классификатор алгоритма Z-DIFF является пороговым фильтром разницы двух подряд идущих данных акселерометра по оси Z (см. рис. 5). Алгоритм Z-DIFF, в отличие от Z-THRESH, обнаруживает быстрые изменения вертикальных ускорений, по которым можно классифицировать дефекты дорожного покрытия. Аналогично предыдущему алгоритму, предполагается, что информация о положении оси Z акселерометра известна (дополнительная виртуальная переориентация акселерометра возможна).

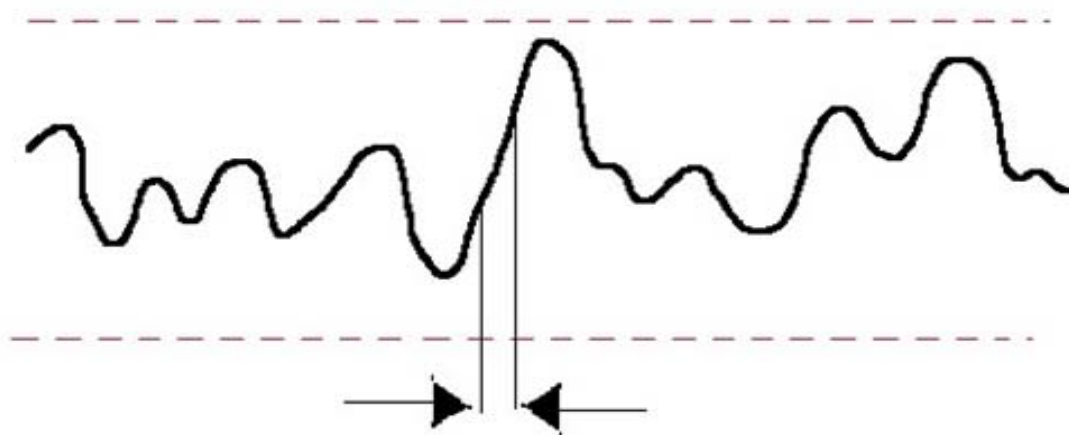


Рисунок 5: Алгоритм обнаружения дефектов дорожного покрытия Z-DIFF

G-ZERO – это алгоритм обнаружения дефектов дорожного покрытия, открытый при использовании средств визуального анализа данных для поиска шаблонов поведения данных с акселерометра. Авторы обнаружили, что существуют определенные события на дороге, характеризующиеся каждой из трех осей акселерометра. При наступлении этих событий данные по каждой оси акселерометра были близки по значению (см. рис. 6). Эмпирический анализ этих наборов данных привел к двум следующим выводам: 1) эти наборы данных получены, когда транспортное средство находилось во временном состоянии свободного падения, например, входа или

выхода из выбоины; 2) эти данные могут быть проанализированы без информации о точном положении осей акселерометра.

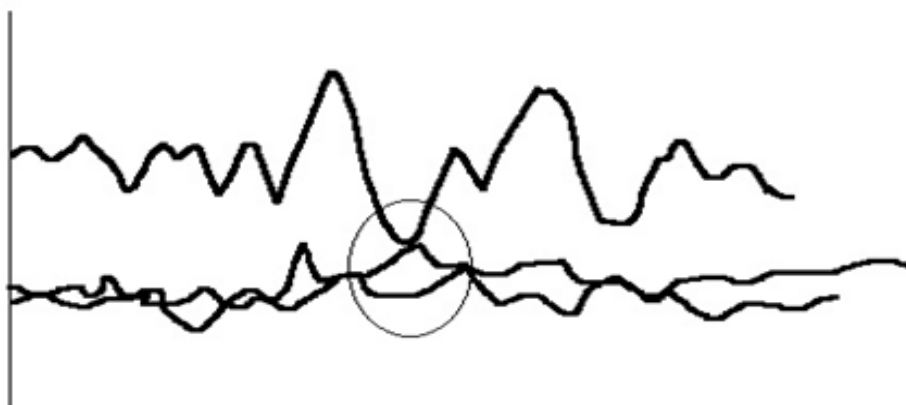


Рисунок 6: Алгоритм обнаружения дефектов дорожного покрытия G-ZERO

STDEV(Z) – это алгоритм обнаружения дефектов дорожного покрытия на основе среднеквадратичного отклонения показаний оси Z акселерометра. Этот алгоритм имеет очень низкий показатель ложный срабатываний, но требует дополнительных исследований.

В статье [2] описывается первая система мониторинга качества дорожного покрытия на основе акселерометра и GPS-контроллера, получившая название P² (Pothole Patrol).

В данной системе мониторинга берутся следующие классы состояния дороги:

- гладкая дорога
- дорожные компенсаторы
- рельсы
- ямы
- люки

Классификатор строился на основе показаний осей X и Z акселерометра, а также скорости автотранспорта так, чтобы максимизировать правильные срабатывания, минимизируя при этом количество ложных срабатываний. Для повышения надежности классификатора в выборку специально добавлялись дороги разных типов, например, дороги с большим количеством люков, хайвеи в идеальном состоянии, дороги в плохом состоянии и другие.

Система P² тестировалась в окрестностях Бостона при поддержке службы такси. На семь машин такси была установлена система мониторинга качества дорожного движения P²: компьютер Soekris 4801 с операционной системой Linux, GPS-контроллер (установлен на крыше автомобиля), трехосевой акселерометр (установлен на всех машинах в одном и том же месте и в одинаковом положении осей акселерометра). В этом эксперименте все автомобили были Toyota Priuses, разных годов производства. Разные водители управляли разными автомобилями в разные дни. Система P² была установлена на автомобилях в течении нескольких недель. В ходе эксперимента автомобили проехали 9730 километров, среди которых 2492 километра являются уникальными. Стоит отметить, что расчетная длина пройденного пути не учитывает полосы, идущие в разных направлениях по одной и той же дороге.

В итоге, классификатор системы мониторинга P² имеет очень низкий процент ложных срабатываний для ям – менее 0,2%.

Еще одна система мониторинга качества дорожного покрытия на основе акселерометра и GPS-контроллера была описана в статье [3]. В отличие от предыдущей системы мониторинга P², система из этой статьи использует акселерометр и GPS-контроллер мобильного устройства. Также стоит отметить, что мобильное устройство по сбору данных устанавливается на велосипеде, а не на автомобиле. Для того, чтобы справиться с огра-

ниченной вычислительной мощностью мобильных устройств, применяется подход машинного обучения: классификатор обучается на обычном компьютере, а затем применяется онлайн на смартфоне.

В данной системе мониторинга берется только два класса состояния дороги:

- гладкая дорога
- ямы

Для классификации было выбрано несколько разных маршрутов, каждый имеет длину от 110 до 130 метров. Каждый маршрут был записан 15 раз. В роли классификатора используется наивный байесовский классификатор. В итоге, точность срабатываний на ямы – 82,857%.

5. Проект системы мониторинга качества дорожного покрытия

Работа системы мониторинга качества дорожного покрытия разбивается на следующие этапы (см. рис. 7):

1. Сбор данных с датчиков мобильного устройства (трехосевой акселерометр и GPS-контроллер)
2. Обработка полученных данных
3. Маркировка соответствующего участка дороги

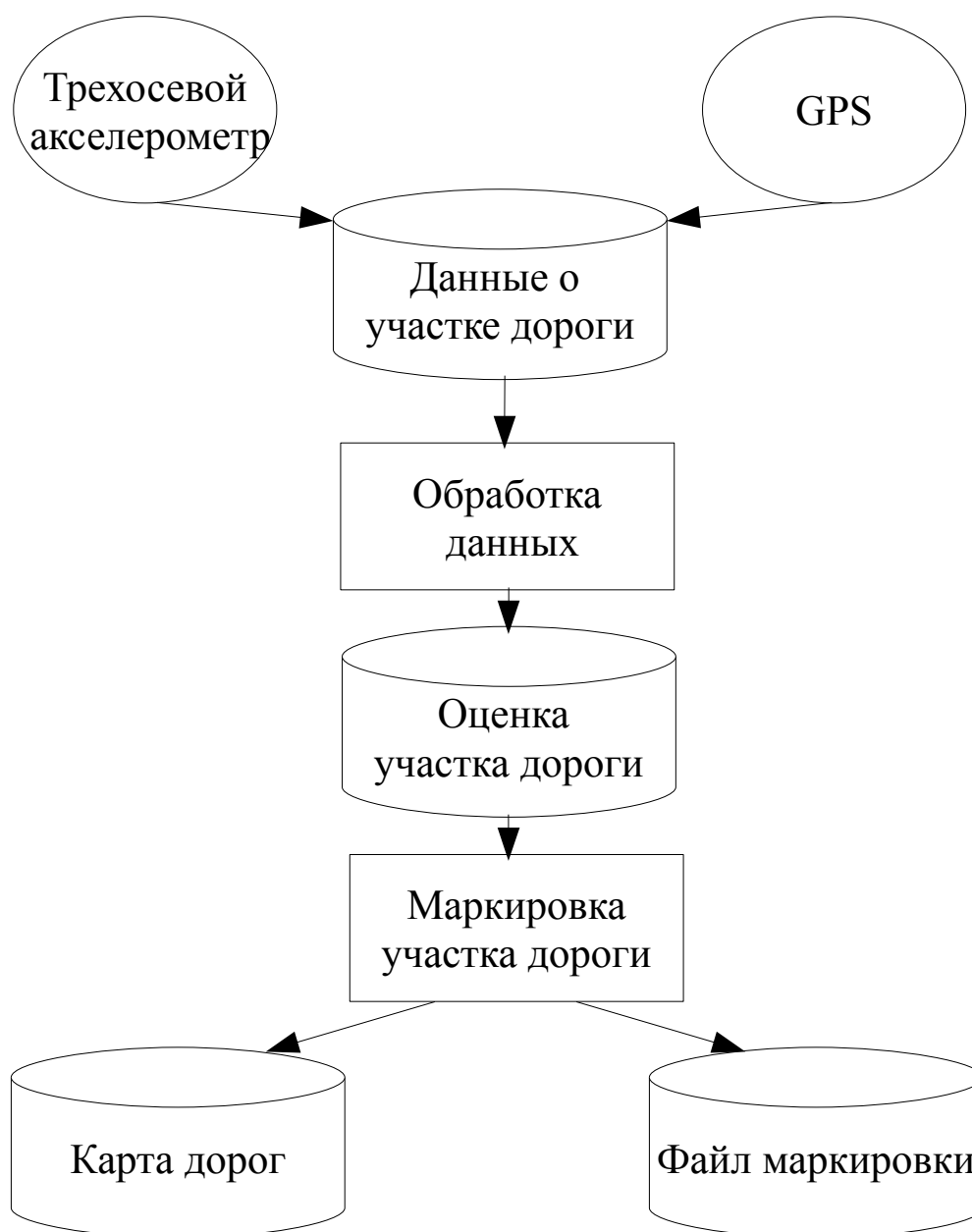


Рисунок 7: Схема работы системы мониторинга

Этап «Сбор данных с датчиков мобильного устройства» разбивается на:

1. Сбор данных с трехосевого акселерометра
2. Сбор данных с GPS-контроллера

На выходе этого этапа получаем поток данных ускорений по каждой оси акселерометра и поток данных местоположения автомобиля, на котором установлено мобильное устройство.

В свою очередь этап «Обработка полученных данных» разбивается на:

1. Применение пороговых фильтров
2. Поиск дефектов дорожного покрытия
3. Оценка качества дорожного покрытия

На вход этапа «Обработка полученных данных» подаются потоки данных, полученные на этапе «Сбор данных с датчиков мобильного устройства». На выходе получаем оценку состояния участка дороги, исследуемый в данный момент

Этап «Маркировка соответствующего участка дороги» разбивается на:

1. Визуализация качества дорожного покрытия
2. Сохранение данных о качестве участка в файл

На основе оценки состояния участка дороги, полученной на предыдущем этапе, на выходе получаем карту качества дорожного покрытия, а также файл, по которым можно восстановить эту карту на другом мобильном устройстве.

Рассмотрим этапы подробнее.

5.1. Сбор данных с датчиков мобильного устройства

Система мониторинга качества дорожного покрытия зависит от точности акселерометра и GPS-контроллера, предоставляющих следующую информацию:

- время
- положение
- скорость
- угол отклонения от севера
- вектор ускорений

В этом разделе будет представлена информация о акселерометре и GPS-контроллере мобильных устройств, будет описано несколько экспериментов, проведенных для проверки функционирования датчиков.

5.1.1. Система координат акселерометра

Трехосевой акселерометр представляет данные в стандартной трехосевой системе координат. Трехосевая система координат датчика выбирается относительно экрана устройства в его обычной (по умолчанию) ориентации. Для телефона ориентация по умолчанию – портретная; для планшетного ПК – альбомная. Когда устройство удерживается в своем обычном положении, ось X направлена по горизонтали и указывает вправо, ось Y направлена вертикально вверх, а ось Z указывает за пределы экрана (навстречу смотрящему). Наиболее важным моментом для системы координат акселерометра является тот факт, что эта система никогда не меняется, когда устройство перемещается или меняется его ориентация. На рисунке 8 показана система координат датчика для телефона, а на рисунке 9 – для планшетного ПК. [4]

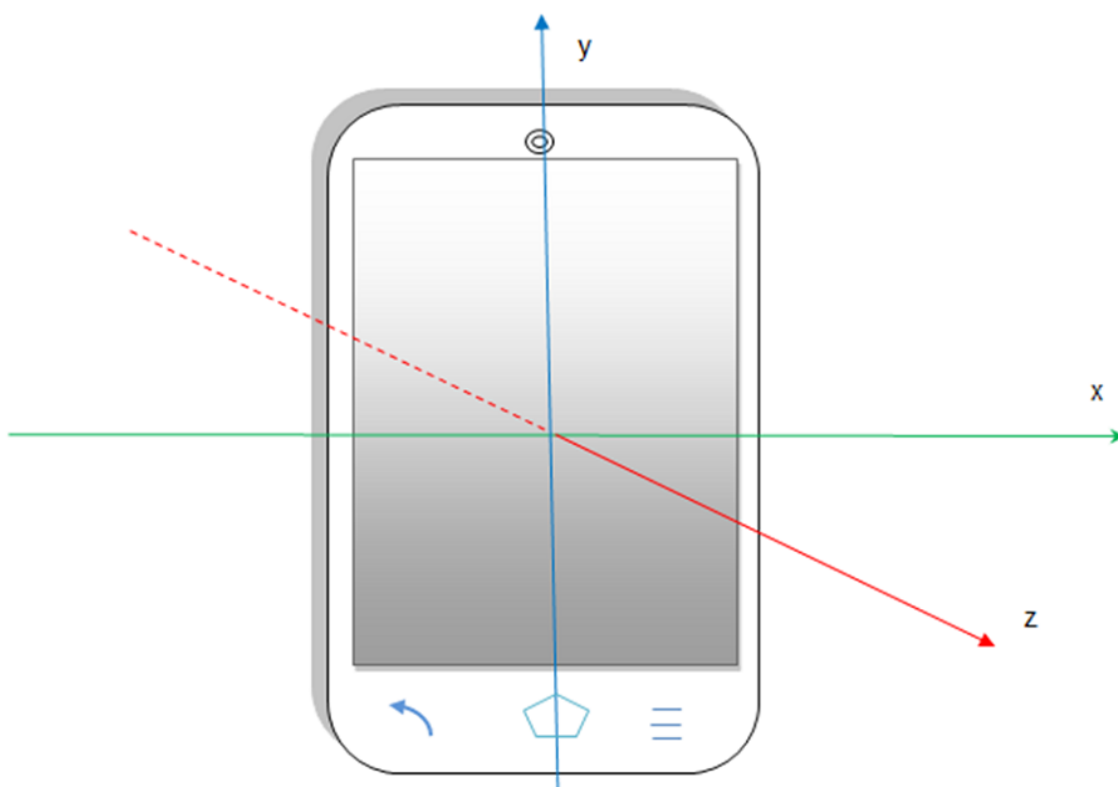


Рисунок 8: Система координат акселерометра для телефона

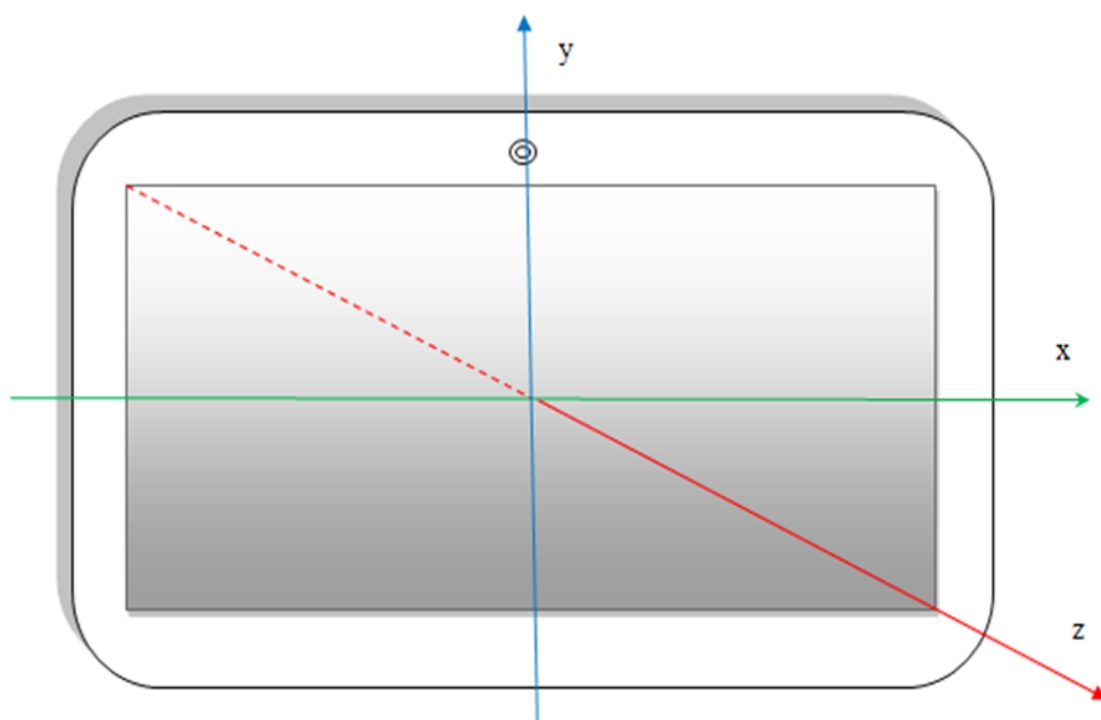


Рисунок 9: Система координат акселерометра для планшетного ПК

5.1.2. Расположение акселерометра

Один из вопросов, возникший при реализации системы мониторинга качества дорожного покрытия, можно описать так: «Влияет ли на качество сигнала трехосевого акселерометра мобильного устройства место его расположения в автомобиле?». Для того чтобы ответить на этот вопрос, было проведено сравнение показаний трехосевого акселерометра для фиксированного отрезка дороги при расположении акселерометра в трех разных местах в кабине одного и того же автомобиля (при одинаковом расположении системы координат датчика):

- на приборной панели (см. рис. 10)
- на лобовом стекле при помощи держателя мобильных устройств (см. рис. 11)
- на панели между сидениями водителя и переднего пассажира (см. рис. 12)

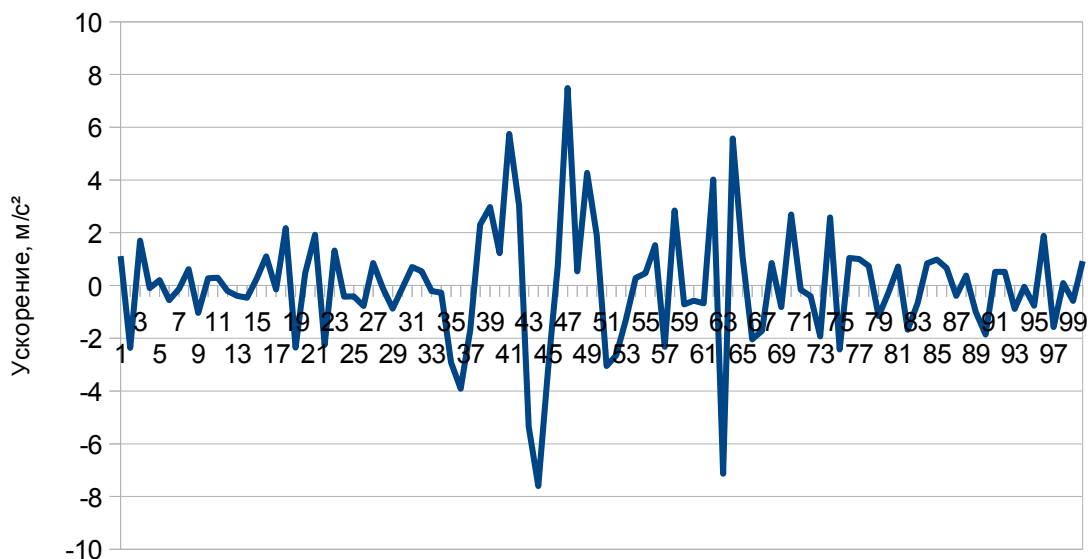


Рисунок 10: Показания акселерометра, расположенного на приборной панели, при прохождении неровности

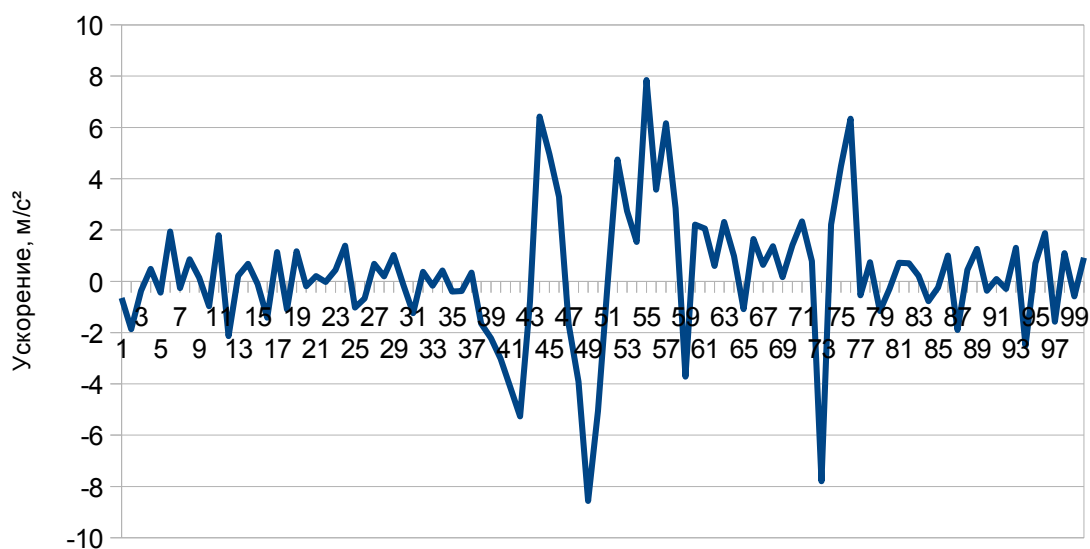


Рисунок 11: Показания акселерометра, расположенного на лобовом стекле, при прохождении неровности

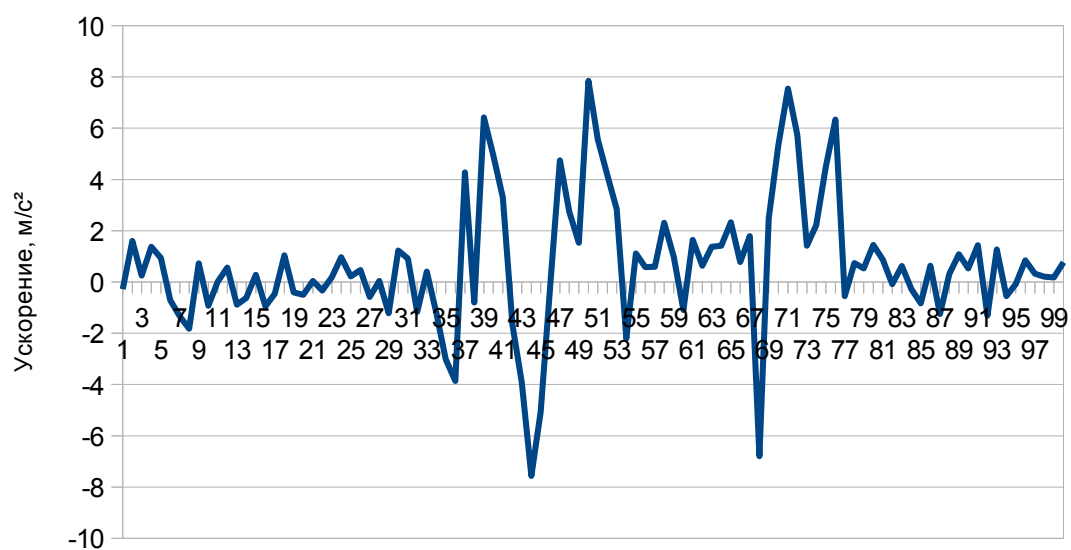


Рисунок 12: Показания акселерометра, расположенного на панели между водителем и передним пассажирским сидениями, при прохождении неровности

Можно сделать вывод, что расположение акселерометра не влияет на сигнал, так как полученные сигналы с акселерометра очень похожи друг на друга.

5.1.3. Точность GPS-контроллера

Точность GPS-контроллера в системе мониторинга качества дорожного покрытия очень важна для того, чтобы правильно локализовать расположение дефектов дорожного покрытия. Для измерения точности GPS-контроллера, было взято 10 показаний датчиков мобильного устройства при прохождении одной и той же искусственной неровности («лежачий полицейский») автомобилем. Искусственная неровность находилась на открытом воздухе, замеры показаний датчиков мобильного устройства брались в ясную погоду. Для каждого такого замера данных с датчиков устройства, определялся пик ускорения по оси Z акселерометра, а затем рассчитывалось местоположение автомобиля, когда этот пик произошел, используя линейную интерполяцию между показаниями GPS-контроллера.

Для расчета точности данных GPS-контроллера была применена формула [5], используемая в навигации для нахождения расстояния между двумя географическими координатами:

$$d = 2 R \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right),$$

где d — это расстояние между двумя географическими точками, через R обозначается радиус Земли, φ_1 и φ_2 — широта географической точки 1 и 2, а λ_1 и λ_2 — долгота географической точки 1 и 2.

В итоге, было подсчитано стандартное отклонение позиции GPS-контроллера, которое равно 3,7 метров, что согласуется с типичными показателями современных GPS-контроллеров на открытом воздухе.

5.2. Обработка полученных данных

В этом разделе будет описан алгоритм, разработанный для обнаружения дорожных аномалий в потоке данных с датчиков акселерометра и GPS-контроллера мобильного устройства.

Алгоритм основан на том, что различные дефекты дорожного покрытия отражаются в особенностях данных ускорения по оси Z. Проблема идентификации аномалий дорожного покрытия является сложной задачей из-за большого количества типов дефекта (яма, трещина, чередование на покрытии впадин и возвышений), а также различных действий водителя (ускорение, торможение, повороты, перестроения). На рисунке 13 показаны основные этапы алгоритма обнаружения дефектов дорожного покрытия.

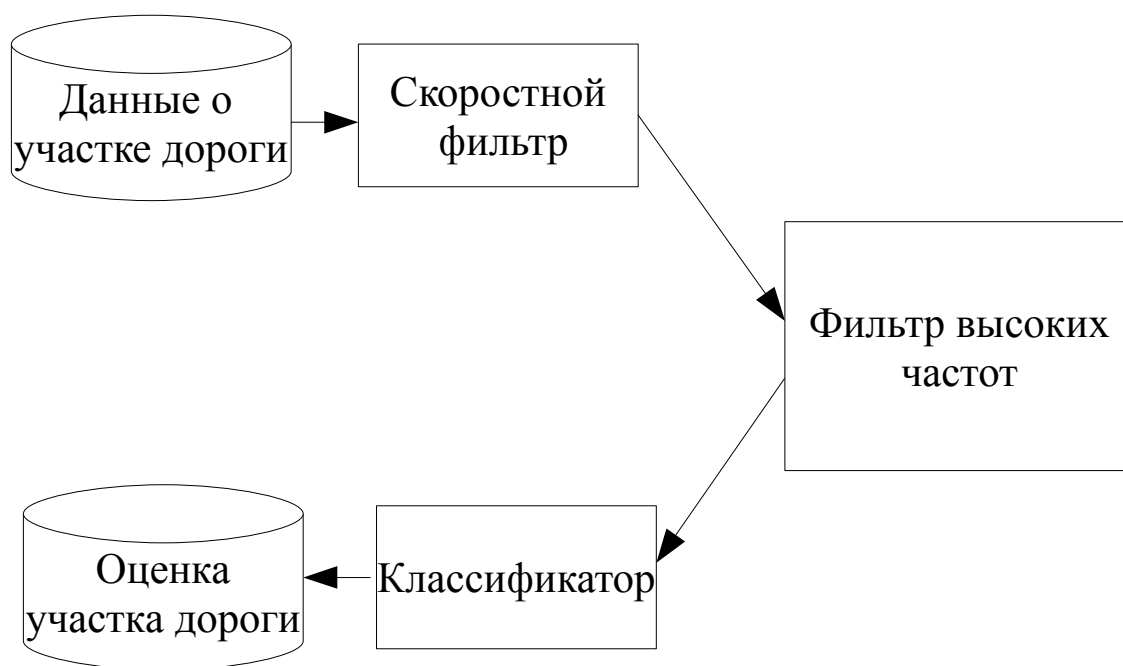


Рисунок 13: Основные этапы алгоритма обнаружения дефектов дорожного покрытия

Рассмотрим каждый этап подробнее.

5.2.1. Скоростной фильтр

Скоростной фильтр нужен для того, чтобы игнорировать данные, когда автомобиль стоит или движется очень медленно. Скоростной фильтр отбрасывает такие события как:

- стояние автомобиля (на перекрестке, перед светофоров, на пешеходном переходе и т.д.)
- наезды на бордюр
- сильные хлопанья дверью
- удары по приборной панели

Было решено брать показания акселерометра, когда скорость движения автомобиля превышает скорость 1,5 м/с (5,4 км/ч). Это также отбрасывает показания акселерометра при ходьбе человека, если забыли выключить программу, когда закончили поездку и забрали мобильное устройство.

5.2.2. Фильтр высоких частот

Фильтр высоких частот удаляет низкочастотные компоненты из сигнала ускорения по оси Z , что позволяет отбрасывать следующие события (хотя почти все эти события главным образом влияют на показания по оси X или Y , но из-за боковой чувствительности показаний трехосевого акселерометра мобильного устройства это дает выбросы ускорений и по оси Z акселерометра):

- ускорение
- торможение
- поворот
- перестроение
- колебания акселерометра

5.2.3. Классификатор

Так как система мониторинга качества дорожного покрытия должна работать в режиме онлайн, а также чтобы справиться с ограниченной вычислительной мощностью мобильных устройств, применяется подход машинного обучения: классификатор обучается оффлайн на персональном компьютере, а затем применяется онлайн на мобильных устройствах.

Пиковое ускорение по оси Z является главной характеристикой дорожных дефектов. Классификатор построен на основе порогового фильтра показаний акселерометра по оси Z . Значение этого порога t вычисляется с помощью перебора по разумным диапазоном значений. Для каждого значения вычисляется его счет по формуле:

$s(t) = \text{corr} - \text{incorr}^2$, где corr — количество дефектов дорожного покрытия помеченных правильно (т. е. как дефекты дорожного покрытия), а incorr — это количество ложных срабатываний. Используется квадрат числа ошибочных обнаружений, чтобы подчеркнуть важность минимизации ложных срабатываний. Окончательное значение порогового фильтра выбирается так, чтобы максимизировать $s(t)$.

Для обучения классификатора было выбрано десять разных маршрутов, каждый имеет длину от 200 до 300 метров. Каждый маршрут был записан десять раз. Для повышения надежности классификатора в выборку специально добавлялись дороги разных типов:

- прямая дорога в идеальном состоянии (трасса)
- маршрут с несколькими поворотами, включая один резкий
- дорога, содержащая трещины
- дорога с чередованием впадин и возвышений («гормошкой»)
- дороги, содержащие большое количество ям
- дороги с нормальным распределением дефектов дорожного покрытия

5.3. Маркировка участка дороги

На основе информации, полученной на предыдущем этапе, участок дороги будет маркироваться одним из следующих состояний:

- гладкая дорога
- жесткая дорога (содержит маленькие трещины или неровности)
- дорога, содержащая крупный дефект дорожного покрытия (яма, глубокие трещины и т. д.)

На карте каждое из этих состояний будет визуализироваться, соответственно, зеленым, желтым и красным цветом.

Также эта информация будет заносится в файл (для возможности восстановления на другом устройстве). Файл имеет следующий формат: координаты точки один — координаты точки два — состояние дороги , где координаты точки один и два — это координаты участка дороги.

6. Выбор операционной системы для мобильных устройств

В самом начале работы возникла проблема выбора операционной системы для мобильных устройств. Операционная система для мобильных устройств выбиралась из следующего списка:

- **Android** — операционная система для смартфонов, планшетов, электронных книг, цифровых проигрывателей, наручных часов, игровых приставок, нетбуков, смартбуков, телевизоров и других устройств. Основана на ядре Linux и собственной реализации виртуальной машины Java от Google. Android позволяет создавать Java-приложения, управляющие устройством через разработанные Google библиотеки. Android Native Development Kit позволяет портировать библиотеки и компоненты приложений, написанные на языке C и других языках. Android доступен для различных аппаратных платформ, таких как ARM, MIPS, x86. [6]
- **iOS** — операционная система для смартфонов, электронных планшетов и носимых проигрывателей, разрабатываемая и выпускаемая американской компанией Apple. Выпускается только для устройств, производимых фирмой Apple. В iOS используется ядро XNU, основанное на микроядре Mach и содержащее программный код, разработанный компанией Apple, а также код из ОС NeXTSTEP и FreeBSD. iOS работает только на планшетных компьютерах и смартфонах с процессорами архитектуры ARM. [7]
- **Windows Phone** — это мобильная операционная система, разработанная Microsoft. Является преемником Windows Mobile, хотя и несовместима с ней. Основана на ядре Windows NT. Windows Phone работает только с процессорами ARM и x86. [8]

- **BlackBerry** — это операционная система, разработанная компанией Research In Motion для семейства смартфонов и планшетов BlackBerry. Эта операционная система базируется на ядре QNX, из которого было взято несколько решений для принудительного управления привилегиями на уровне приложений и файлов. BlackBerry работает только на смартфонах с процессорами архитектуры ARM. [9]

Основным фактором при выборе мобильной платформы для системы мониторинга качества дорожного покрытия является распространенность платформы на рынке, поскольку большее количество пользователей способствует улучшению качества системы мониторинга состояния поверхности дороги. Процентное соотношение продаж смартфонов за 4 квартал 2015 года можно увидеть на рисунке 1. Более подробную динамику продаж смартфонов можно посмотреть на рисунке 2.

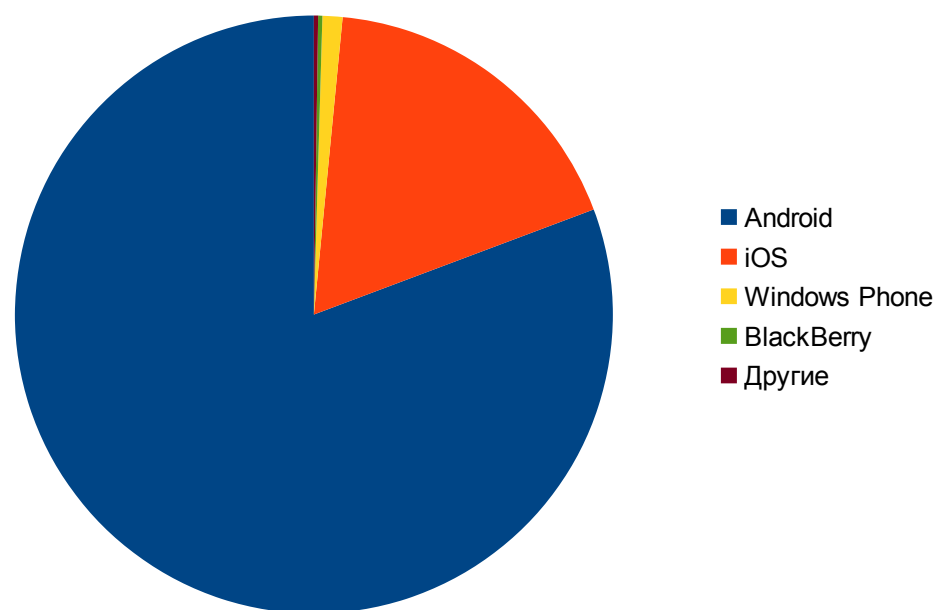


Рисунок 1: Продажи смартфонов в 4 квартале 2015 года по данным Gartner [10]

Мировые продажи смартфонов

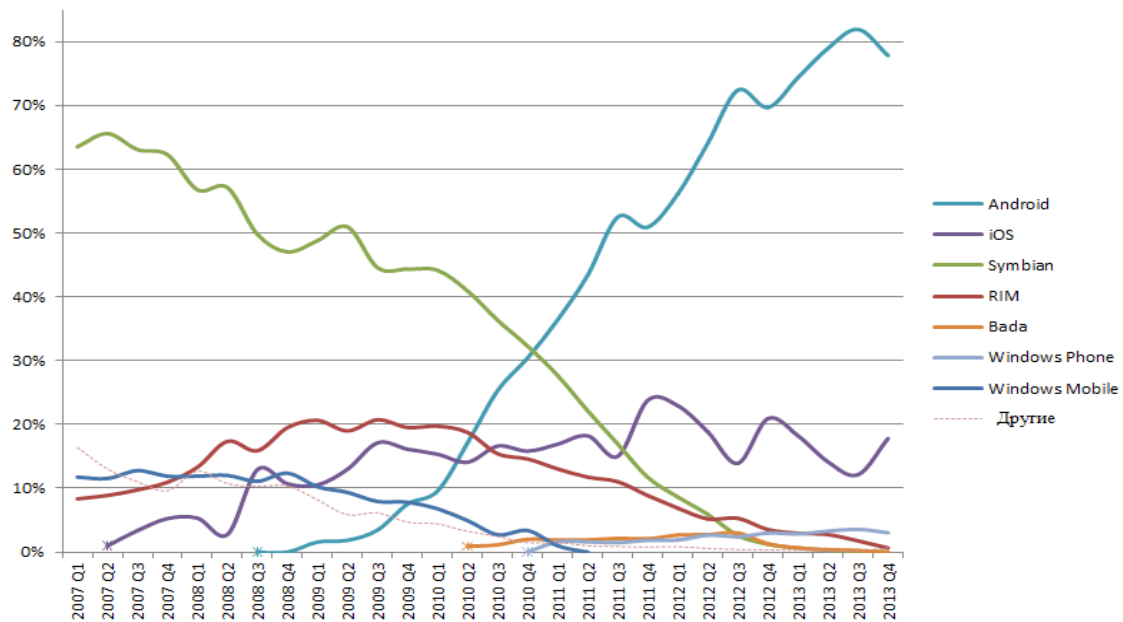


Рисунок 14: Мировые продажи смартфонов с 2007 по 2013 года по данным Gartner

Было решено выбрать операционную систему Android, так как она очевидно является лидером по количеству пользователей. Android превосходит своих конкурентов по масштабности и возможности настроек. Большинство производителей мобильных гаджетов выпускают свои гаджеты на Android во всех ценовых категориях. К тому же на PlayMarket предлагается огромное число бесплатных приложений, что добавляет популярности Android. Еще одним большим плюсом Android является то, что Google предоставляет для свободного скачивания инструментарий для разработки (Software Development Kit). Android SDK очень удобен и легок в применении, поскольку к нему написана подробная документация и есть большое количество примеров мобильных приложений, которые демонстрируют использование встроенных алгоритмов.

7. Выбор картографического сервиса

Следующей проблемой при создании системы мониторинга качества дорожного покрытия стала проблема выбора картографического сервиса. Было решено выбрать один из следующих картографических сервисов:

- **Google Maps** — набор приложений, построенных на основе бесплатного картографического сервиса и технологии, предоставляемых компанией Google. Созданы в 2005 году. [11]
- **Яндекс.Карты** — поисково-информационная картографическая служба Яндекса. Открыт в 2004 году. На службе представлены подробные карты всего мира. Для России используются только собственные карты компании, обновляющиеся ежемесячно. [12]
- **Yahoo! Maps** — бесплатный картографический сервис, предоставляемый компанией Yahoo!. Картографические данные представлены NAVTEQ и Tele Atlas. Высокодетальные космоснимки метрового разрешения представлены Yahoo! Maps компанией GeoEye. Имеются снимки разрешения 1-2 метра практически на всю территорию США, а также крупных городов по всей планете. [13]
- **Maps.me** — бесплатное приложение для мобильных устройств на основе свободной географической карты OpenStreetMap. С ноября 2014 года Maps.me принадлежит Mail.ru Group и развивается в рамках её международного проекта My.com. [14]
- **Bing Maps** — картографический сервис Microsoft, часть портала Bing. [15]

Было решено выбрать картографический сервис Google Maps. Google Maps это бесплатная служба, не содержащая рекламы, хотя такая возможность зарезервирована на будущее по их использованию. В Google Maps доступно 20 уровней масштаба, что в сумме с одним из лучших покрытий в мире дает очень хорошую детализацию по всему миру. Google Maps легок и удобен в применении, к нему написана подробная документация и есть большое количество примеров, которые хорошо демонстрируют способы использования Google Maps. Но все же основной критерий выбора Google Maps — наличие Google Maps Android API, аналога которому нет у остальных картографических сервисов. Google Maps Android API позволяет делать нативные Android приложения, которые являются более быстрыми и менее ресурсоемкими по сравнению с гибридными приложениями.

8. Аспекты реализации

Приложение выполнено на операционной системе Android на языке программирования Java с использованием картографического сервиса Google Maps.

8.1. Фреймворк сенсоров на платформе Android

Фреймворк сенсоров операционной системы Android предоставляет механизмы доступа к датчикам и их данным, за исключением GPS, доступ к которому осуществляется с помощью сервиса определения местоположения платформы Android. Для того чтобы использовать датчики мобильного устройства нужно подключить пакет `android.hardware`. В таблице 1 перечислены классы и интерфейсы сенсоров на платформе Android.

Фреймворк сенсоров представляет данные датчика вместе с объектами `SensorEvent`. Класс может управлять данными конкретного датчика с помощью интерфейса `SensorEventListener` и регистрации `SensorManager` для данного датчика. Фреймворк датчиков информирует класс об изменениях состояния значений датчика с помощью двух следующих методов вызова `SensorEventListener`, используемых классом:

- `void onAccuracyChanged(Sensor sensor, int accuracy)`
- `void onSensorChanged(SensorEvent event)`

`onAccuracyChanged` – вызывается, когда меняется точность данных сенсора и в начале получения данных. Дает объект-сенсор и уровень точности:

- `SENSOR_STATUS_ACCURACY_HIGH` – максимально возможная точность
- `SENSOR_STATUS_ACCURACY_MEDIUM` – средняя точность, калибровка могла бы улучшить результат

- `SENSOR_STATUS_ACCURACY_LOW` – низкая точность, необходима калибровка
- `SENSOR_STATUS_UNRELIABLE` – данные сенсора ненадежны. Либо нужна калибровка, либо невозможно чтение данных.

| Имя | Тип | Описание |
|---------------------|-----------|--|
| SensorManager | Класс | Используется для создания экземпляра сервиса датчика. Предоставляет различные методы доступа к датчикам, возможности регистрации и отмены регистрации приемников событий датчиков и т.д. |
| Sensor | Класс | Используется для создания экземпляра конкретного датчика. |
| SensorEvent | Класс | Используются системой для публикации данных датчика. Сюда относятся исходные значения данных датчика, тип датчика и точность данных. |
| SensorEventListener | Интерфейс | Предоставляет методы вызова для получения оповещений от класса SensorManager после изменения данных или точности показаний датчика. |

Таблица 1: Классы и интерфейсы сенсоров на платформе Android

`onSensorChanged` – получаем данные от сенсора в объекте `SensorEvent`. Датчик акселерометра представляет данные, используя систему координат, где три значения в объекте `SensorEvent` это значения `values[0]`, `values[1]` и `values[2]`, представляющие данные по соответствующим им осям координат X, Y и Z.

Метод `onSensorChanged` часто вызывается системой для представления данных датчика. Сложные вычисления с данными датчиков должны быть вынесены за пределы этого метода.

Операционная система Android не требует наличия определенного датчика на мобильном устройстве, поэтому перед использованием какого-либо сенсора в приложении нужно сначала проверить его действительную доступность.

Если форма, использующая слушатель (приемник) датчика, станет невидимой или диалог будет остановлен, нужно отменить регистрацию приемника датчика. Это может быть сделано с помощью метода `OnPause()` или в методе `OnStop()`. Иначе датчик будет продолжать сбор данных и, в результате, разряжать батарею.

8.2. GPS-контроллер и местоположение

GPS (Global Positioning System) является системой, основанной на получении спутниковых данных, которая выдает точную информацию о географическом положении по всему миру. GPS-контроллер доступен на большинстве телефонах и планшетных ПК с операционной системой Android. GPS-контроллер предоставляет точные данные о местоположении для работающих на мобильном устройстве приложений. На платформе Android устройство GPS не управляется непосредственно фреймворком сенсоров. Вместо этого сервис определения местоположения Android взаимодействует и передает данные GPS-контроллера в приложения через вызовы приемника местоположения.

В таблице 2 перечислены основные классы и интерфейсы, используемые для доступа к сервисам определения местоположения на платформе Android.

| Имя | Тип | Описание |
|------------------|-------------------|---|
| | | Используется для доступа к сервисам определения местоположения. Предоставляет различные методы для запроса |
| LocationManager | Класс | периодических обновлений данных местоположения для приложения или для отправки предупреждений о приближении |
| LocationProvider | Абстрактный класс | Абстрактный супер-класс для провайдеров (поставщиков) определения местоположения |
| Location | Класс | Используется поставщиками данных местоположения для инкапсуляции географических данных |
| LocationListener | Интерфейс | Используется для получения оповещений о местоположении из LocationManager |

Таблица 2: Классы и интерфейсы, используемые для доступа к сервисам определения местоположения в ОС Android

По аналогии с механизмом использования фреймворка сенсоров для получения данных датчиков, приложение использует несколько методов вызовов, определенных в интерфейсе LocationListener для сбора обновлений данных GPS-контроллера о местоположении. LocationManager отправляет в приложение оповещения об обновлениях данных GPS-контроллера с помощью ЭТИХ ВЫЗОВОВ:

- **void** onLocationChanged(**Location** location)
- **void** onProviderDisabled(**String** provider)
- **void** onProviderEnabled(**String** provider)
- **void** onStatusChanged(**String** provider,
int status, **Bundle** extras)

`onLocationChanged` – получаем новые данные о местоположении (а также время, скорость и угол отклонения от севера) от сенсора в объекте `Location`.

`onProviderDisabled` – указанный провайдер был отключен пользователем.

`onProviderEnabled` – указанный провайдер был включен пользователем.

`onStatusChanged` – изменился статус указанного провайдера. В поле `status` может быть одно из следующих значений:

- `OUT_OF_SERVICE` – провайдер пока недоступен, и не ожидается, что будет доступен в ближайшее время
- `TEMPORARILY_UNAVAILABLE` – провайдер пока недоступен, но ожидается, что скоро будет доступен
- `AVAILABLE` – провайдер доступен

Провайдеры включаются и отключаются в настройках системы. Тем самым, просто определяется доступен ли провайдер для получения от него координат. Программное включение/выключение провайдеров через стандартные методы недоступно.

Для получения данных GPS-контроллера о местоположении в приложении нужно запросить разрешение на получение информации о местоположении в файле манифеста `Android`:

```
<manifest ...>
...
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
    android:name="android.permission.ACCESS_COARSE_LOCATION" />
...
</manifest>
```

Сервис GPS предлагает наиболее точную информацию о местоположении устройства. С другой стороны, являясь аппаратным компонентом, он потребляет дополнительную энергию. Поэтому нужно запрашивать обновления данных о местоположении только в случае необходимости, и оперативно отменять регистрацию менеджера местоположений, как только данные больше не нужны. Также требуется время, чтобы сервис GPS мог получить первые данные о местоположении. Когда приложение ожидает данные от сервиса GPS, чтобы получить их точное обновление, для выполнения части работы можно сначала использовать данные о местоположении, предоставляемые методом `getLastKnownLocation()` в классе `LocationManager`.

8.3. Google Maps Android API

С помощью Google Maps Android API в приложение можно добавлять карты на основе данных Google. Этот API-интерфейс автоматически управляет доступом к серверам Google, загрузкой данных, отображением карт и реакцией на жесты, выполняемые на картах. Кроме того, можно использовать вызовы API, чтобы добавлять маркеры, многоугольники и наложения к основной карте, а также изменять способ отображения определенной области на карте. Эти объекты предоставляют дополнительную информацию о местах на карте и обеспечивают возможности взаимодействия пользователей с картой.

Чтобы использовать Google Maps Android API, необходимо зарегистрировать проект приложения в Google Developers Console и получить ключ Google API, который нужно добавить в приложение. Необходимый тип ключа API – ключ для приложений Android. Этот ключ предоставляется бесплатно. Можно использовать его с любым из приложений, обращающихся к Google Maps Android API, без каких-либо ограничений по количеству пользователей.

Основные шаги добавления карты:

1. Получить ключ, включить ключ API в манифест приложения. В файле `AndroidManifest.xml` добавить следующий дочерний элемент к элементу `<application>`, вставив его перед закрывающим тегом `</application>`:

```
<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="API_KEY" />
```
2. Добавить элемент `<fragment>` в файл макета Activity, чтобы определить объект Fragment. В этом элементе установить значение `"com.google.android.gms.maps.MapFragment"` для атрибута `name`. Это автоматически включает MapFragment в Activity.
3. Для работы с картой внутри приложения необходимо будет реализовать интерфейс `OnMapReadyCallback` и использовать метод обратного вызова `onMapReady(GoogleMap)`, чтобы получить дескриптор объекта `GoogleMap`. Обратный вызов выполняется тогда, когда карта готова к использованию. В этом случае она предоставляет экземпляр `GoogleMap`, отличный от `null`. Объект `GoogleMap` является внутренним представлением самой карты. Чтобы установить параметры представления для карты, следует изменить ее объект `GoogleMap`.
4. Вызвать `getMapAsync()` для фрагмента, чтобы зарегистрировать обратный вызов. `getMapAsync()` необходимо вызывать из основного потока, обратный вызов также будет выполнен в основном потоке. Если на устройстве пользователя службы Google Play не установлены, обратный вызов не будет выполнен до тех пор, пока пользователь их не установит.

Google Maps API позволяет указать, какая часть мира должна отображаться на карте. Это достигается с помощью изменения положения камеры (в отличие от перемещения карты). Изменения камеры не приведут к каким-либо изменениям добавленных маркеров, наложений или других графических элементов.

Модель представления карты предполагает направление камеры вниз на плоскость. Положение камеры (и, следовательно, визуализация карты) указывается с помощью следующих свойств: цель (местоположение с широтой/долготой), масштабирование, направление и наклон.

При изменении камеры есть возможность применить анимацию для ее перемещения. Анимация интерполирует изображение между текущими атрибутами камеры и ее новыми атрибутами. Также можно управлять длительностью анимации.

Все программные перемещения камеры рассчитываются относительно размера объекта GoogleMap с учетом всех отступов, добавленных к карте. Например, добавление отступа 100 пикселей к левому краю карты приведет к смещению центра карты вправо на 50 пикселей.

Чтобы изменить положение камеры нужно использовать `CameraUpdate`. Google Maps API позволяет создавать много различных типов `CameraUpdate` с помощью `CameraUpdateFactory`. Метод `CameraUpdateFactory.zoomTo(float)` предоставляет объект `CameraUpdate`, который изменяет уровень масштабирования до заданного значения, сохраняя все остальные свойства без изменений. Существуют два вспомогательных метода для общего изменения положения камеры. Метод `CameraUpdateFactory.newLatLng(LatLng)` предоставляет `CameraUpdate`, который изменяет значение широты и долготы камеры, сохраняя все остальные свойства. Метод `CameraUpdateFactory.newLatLngZoom(LatLng, float)` предоставляет объект `CameraUpdate`, который изменяет значения широты,

долготы и масштабирования камеры, сохраняя все остальные свойства без изменений.

Чтобы применить `CameraUpdate` к камере, можно либо переместить камеру моментально, либо применить к ее перемещению плавную анимацию. Для моментального перемещения камеры можно вызвать `GoogleMap.moveCamera(CameraUpdate)`. Чтобы сделать пользовательский интерфейс более приятным, особенно при небольших перемещениях, изменения можно анимировать. Для этого вместо вызова `GoogleMap.moveCamera` вызывается `GoogleMap.animateCamera`. [16]

9. Тестирование реализованной системы

Главная цель тестирования реализованной системы мониторинга качества дорожного покрытия является определение точности и эффективности, с которой данная система способна обнаруживать некачественные дорожные участки.

К системе мониторинга качества дорожного покрытия выдвигаются следующие требования:

- Система должна иметь очень низкий процент отрицательных ложных срабатываний для гладкой дороги (т. е. гладкий участок дороги система не должна отмечать как участок с дефектами дорожного покрытия). Так как большинство дорог на самом деле являются гладкой дорогой, то даже маркировка небольшого процента гладкой дороги, как дороги с дефектом, приведет к недопустимо большому числу неправильных определений состояния дорожного покрытия
- Ложные срабатывания для дорог с дефектами (т. е. участок дороги с каким-либо дефектом маркируется как гладкий участок дороги) допустимы. Хотя, очевидно, что хотелось бы правильно обнаружить как можно больше, насколько это возможно, участков дороги с дефектами покрытия, но потеря нескольких неисправностей покрытия допустимо.

В итоге была разработана система мониторинга качества дорожного покрытия, которая способна успешно обнаруживать дефекты дорожного покрытия при помощи данных от трехосевого акселерометра и GPS-контроллера, записанных во время обычной езды (на рисунке 15 показан пример работы реализованного приложения). Детектор имеет довольно низкий процент ложных срабатываний для участков дорог с дефектами на тестовом наборе данных (см. таблица 3).

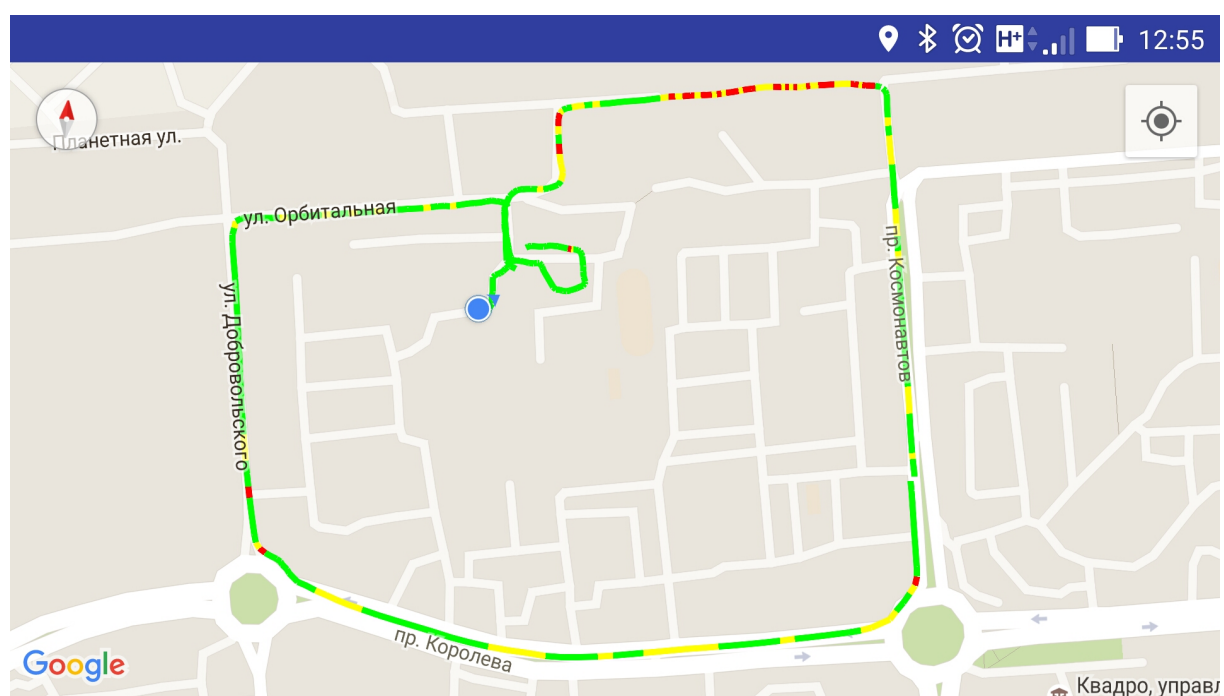


Рисунок 15: Пример работы реализованного приложения

| | Гладкая дорога (правда) | Дорога с дефек- том (правда) | Точность пред- сказания для класса |
|---|----------------------------|---------------------------------|--|
| Гладкая дорога (предсказано) | 526 | 5 | 99,06% |
| Дорога с дефек- том (предсказа- но) | 7 | 38 | 84,44% |
| Точность класса | 98,69% | 88,37% | Общая точность: 97,91% |

Таблица 3: Точность системы мониторинга качества дорожного покрытия

10. Заключение

В данной работе была рассмотрена задача мониторинга качества дорожного покрытия с помощью мобильных устройств.

Были рассмотрены похожие системы мониторинга качества дорожного покрытия, а также были изучены различные алгоритмы обнаружения дефектов дорожного покрытия на основе показаний акселерометра. Итогом стала система мониторинга качества дорожного покрытия с помощью трехосевого акселерометра и GPS-контроллера мобильных устройств на платформе Android. Проект был реализован на языке программирования Java с использованием картографического сервиса Google Maps.

В рамках проекта был реализован алгоритм обнаружения дефектов дорожного покрытия.

11. Список литературы

1. B. Lanjewar, Jyoti Khedkar, Rahul Sagar, Rasika Pawar, Kunal Gosavi. Survey of Road Bump and Intensity Detection algorithms using Smartphone Sensors. IJCSIT, Vol. 6, 2015. – URL: <http://www.ijcsit.com/docs/Volume6/vol6issue06/ijcsit2015060659.pdf> (дата обр. 11.05.2016)
2. Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, Hari Balakrishnan. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. MobiSys, 2008. – URL: <https://www.cs.uic.edu/~jakob/papers/p2-mobisys08.pdf> (дата обр. 11.05.2016)
3. Marius Hoffmann, Michael Mock, Michael May. Road-quality classification and bump detection with bicycle-mounted smartphones. – URL: <http://ceur-ws.org/Vol-1088/paper7.pdf> (дата обр. 11.05.2016)
4. Miao Wei. Software Developer's Guide for Sensors on Intel® Atom™ based Android* Tablets. 2012. – URL: <https://software.intel.com/en-us/android/articles/software-developers-guide-for-sensors-on-intel-atom-based-android-tablets> (дата обр. 11.05.2016)
5. Применение формулы гаверсина в навигации. – URL: https://en.wikipedia.org/wiki/Haversine_formula (дата обр. 11.05.2016)
6. Описание ОС Android. – URL: <https://ru.wikipedia.org/wiki/Android> (дата обр. 11.05.2016)
7. Описание ОС iOS. – URL: <https://ru.wikipedia.org/wiki/iOS> (дата обр. 11.05.2016)
8. Описание ОС Windows Phone. – URL: https://ru.wikipedia.org/wiki/Windows_Phone (дата обр. 11.05.2016)

9. Описание ОС BlackBerry. – URL:
https://ru.wikipedia.org/wiki/BlackBerry_OS (дата обр. 11.05.2016)
10. Статистика использования операционных систем на мобильных де-
вайсах. – URL: <http://www.gartner.com/newsroom/id/3215217> (дата
обр. 11.05.2016)
11. Описание картографического сервиса Google Maps. – URL:
https://en.wikipedia.org/wiki/Google_Maps (дата обр. 11.05.2016)
12. Описание картографического сервиса Яндекс Карты. – URL:
<https://ru.wikipedia.org/wiki/Яндекс.Карты> (дата обр. 11.05.2016)
13. Описание картографического сервиса Yahoo! Maps. – URL:
https://ru.wikipedia.org/wiki/Yahoo!_Maps (дата обр. 11.05.2016)
14. Описание картографического сервиса Maps.me. – URL:
<https://ru.wikipedia.org/wiki/Maps.me> (дата обр. 11.05.2016)
15. Описание картографического сервиса Bing Maps. – URL:
https://ru.wikipedia.org/wiki/Bing_Maps (дата обр. 11.05.2016)
16. Начало работы с Google Maps. – URL:
<https://developers.google.com/maps/documentation/android-api/start>
(дата обр. 11.05.2016)