

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Математико-механический факультет

Кафедра системного программирования

Фоменко Екатерина Сергеевна

Уточнение картографической информации с
использованием классификации GPS треков по типу
транспортного средства

Дипломная работа

Допущена к защите.

Зав. кафедрой:

д. ф.-м. н., профессор Терехов А.Н.

Научный руководитель:

к.ф.-м.н. Барашев Д.В.

Рецензент:

ст.преп. Луцив Д.В.

Санкт-Петербург

2013

SAINT PETERSBURG STATE UNIVERSITY

Mathematics & Mechanics Faculty

Software Engineering Chair

Ekaterina Fomenko

Cartographic information refinement using classification
of GPS tracks by transportation mode type

Graduation Thesis

Admitted for defence.

Head of the chair:

Professor A.N. Terekhov

Scientific supervisor:

Ph. D. D.V. Barashev

Reviewer:

Senior Lect. D.V. Luciv

Saint-Petersburg

2013

Оглавление

Введение	5
1. Постановка задачи	7
2. Обзор предметной области	8
2.1. Выбор картографической системы	8
2.2. Альтернативные редакторы	8
2.3. Классификация треков по типу транспортного средства	9
2.4. Наложение треков на карту	9
3. Реализация	11
3.1. Интеграция с OSM	11
3.1.1. Структура OpenStreetMap	11
3.1.2. Базовые типы географических данных в OpenStreet- Map	13
3.1.3. OSM RESTful API	14
3.1.4. Получение данных OSM	15
3.1.5. Изменение данных OSM	16
3.1.6. Фасад над OpenStreetMap	18
3.2. Анализ GPS треков	19
3.2.1. Форматы треков	19
3.2.2. Координаты точек треков	20
3.2.3. Вычисление расстояния между точками	20
3.2.4. Вычисление расстояния от точки до ломаной	21
3.2.5. Вычисление азимута	21
3.3. Классификация треков по типу транспортного средства	21
3.3.1. Сегментация треков на пешие и не пешие сегменты	22
3.3.2. Вычисление характеристик сегментов	23
3.3.3. Обучение классификатора	25
3.4. Уточнение OpenStreetMap с использованием информации о средстве передвижения	27
3.4.1. Общий подход уточнения карты	28

3.4.2.	Сопоставление треков и дорожного графа	28
3.4.3.	Уточнение тега 'maxspeed:practical'	29
3.4.4.	Уточнение тега 'smoothness'	29
4.	Тестирование системы	31
4.1.	Тестирование анализа GPS треков	31
4.2.	Тестирование классификации GPS треков по типу транспортного средства	31
4.2.1.	Данные для тестирования	32
4.2.2.	Подход тестирования	32
4.2.3.	Результаты тестирования	33
4.3.	Эксперименты	34
4.3.1.	Эксперименты по вычислению типа транспортного средства	34
4.3.2.	Эксперименты по уточнению тегов дорог в OpenStreetMap	34
5.	Дальнейшие исследования	35
	Заключение	36
6.1.	Алгоритм классификации треков по типу транспортного средства	36
6.2.	Подсистема интеграции с OpenStreetMap	37
6.3.	Подсистема автоматического уточнения OpenStreetMap .	38

Введение

На протяжении нескольких последних лет происходит стремительное развитие веб-картографии. Онлайн-карты становятся незаменимыми помощниками при планировании путешествий, поиске наилучших маршрутов. Одной из популярных и весьма актуальных онлайн-карт стал проект OpenStreetMap (OSM) [1], являющийся одним из лучших примеров использования свободного программного обеспечения и краудсорсинга ¹.

Для создания карты используются данные с персональных GPS треков, спутниковые снимки, панорамы. Работа с OSM осуществляется по принципу всемирно известного сайта Wikipedia²: карту может редактировать любой пользователь интернета. Данные проекта распространяются на условиях свободной лицензии. Стив Кост, создатель проекта OSM, был вдохновлён успехом Википедии и решил, что принцип вики, заложенный в основу Википедии, может применяться и для веб-картографии. Расчет оправдался, и OSM в настоящее время пользуется большой популярностью. В некоторых регионах ее точность превосходит аналоги на 20-30% [5]. Это возможно благодаря активности волонтеров: когда их количество достигает 10—13 человек на кв. км, геометрическая точность карты становится достаточно высокой; а также тому, что OSM в первую очередь опирается на точные координаты GPS треков, а не на спутниковые снимки ³.

Проект OSM охватывает всю поверхность земного шара. Сердцем проекта является не сама карта, а база данных, содержащая сведения о точках на земной поверхности. Эти точки могут быть получены из GPS треков, записанных GPS-приемниками. Такие треки создаются добровольцами в процессе путешествий по исследуемому району пешком, на велосипеде или на автомобиле. Затем треки экспортируются из GPS-устройства в специальную программу.

¹Краудсорсинг - передача некоторых производственных функций неопределённому кругу лиц, решение общественно значимых задач силами добровольцев

²<http://www.wikipedia.org/>

³Информация была заимствована отсюда: <http://ru.wikipedia.org/wiki/OpenStreetMap>

Немаловажную роль здесь играет качество дорог и возможные достопримечательности: места остановок, отдыха вдоль этих дорог. Человек, планируя свой поход, сможет тщательно исследовать местность, оценить время прохождения данного пути, исходя из качества дороги, выбрать места остановок, средство передвижения. На данный момент треки, загруженные в OSM, являются лишь набором точек, полученных с помощью GPS, и не несут никакой дополнительной информации. При помощи анализа треков и применения различных алгоритмов, можно получить такую информацию, как способ перемещения вдоль пути, записанного на трек, качество дороги, места остановок и пр.

Дополнение дорог информацией, однако, осуществляется вручную, что занимает много времени и усилий.

В этой дипломной работе предложена расширяемая система по вычислению транспортного средства, а также качества дороги, и автоматическому дополнению карты OSM.

1. Постановка задачи

Основным назначением карт является помощь пешеходам в ориентировании на тропинках и водителям на дорогах и шоссе. Если будет известно средство передвижения, скорость передвижения, а также сведения о траектории дороги, то можно будет сделать некоторые выводы о качестве дороги на определенных ее участках, тем самым помогая решить проблемы с выбором транспортного средства для данного похода. Также, по трекам можно отслеживать “перемещения на месте”, то есть когда основной путь ненадолго прекращается и человек явно разглядывает красивую местность, или же зашел перекусить и т.д. Таким образом, можно максимально точно дополнять дороги на карте, делая тем самым ее использование наиболее удобным и практичным.

Целью данной работы является создание расширяемой системы, способной добавлять новую информацию о дорогах в OSM на основе анализа GPS треков и используемого транспортного средства.

Для достижения цели были поставлены следующие задачи:

1. Реализовать алгоритм классификации треков по типу транспортного средства.
2. Реализовать платформу, предоставляющую API для модификации OSM.
3. Используя полученные результаты, реализовать автоматическое уточнение качества дорог.

2. Обзор предметной области

2.1. Выбор картографической системы

В настоящее время существует множество онлайн-карт. Для выполнения поставленных задач преимущество отдавалось картам на основе краудсорсинга. Рассматривались Викимания ⁴, Google Map Maker ⁵, Яндекс.Народная карта ⁶, OpenStreetMap [1].

Карты Google Map Maker и Яндекс.Народная карта являются проприетарными, а, следовательно, для свободного использования данных и внесения изменений в карты необходимо согласие администраций соответствующих компаний. Этот вариант непригоден для свободного тестирования написанной системы, а также использования данных, загруженных другими пользователями.

Викимания является некоторой надстройкой над различными картами (в том числе и OSM), она лишь отмечает и дополняет информацией места на космоснимках и прочих привязанных подложках.

OpenStreetMap, в свою очередь, дополняет карту объектами, которые не видны на космоснимках, на картах OSM отмечен рельеф, по ним можно навигироваться, составлять маршруты для путешествий, у них определена структура объектов (у дороги, например, могут быть такие свойства как тип и качество).

Таким образом, в качестве основы картографических данных в этой дипломной работе была выбрана OpenStreetMap.

2.2. Альтернативные редакторы

На данный момент для модификации OSM можно использовать несколько редакторов, таких как онлайн-редактор Potlatch, Merkaartor, Mapzen, JOSM, Vespucci. Все они являются полнофункциональными редакторами, но модификация карты в них осуществляется посредством

⁴<http://wikimania.org>

⁵<http://www.google.com/mapmaker>

⁶<http://n.maps.yandex.ru>

пользователя. Т.е. пользователь вручную загружает треки, дорисовывает дороги, дополняет карту новыми данными. В дипломе предложена автоматическая работа с картой, а именно: автоматическая загрузка треков, обработка, классификация и нанесение на карту изменений.

2.3. Классификация треков по типу транспортного средства

В основе большинства работ по вычислению способа перемещения лежит исследование [2] по классификации треков по типу транспортного средства, предложенное в 2008 году, и улучшенное этими же авторами в 2010 году [8]. В данной дипломной работе также используется это исследование.

Последующие работы на эту тематику брали за основу вышеупомянутую работу и модифицировали ее необходимым образом. В частности, в Чикаго в 2011 году [6] были проведены исследования, которые улучшили результаты работы по классификации треков [2] благодаря использованию дополнительной информации:

1. о местонахождении автобусов в конкретный момент времени, свободно распространяемом правительством Чикаго;
2. о географическом расположении железных дорог.

2.4. Наложение треков на карту

Фундаментом для многих работ по наложению треков на карту является алгоритм Map-matching [3]. Он является достаточно эффективным, что делает его популярным в данной области.

Если же работы по классификации треков по типу транспортного средства (см. 2.3) являются "свежими" (2011-2012гг.), то первые активные исследования по наложению треков на карту начали проводиться в 2004 году, как раз во время появления онлайн-карт.

Исследования в этой области проводились с акцентом на дополнение карт новыми дорогами, привязку пути к дорожному графу, или же

на геометрическое изменение дорог [4] , но целью данной дипломной работы является именно уточнение типа дороги.

3. Реализация

Работа, выполненная в рамках данного диплома, представлена в виде системы, содержащей несколько слабосвязанных компонент. В этой главе пойдет речь об их реализации и связях между ними.

На рисунке 1 показана высокоуровневая архитектура выполненных компонент:

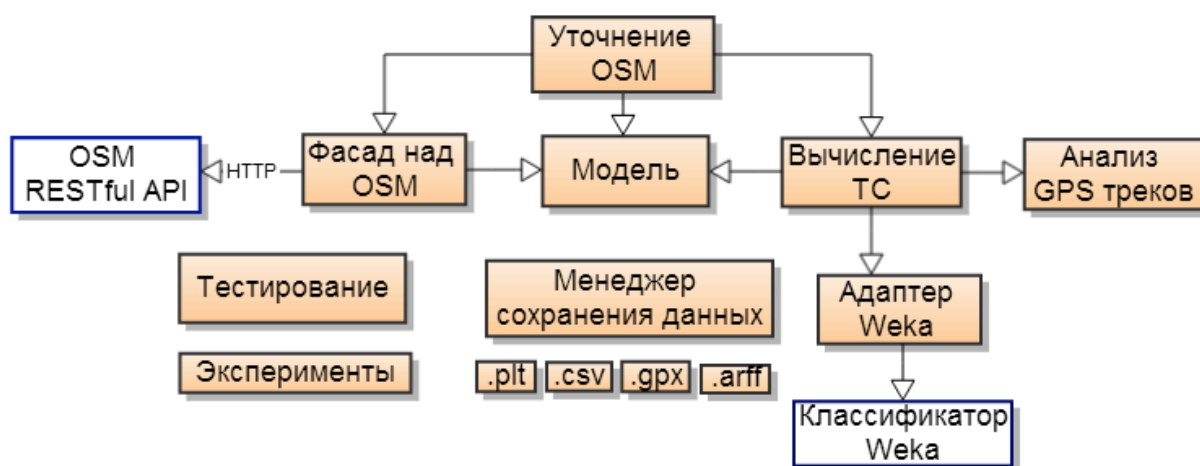


Рис. 1. Диаграмма компонент

На данной диаграмме закрашенные блоки представляют компоненты, которые были реализованы в рамках данной работы, белым цветом показаны используемые готовые продукты.

3.1. Интеграция с OSM

3.1.1. Структура OpenStreetMap

Прежде, чем перейти к разбору компонент, стоит описать модель данных OpenStreetMap.

Сайт [1] состоит из собственно карты, онлайн инструментов по её редактированию (Potlatch 1 и Potlatch 2), дневников участников, форума, справочной вики и раздела помощи. Также существуют и сторонние редакторы карты OSM (см. 2.2).

Структуру данных OSM можно представить схематично на следующем рисунке (рис. 2):

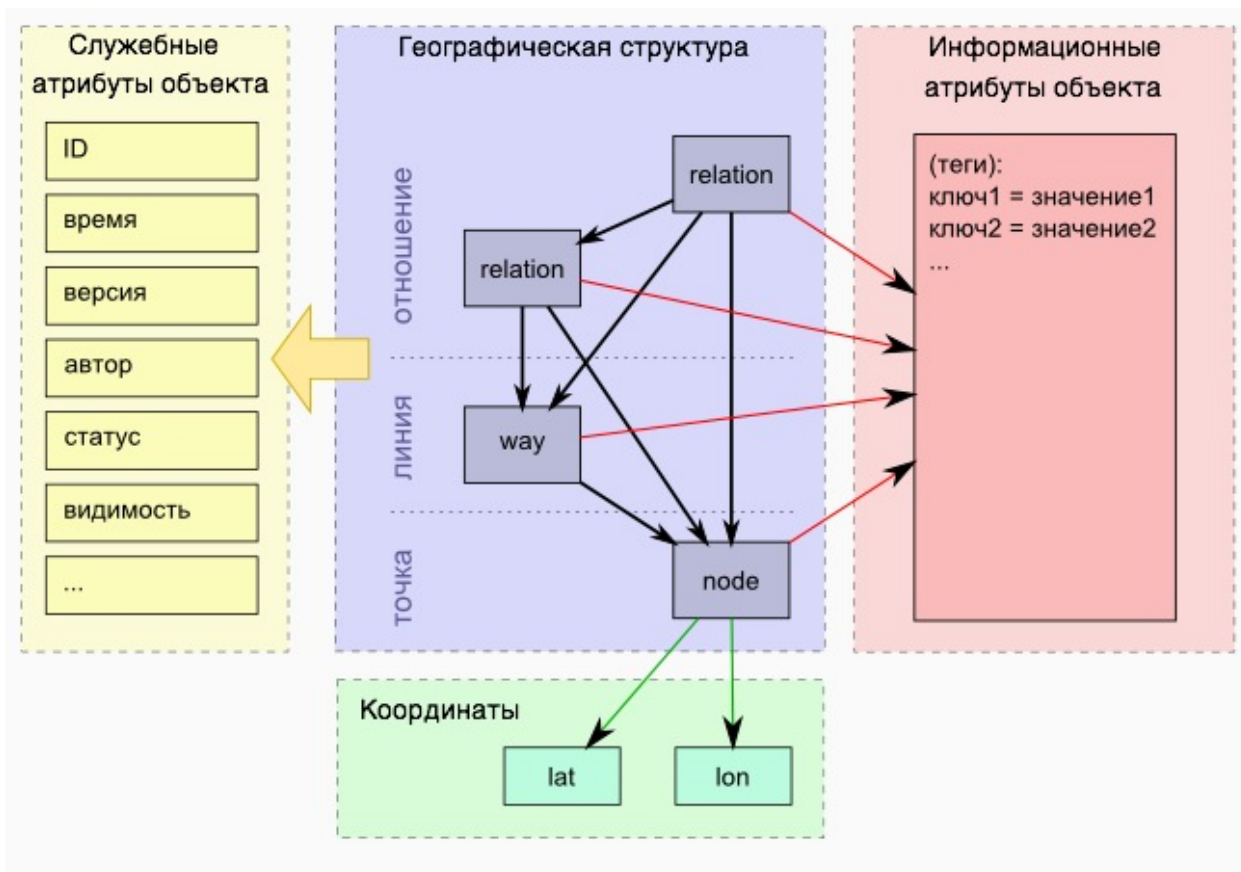


Рис. 2. Структура данных OpenStreetMap

Все данные можно условно разбить на три основные группы:

1. Географическая структура: описывает объект в виде иерархической связи как некоторую пространственную сущность с географическими координатами.
2. Информационные атрибуты объекта: описательная характеристика объекта, не имеющая к пространственной географической структуре объекта прямого отношения (его название, физические, логические и прочие свойства).
3. Служебные атрибуты объекта: необходимы для организации процесса хранения и обработки информации в виде набора данных (такие как уникальный идентификатор, состояние объекта в базе, время последней правки объекта в базе и т.д.).

3.1.2. Базовые типы географических данных в OpenStreetMap

К базовым типам в OSM относят:

1. node (точка) — это минимальный набор данных, который содержит в себе информацию о паре координат: широта, долгота (lat, lon) и является базовым в иерархической модели. Это единственный тип данных, который хранит координаты в виде широты и долготы. В XML нотации, объект данного типа будет выглядеть так:

```
<node id='19' lat='58.888047127548994' lon='49.747870758186764' />
```

2. way (линия) — упорядоченный список точек, составляющих линию или полигон (правильная линия содержит от двух точек и более) XML-нотация объекта типа линия будет заключаться в описании всех необходимых точек, после чего следует сама запись о линии, в которой перечисляются все её точки. В простейшем варианте это будет выглядеть так:

```
<node id='23' lat='58.875047918145675' lon='49.785240674006126' />
<node id='22' lat='58.86687448573524' lon='49.737090974777324' />

<way id='24'>
  <nd ref='22' />
  <nd ref='23' />
</way>
```

Порядок перечисления точек в линии важен, он характеризует последовательность точек и направление линии. У линии всегда есть начало и конец, даже если она замкнутая (в этом случае они просто совпадают).

3. `relation` (отношение) — группы точек, линий и других отношений, которым назначаются некоторые свойства.

```
<relation id='31'>
  <member type='way' ref='24' />
  <member type='node' ref='19' />
</relation>
```

4. `tag` (тег) — пары «ключ-значение», которые могут присваиваться точкам, линиям и отношениям. Теги относятся к информационной части структуры данных OSM и в основном заполняются пользователями, владеющими информацией об объекте.

```
<way id='24'>
  <nd ref='22' />
  <nd ref='23' />
  <tag k='highway' v='primary' />
</way>
```

В вышеприведенном примере добавляется тег `highway` со значением `primary`, означающий, что данная дорога является основной.

Зачастую считают, и мы будем считать так же, что базовых элементов три: точка, линия, отношение, а тег является дополнительным типом.

3.1.3. OSM RESTful API

OSM предоставляет веб-интерфейс ⁷, позволяющий как получать данные, так и изменять их. В данной главе будут подробнее рассмотрены его возможности.

Запросы на сервер логически можно разделить на две группы:

1. получение данных OSM;
2. изменение данных OSM;

Рассмотрим каждую из этих групп подробно.

⁷В настоящее время API доступен по следующей ссылке: <http://api.openstreetmap.org/>

3.1.4. Получение данных OSM

1. Извлечение картографических данных, ограниченных прямоугольником Пример вызова:

```
GET /api/0.6/map?bbox=left ,bottom ,right ,top
```

где:

- left - долгота левого края прямоугольника (самая западная);
- bottom - широта нижнего края прямоугольника (самого южного);
- right - долгота правого края прямоугольника (самая восточная);
- top - широта верхнего края прямоугольника (самая северная).

Эта команда возвращает:

- все узлы, находящиеся внутри заданного прямоугольника;
- все пути, которые ссылаются на хотя бы один узел внутри заданного прямоугольника и все узлы снаружи ограничительного прямоугольника, которые входят в эти пути;
- Все отношения, которые ссылаются на каждый из узлов или путей, включённых по вышеуказанным правилам (но не рекурсивно).

2. Получение элемента OSM по его id

Пример вызова:

```
GET /api/0.6/[node|way|relation]/#id
```

Возвращает представление элемента в виде XML:

```
<osm>  
<node id="123" lat="..." lon="..." version="142"  
  changeset="12" user="fred" uid="123" visible="true"  
  timestamp="2005-07-30T14:27:12+01:00">
```

```
<tag k="note" v="Just a node"/>
...
</node>
</osm>
```

3. Получение точек GPS треков

Следующая команда возвращает точки GPS треков в формате GPX, находящиеся внутри заданного прямоугольника.

Пример вызова:

```
GET /api/0.6/trackpoints?bbox=left , bottom , right ,
top&page=pageNumber
```

где:

- left, bottom, right, top - аналогично пункту 1;
- pageNumber - определяет, какие группы точек по счету возвращаются, так как ответ на запрос не может содержать более 5000 точек.

3.1.5. Изменение данных OSM

Для облегчения идентификации вносимых изменений существует концепция пакетов правок (changesets). Каждая модификация одного или более элементов должна ссылаться на открытый пакет правок, который подобно другим элементам может содержать теги. Новый пакет правок может быть открыт в любое время, и на него могут ссылаться несколько вызовов API.

1. Создание новых объектов на карте OSM.

Пример вызова:

```
PUT /api/0.6/changeset/create
```

Пакет правок создания элементов карты:


```
<osm>
  <changeset>
    <tag k="created_by" v="JOSM 1.61" />
    <tag k="comment" v="Just adding some
      streetnames" />
    ...
  </changeset>
  ...
</osm>
```

В данном примере добавляются новые названия некоторых улиц.

2. Получение пакета правок по его id.

Пример вызова:

```
GET /api/0.6/changeset/#id
```

Возвращает пакет правок по данному id в формате OSM-XML:

```
<osm>
  <changeset id="10" user="fred" uid="123"
    created_at="2008-11-08T19:07:39+01:00" open="
    true" min_lon="7.0191821" min_lat="49.2785426
    " max_lon="7.0197485" max_lat="49.2793101">
    <tag k="created_by" v="JOSM 1.61" />
    <tag k="comment" v="Just adding some
      streetnames" />
    ...
  </changeset>
</osm>
```

3. Изменение пакета правок.

Пример вызова:

```
PUT /api/0.6/changeset/#id
```

Вносит изменение в пакет правок с данным id. Возвращает пакет правок, содержащий новую версию набора изменений.

4. Заккрытие пакета правок.

Т.к. сервер не знает, когда заканчивается одна серия правок и когда должна начаться другая, существует такое понятие как закрытие пакета правок. Пакет обычно закрывается вручную. Также, во избежание накопления незакрытых пакетов правок существует механизм автоматического закрытия пакетов.

Пример вызова:

```
PUT /api/0.6/changeset/#id/close
```

5. Получение всех изменений, связанных с данными пакетом правок

Пример вызова:

```
GET /api/0.6/changeset/#id/download
```

Возвращает OsmChange ⁸ документ, в котором описаны все изменения, связанные с данными пакетом правок.

3.1.6. Фасад над OpenStreetMap

Функциональность, связанная с передачей данных на сервер OSM не должна являться публичной для остальных компонент системы. Поэтому был выделен интерфейс, раскрывающий главную функциональность, необходимую для работы остальных компонент, который внутри своей реализации агрегирует несколько вызовов на сервер OSM.

```
public interface IMapApi {  
    List<GpsTrack> getTracks(BBox boundingBox);  
    void downloadMapPart(String targetFileName, BBox  
        boundingBox);  
    void updateWayTag(long wayId, String key, String  
        value);  
}
```

⁸OsmChange - формат данных для описания различий между двумя дампами данных OSM

```
void updateNodeTag(long wayId, String key, String
    value);
}
```

где BBox представляет объект, содержащий информацию о прямоугольнике, ограничивающем некую область на карте.

```
public class BBox {
    double left;
    double bottom;
    double top;
    double right;
}
```

Также, при помощи данной абстракции происходит отделение логики реализации карты от ее контракта. Таким образом, появляется теоретическая возможность легкой подмены одного картографического движка на другой. Она может понадобиться, например, если потребуется вносить изменения в Google Map Maker или любую другую карту.

3.2. Анализ GPS треков

Несмотря на то, что полученные с сервера OSM треки обладают минимальным набором информации, этого достаточно, чтобы на ее основе построить сложную систему классификации треков по типу транспортного средства.

3.2.1. Форматы треков

Независимо от формата представления трека, он должен содержать информацию о наборе точек, содержащих определенные параметры: широта, долгота, высота над уровнем моря, время прохождения данной точки.

Различные форматы накладывают свои ограничения на то, какие из параметров обязательные, а какие нет. В данной работе эти параметры

играют принципиальное значение, поэтому треки, в которых отсутствовала информация о времени прохождения точки, не учитывались.

Наиболее распространенным форматом хранения треков является формат GPX, основанный на XML. Большинство GPS-навигаторов используют этот формат для хранения треков. В данном формате OSM предоставляет доступ к трекам, которые загружают на сайт пользователи карты.

Также, в данной работе использовался формат PLT, который ничем принципиально не отличается от GPX кроме того, что он не основывается на XML, а является просто текстовым.

3.2.2. Координаты точек треков

Большинство популярных форматов хранения GPS треков (в том числе GPX и PLT) используют единую систему координат для записи широты и долготы точек трека WGS 84 ⁹.

Эта система определяет координаты относительно центра масс Земли, погрешность составляет менее 2 см. Широта представляет собой значение от -90 до 90. Долгота – от -180 до 180.

3.2.3. Вычисление расстояния между точками

Для вычисления расстояния между двумя точками в данной работе использовалась формула Гаверсинусов ¹⁰.

$$\Delta\sigma = 2 \arcsin \left\{ \sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos \phi_1 \cos \phi_2 \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right\},$$

где ϕ_i, λ_i – широта и долгота двух точек в радианах, $\Delta\lambda = (\lambda_2 - \lambda_1)$ – разница координат по долготу, $\Delta\sigma$ – угловая разница.

Для перевода углового расстояния в метрическое, нужно угловую разницу умножить на радиус Земли (6372795 метров), единицы конеч-

⁹Подробно про WGS 84: http://ru.wikipedia.org/wiki/WGS_84

¹⁰Подробно про формулу Гаверсинусов: <http://gis-lab.info/qa/great-circles.html>

ного расстояния будут равны единицам, в которых выражен радиус (в данном случае - метры).

3.2.4. Вычисление расстояния от точки до ломаной

В данной работе, кроме вычисления расстояния между двумя точками, также требовалось вычисление расстояния от точки трека до последовательно соединенного набора отрезков (в частности, дорожного графа). Такое расстояние вычислялось как минимальное расстояние от точки до входящих в ломаную отрезков. Вычисления расстояния от точки до отрезка было реализовано через решение треугольника по трем известным сторонам. Стороны треугольника были вычислены при помощи подхода, описанного в предыдущем параграфе.

Для трансформации была использована УТМ проекция¹¹.

3.2.5. Вычисление азимута

Для вычисления угла между направлением вектора, представленного двумя точками на карте, и направлением на север была использована Сферическая теорема косинусов¹².

$$\Theta = \text{atan2}(\sin \Delta\lambda \cos \phi_2, \cos \phi_1 \sin \phi_2 - \sin \phi_1 \cos \phi_2 \cos \Delta\lambda),$$

где ϕ_i, λ_i – широта и долгота двух точек в радианах, $\Delta\lambda = (\lambda_2 - \lambda_1)$ – разница координат по долготу, Θ – угол между направлением вектора и направлением на север.

3.3. Классификация треков по типу транспортного средства

Одной из основных задач данной работы является классификация GPS треков по типу транспортного средства. Для ее решения был выбран подход, описанный в статье [8].

¹¹Определение УТМ проекции: <http://goo.gl/eouoF>

¹²Подробно про Сферическую теорему косинусов: http://en.wikipedia.org/wiki/Spherical_law_of_cosines

В данном подходе большой акцент делается на то, что внутри одного трека может быть использовано несколько видов транспортного средства. Также, исходя из опыта работы с пользовательскими GPS треками, авторы статьи [8] выяснили, что большинство переходов с одного транспортного средства на другое происходит одним из следующих способов:

- переход с ходьбы на некоторое транспортное средство;
- переход с некоторого транспортного средства на ходьбу.

При этом, переход с некоторого транспортного средства на другое транспортное средство обычно осуществляется путем передвижения пешком. Однако, бывают случаи быстрого перехода с одного средства передвижения на другое без заметного пешего пути, но таким случаем решено пренебречь.

3.3.1. Сегментация треков на пешие и не пешие сегменты

Для нахождения точек переходов с пешего движения на другое транспортное средство или с некоторого транспортного средства на пешее движение был предложен следующий алгоритм.

Для удобства введем новые понятия, которые будут использоваться далее в алгоритме:

- $V(x)$ - фактическая скорость в данной точке;
- $A(x)$ - фактическое ускорение в данной точке;
- Сегмент - набор последовательных точек, при прохождении которых использовалось единственное транспортное средство, $s.length$ - количество точек сегмента s ;
- $Dist(s, i, j)$ - расстояние между i -ой и j -ой точками сегмента s ;
- $IsFoot$ - некая бинарная характеристика точки; обозначает, является ли данная точка пешей;

- $TimeSpan(s)$ - длительность прохождения сегмента пользователем;
 - $IsCertain$ - некая бинарная характеристика сегмента; обозначает, является ли данный сегмент конкретным, надежным.
1. Выбрав низкое значение скорости V_s и ускорения A_s , пометить все точки трека следующим образом:

$$IsFoot(x) = (V(x) < V_s) \vee (A(x) < A_s)$$

2. Если расстояние или время прохождения сегмента меньше некоторого порога V_{s_seg} , то слить его вместе с предыдущим сегментом.
3. Выбрав некоторые высокие пороги S_{seg} и T_{seg} , пометить все точки трека следующим образом:

$$IsCertain(s) = \sum_{i=0}^{s.length-2} Dist(s, i, i+1) > S_{seg} \wedge TimeSpan(s) < T_{seg},$$

4. Если количество подряд идущих сегментов, помеченных как $IsCertain == True$, больше некоторого порога, то объединяем их в один сегмент и помечаем его как $IsCertain = True$ и $IsFoot = False$.

Таким образом, получился список сегментов, в котором чередуются пешие и не пешие сегменты. После разбиения необходимо вычислить характеристики каждого из участков, так как дальнейшая классификация по типу транспортного средства будет проводиться на уровне сегментов.

3.3.2. Вычисление характеристик сегментов

В данном случае характеристики сегментов - это числовые величины, определяющие отличие сегментов различных транспортных средств друг от друга. В статье приводились два вида характеристик: Базовые и Сложные характеристики.

1. Базовые характеристики сегментов:

- (а) средняя скорость движения;
- (б) максимальная скорость движения;
- (в) максимальное ускорение движения.

2. Сложные характеристики:

- (а) Коэффициент изменения направления (HCR - Heading Change Rate) - показатель того, как часто трек меняет свою траекторию более чем на некое значение угла.

$$HCR = |P_c|/Distance,$$

где P_c - множество точек GPS треков, в которых пользователь меняет свое направление движения, $|P_c|$ - число элементов множества P_c , $Distance$ - длина сегмента.

- (б) Коэффициент количества остановок (SR - Stop Rate) - как часто скорость близка к нулю.

$$SR = |PS|/Distance,$$

где PS - набор точек GPS треков, $PS = \{P_{s1}, P_{s2}, \dots, P_{sn}\}$, $PS_{si} = \{P_i | p_i \in P, p_i * V < V_s\}$, V_s - некоторый порог скорости.

На рисунке 3 можно увидеть следующую закономерность $SR(Walk) > SR(Bus) > SR(Driving)$.

- (в) Коэффициент изменения скорости (VCR - Velocity Change Rate) - как часто ускорение превышает некое значение.

$$VCR = |P_v|/Distance,$$

где $P_v = \{p_i | p_i \in P, p_i * VRate > V_r\}$, $p_i * VRate = |V_2 - V_1|/V_1$ - точки, скорость которых $VRate$ превосходит заданный порог V_r скорости.

Пример можно посмотреть на рисунке 3.

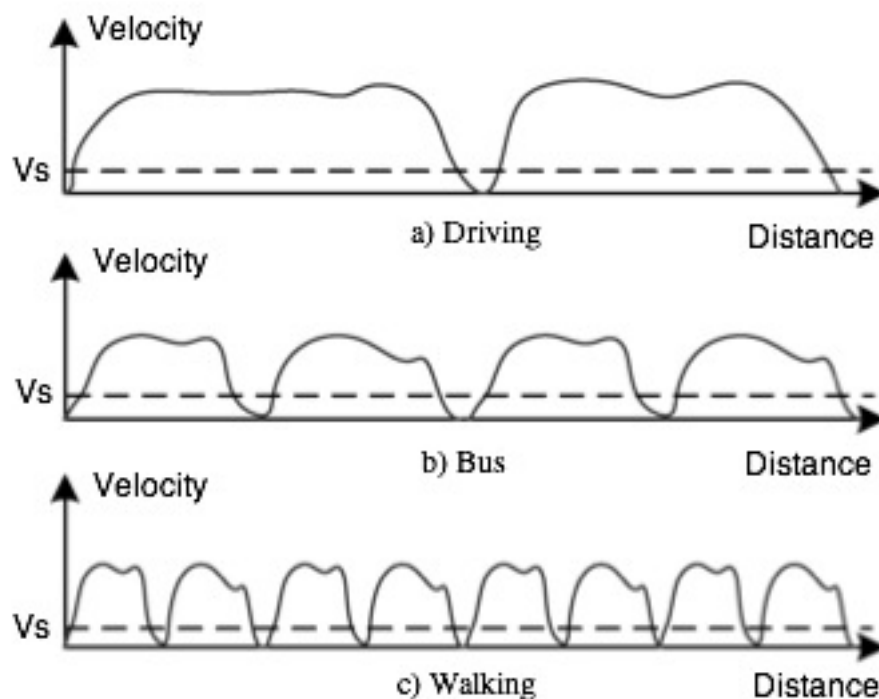


Рис. 3. Коэффициенты изменения скорости (Velocity Change Rate) и остановок (Stop Rate) для машины, автобуса и пешехода

В итоге, получены 6 различных характеристик, по которым будет проводиться классификация сегментов.

3.3.3. Обучение классификатора

Исходя из экспериментов, проведенных в статье [8], среди нескольких видов классификаторов выигрывает Дерево Решений (Decision Tree). Эксперименты в рамках данной работы также подтвердили, что этот вид классификаторов дает наилучший результат.

Среди бесплатных библиотек, реализующих алгоритмы машинного обучения, была выбрана библиотека с открытым исходным кодом WEKA ¹³, созданная в Университете Waikato.

Данный выбор был обусловлен следующими факторами:

1. широкий спектр решаемых задач (в их числе различные классификаторы, методы кластеризации);

¹³weka.sourceforge.net

2. стабильность;
3. относительно хорошая текстовая документация и множество обучающих видео-уроков вследствие высокой популярности библиотеки в интернете.

Данная библиотека использует определенный текстовый формат файлов ARFF для представления классифицируемых данных. Он определяет список объектов и характеристики этих объектов. Ниже приведен частичный пример файла данного формата:

```
@relation TraingSet
@attribute SegmentId string
@attribute MaxSpeed numeric
@attribute AverageSpeed numeric
@attribute MaxAcceleration numeric
@attribute Distance numeric
@attribute HeadingChangeRate numeric
@attribute StopRate numeric
@attribute VelocityChangeRate numeric
@attribute Vehicle {walk , car , bike , subway , bus}
@data
C:/Diploma/Data/YuZhengData/tracks/001/trajectory
/2007-05-07-GPS.plt_14-260 19.361484177525593
1.2467410643580223 0.32913551401496793
618.4228340880729 0.21021216040914492
0.36544575578820576 0.10834011344163623 car
C:/Diploma/Data/YuZhengData/tracks/001/trajectory
/2007-05-16-GPS.plt_13-116 3.709217793166626
2.7329757404276487 0.1505482744170059
612.0504976079377 0.02450778172491345
0.047381711334832675 0.0016338521149942302 walk
...
```

Для обучения Деревя Решений использовались тестовые данные ¹⁴,

¹⁴Данные доступны по ссылке <http://research.microsoft.com/apps/pubs/?id=141896>

предоставленные одним из авторов статьи [8]. Он представляет собой совокупность GPS треков в формате PLT, а также сопутствующие файлы, содержащие информацию о транспортном средстве на разных сегментах треков.

Процесс обучения классификатора можно описать следующим образом:

1. Из набора файлов получить множество GPS треков с известным типом способа передвижения в каждой точке.
2. Объединив последовательные точки с одинаковым транспортным средством, получить множество сегментов, помеченных некоторым транспортным средством.
3. Вычислить все 6 характеристик для каждого сегмента (см. раздел 3.3.2).
4. Создать ARFF-файл, содержащий характеристики сегментов и передать его в WEKA для обучения классификатора.

3.4. Уточнение OpenStreetMap с использованием информации о средстве передвижения

Обладая информацией о транспортном средстве, можно значительно улучшить онлайн-карты: изменять качество дороги, обозначать список возможных способов передвижения на данном участке пути, использовать в рекомендательных системах для туристических походов, оказывать рекомендательную помощь в муниципальном строительстве и ремонте дорог. Возможности расширения данной системы практически неограничены.

Однако, большая часть данной работы была сконцентрирована не вокруг самого процесса улучшения карты, и в постановку задачи попала лишь небольшая часть возможного спектра улучшений.

3.4.1. Общий подход уточнения карты

Одной из поставленных задач данной работы является улучшение OSM, а именно - изменение свойств дорог.

Ниже приведена последовательность действий, необходимых для достижения поставленной цели:

1. Выбрать квадрат онлайн-карты, который требуется улучшить.
2. Используя подсистему интеграции с OSM, получить список GPS треков, загруженных в данный квадрат и всех объектов карты внутри него.
3. Используя подсистему вычисления транспортного средства, получить треки, сегментированные таким образом, чтобы внутри каждого сегмента движение происходило с использованием единственного транспортного средства.
4. Выполнить желаемые улучшения картографической информации OSM.

В данной работе было выполнено улучшение OSM путем изменения двух тегов у объекта типа 'way' (дорога).

3.4.2. Сопоставление треков и дорожного графа

Для изменения дорог, вдоль которых двигался пользователь GPS трека, можно было бы выполнить полное сопоставления дорожного графа и траектории трека. Классические алгоритмы [3] делают большой акцент на то, чтобы путь, составленный из ребер дорожного графа, был непрерывен для корректного отображения пользователю. Однако, такая информация избыточна в контексте данной работы. В рамках поставленных задач достаточно обладать статистической информацией о том, какие точки попали на ту или иную дорогу. Данная информация получена из географической близости точек трека к дороге, и некоторые ошибки GPS-позиционирования не сильно повлияют на общий результат работы системы.

3.4.3. Уточнение тега 'maxspeed:practical'

Объект дороги в OSM обладает большим количеством свойств. Одним из этих свойств является тег 'maxspeed:practical'. Значение этого тега обозначает максимальную реальную скорость, которую развивают участники движения на данной дороге.

Для корректного улучшения значения данного тега рассматриваемых дорог требуется выполнить следующие действия:

1. Найти количество точек треков, находящихся в непосредственной близости к дороге.
2. Если удельное количество точек на 1 метр пути не превышает некоторый порог, то закончить алгоритм. Иначе, найти такую максимальную скорость X , что существует как минимум N точек среди найденных в пункте 1, скорость передвижения в которых больше, чем X . Таким образом можно будет утверждать, что новое значение не является некоторой погрешностью спутника или GPS-приемника.
3. Если значение максимальной скорости превышает текущее значение тега, выставить значение тега в найденную величину.

3.4.4. Уточнение тега 'smoothness'

Вторым свойством, которое изменяет данная система, является тег 'smoothness'. Оно обозначает качество дороги. Данный тег может иметь 8 значений¹⁵: 'excellent', 'good', 'intermediate', 'bad', 'very_bad', 'horrible', 'very_horrible', 'impassable'.

Исходя из простого логического смысла значений тега, можно ввести отношения «хуже» и «лучше» между различными значениями тега.

Для корректного улучшения значения данного тега рассматриваемых дорог требуется выполнить следующие действия:

¹⁵Подробное описание значений тега можно найти тут: <http://goo.gl/KLldp>

1. Найти количество точек треков, находящихся в непосредственной близости к дороге.
2. Если удельное количество точек на 1 метр пути не превышает некоторый порог, то закончить алгоритм. Иначе, посчитать среднюю скорость точек на данной дороге.
3. В зависимости от значения средней скорости при помощи эвристик, описанных ниже, изменить тег 'smoothness'.

Эвристики изменения тега 'smoothness' в зависимости от средней скорости (V_{mean}):

1. Если $V_{mean} > 60$ км/ч и значение тега «хуже», чем 'good', то нужно изменить значение на 'good'.
2. Если 50 км/ч $< V_{mean} < 60$ км/ч и значение тега «хуже», чем 'intermediate', то нужно изменить значение на 'intermediate'.

4. Тестирование системы

Данная система, как уже известно, состоит из набора слабосвязанных компонент. Для корректного функционирования подсистем было проведено их тестирование в изоляции. Тестами покрывались наиболее важные участки, такие как базовые формулы или алгоритмы.

4.1. Тестирование анализа GPS треков

Проводилось тестирование работы следующих задач:

- вычисление расстояния между точками на карте;
- вычисление расстояния между точкой и отрезком на карте;
- вычисление направления движения вектора, образованного точками на карте;
- вычисление скорости и ускорения в точке GPS трека.

Для функционального тестирования использовались данные с общеизвестных веб-карт типа Google.Maps и Яндекс.Карты. Были произведены замеры расстояния между точками как на больших расстояниях, так и на коротких. Тестирование проводилось на основе технологии JUnit.

4.2. Тестирование классификации GPS треков по типу транспортного средства

Не менее важной задачей является тестирование качества классификации GPS-треков по типу транспортного средства. Для решений данной проблемы был использован набор данных ¹⁶, предоставленный одним из авторов статьи [8] (как уже упоминалось в главе 3.3.3).

¹⁶Данные доступны по ссылке <http://research.microsoft.com/apps/pubs/?id=141896>

4.2.1. Данные для тестирования

Ниже приведена информация о тестовом наборе данных:

1. Общий размер данных - более 560 Мб(включающий некорректные данные).
2. Количество треков, содержащих непустые метки транспортного средства - 2018 треков.
3. Количество сегментов треков с непустыми метками транспортного средства - 6 600 сегментов.
4. Количество точек треков с непустыми метками транспортного средства - 3 446 171 точек.

Была проведена статистика по количеству используемых транспортных средств в наборе данных:

- пешком - 957,983 точек;
- автомобиль - 471,388 точек;
- автобус - 660,139 точек;
- велосипед - 317,170 точек;
- поезд - 233,562 точек.

В данном наборе находится большое количество некорректных данных, в том числе: количество треков, в которых не проставлена метка 'Время'. Несмотря на это, количество корректных данных остается внушительным и по нему возможно построить качественную классификацию.

4.2.2. Подход тестирования

Тестирование качества классификации проводилось при помощи библиотеки WEKA.

Процесс тестирования качества классификации заключался в следующем:

1. Прочитать тестовые данные, объединив их по сегментам.
2. Используя 66% тестового набора данных, посчитать характеристики на получившихся сегментах и обучить классификатор.
3. Используя оставшиеся 33% набора данных, вычислить предполагаемые сегменты при помощи алгоритма сегментации на пешие и не пешие сегменты, описанного в разделе 3.3.1.
4. Сравнить результаты классификации при помощи двух метрик тестирования классификации:
 - (а) По сегментам: $N_{success}/N_{total}$, где N_{total} – общее количество сегментов, $N_{success}$ – количество корректно вычисленных сегментов.
 - (б) По расстоянию: $S_{total}/S_{success}$, где S_{total} – общее расстояние, $S_{success}$ – расстояние между последовательными точками, в которых было корректно определено транспортное средство.

4.2.3. Результаты тестирования

При вычислении 5ти транспортных средств и использовании подхода тестирования, описанного выше, система классификации дает следующий процент корректно определенных транспортных средств:

- по сегментам - 55%;
- по расстоянию - 46%.

В статье [8], взятой за основу, полученные результаты имели точность около 70% по обеим метрикам. Из этого очевидно вытекает возможное улучшение данной работы в дальнейших исследованиях.

4.3. Эксперименты

Кроме автоматического тестирования был проведен ряд экспериментов, которые являлись по факту аналогом ручного тестирования, в процессе которого проводилось визуальное изучение и проверка промежуточных результатов на соответствие реальности.

4.3.1. Эксперименты по вычислению типа транспортного средства

В компоненте определения транспортного средства проводились эксперименты значимости тех или иных характеристик сегментов GPS треков. Для данной задачи был реализован экспорт характеристик в формат CSV ¹⁷. В процессе экспериментов CSV-файлы анализировались в Microsoft Excel при помощи графиков зависимости одних характеристик от других.

4.3.2. Эксперименты по уточнению тегов дорог в OpenStreetMap

В рамках уточнения карты OSM проводились эксперименты путем запуска системы напротив тестового сервера. Для тестирования рассматривались некоторые мировые столицы, однако тестовый сервер позволил получить данные значительного объема лишь из 2 столиц - Лондона и Вашингтона.

В процессе этих экспериментов вносились правки в работу алгоритма. Так благодаря им было принято решение не рассматривать дороги, в которых отсутствует тег 'highway', а также игнорировать дороги с тегами 'waterway' и 'railway'. Была подтверждена работоспособность данного алгоритма, были адекватно исправлены в тестовом режиме теги более 150 дорог в Лондоне и Вашингтоне.

¹⁷Простой текстовый формат представления данных в табличном виде

5. Дальнейшие исследования

В то время как статьи [2, 7, 8] под руководством одного и того же автора, послужили началом большого количества теоретических исследований в области онлайн-карт, данная работа может представлять собой основу для множества инженерных работ в области улучшения свободно редактируемых систем веб-картографии.

С этой точки зрения работа может рассматриваться как стартовая платформа для анализа и уточнения OSM.

Исследования по расширению системы:

1. Увеличение точности вычисления типа транспортного средства, посредством лучшей настройки классификатора, используя подход, описанный в статье [9].
2. Изменение других свойств дорог.
3. Распознавание мест остановок (отдыха, достопримечательностей) вдоль дорог.
4. Оказание помощи в муниципальном строительстве дорог.
5. Уточнение геометрии дорог.

Заключение

Поставленные цели данной дипломной работы были успешно выполнены:

- Реализован алгоритм классификации треков по типу транспортного средства, описанный в главе 3.3.
- Реализована компонента интеграции всей системы с сервером OpenStreetMap.
- Реализован алгоритм автоматического обновления свойств карты на основе информации о типе транспортного средства GPS треков.

Каждая компонента требовала отдельного исследования с нуля и отдельной реализации. Далее подробнее показан объем работы, выполненный для реализации каждой из них.

6.1. Алгоритм классификации треков по типу транспортного средства

Была выполнена подсистема классификации GPS треков по типу транспортного средства, обладающая достаточной точностью, которая позволяет выполнять автоматическое уточнение карты при помощи полученных данных.

Реализация компоненты включала в себя следующие задачи:

- Изучение существующих теоретических исследований в области классификации треков по типу транспортного средства. Выбор подходящего исследования для данной работы.
- Реализация выбранного теоретического подхода:
 - Изучение математических основ и алгоритмов анализа геолокационных данных.
 - Реализация вычисления базовых и сложных характеристик GPS треков, использованных в выбранном исследовании.

- Пошаговая реализация алгоритма.
- Классификация:
 - Выбор библиотеки машинного обучения для классификации.
 - Исследование возможности обучения классификатора на достоверных тестовых данных. Поиск тестовых данных.
 - Обучение классификатора на найденных тестовых данных.
 - Оценка успешности работы классификатора.

6.2. Подсистема интеграции с OpenStreetMap

Выполненная подсистема позволяет получить простой объектно-ориентированный доступ к данным карты, GPS трекам, публично загруженным на сервер OSM, а также производить любое желаемое изменение данных карты путем простого вызова метода. Данная подсистема может использоваться в большом количестве различных исследований в области картографии.

С другой стороны, вся остальная система абстрагирована от реализации картографического движка, а это значит, что ее можно использовать с любой другой онлайн-картой, позволяющей производить свободное редактирование своих данных при минимальном количестве усилий.

Выполненные задачи включают в себя:

- Изучение структуры OpenStreetMap.
- Изучение и реализации существующего OSM RESTful API.
- Реализация простого фасада, абстрагирующего сложную логику чтения и обновления данных OSM и GPS треков.

6.3. Подсистема автоматического уточнения OpenStreetMap

В рамках данной работы был реализован алгоритм автоматического уточнения карты OSM. Реализованная программа была запущена в тестовом режиме.

Для глобального запуска на релизном сервере система не готова из-за слишком дорогой стоимости ошибочных срабатываний алгоритмов. Подобный запуск должен быть тщательно проанализирован и оттестирован.

Данная система может быть расширена, что позволит ей в будущем, возможно, стать основой для глобального подхода к улучшению OSM.

Выполненные задачи включают в себя:

- Изучение возможностей автоматического обновления OSM.
- Изучение и поиск свойств карты, которые можно изменить исходя из знаний о типе транспортного средства.
- Создание и реализация алгоритма автоматического обновления выбранных свойств карты.

Список литературы

- [1] Coast Steve. OpenStreetMap // Официальный сайт OpenStreetMap. — 2004. — URL: <http://www.openstreetmap.org/> (дата обращения: 23.05.2013).
- [2] Learning transportation mode from raw gps data for geographic applications on the web / Yu Zheng, Like Liu, Longhao Wang, Xing Xie // Proceedings of the 17th international conference on World Wide Web. — WWW '08. — New York, NY, USA: ACM, 2008. — P. 247–256. — URL: <http://doi.acm.org/10.1145/1367497.1367532>.
- [3] Efficient Map-Matching of Large GPS Data Sets - Tests on a Speed Monitoring Experiment in Zurich, volume 244 of Arbeitsbericht Verkehrs und Raumplanung : Rep. ; Executor: F. Marchal, J. Hackney, K. W. Axhausen : 2004.
- [4] Mining GPS Traces for Map Refinement / Stefan Schroedl, Kiri Wagstaff, Seth Rogers et al. // Data Min. Knowl. Discov. — 2004. — . — Vol. 9, no. 1. — P. 59–87. — URL: <http://dx.doi.org/10.1023/B:DAMI.0000026904.74892.89>.
- [5] Neis Pascal, Zielstra Dennis, Zipf Alexander. The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011 // Future Internet. — 2011. — Vol. 4, no. 1. — P. 1–21. — URL: <http://www.mdpi.com/1999-5903/4/1/1>.
- [6] Transportation mode detection using mobile phones and GIS information / Leon Stenneth, Ouri Wolfson, Philip S. Yu, Bo Xu // Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. — GIS '11. — New York, NY, USA: ACM, 2011. — P. 54–63. — URL: <http://doi.acm.org/10.1145/2093973.2093982>.
- [7] Understanding mobility based on GPS data / Yu Zheng, Quannan Li, Yukun Chen et al. // Proceedings of the 10th international conference on

Ubiquitous computing. — UbiComp '08. — New York, NY, USA: ACM, 2008. — P. 312–321. — URL: <http://doi.acm.org/10.1145/1409635.1409677>.

[8] Understanding transportation modes based on GPS data for web applications / Yu Zheng, Yukun Chen, Quannan Li et al. // ACM Trans. Web. — 2010. — . — Vol. 4, no. 1. — P. 1:1–1:36. — URL: <http://doi.acm.org/10.1145/1658373.1658374>.

[9] Using mobile phones to determine transportation modes / Sasank Reddy, Min Mun, Jeff Burke et al. // ACM Trans. Sen. Netw. — 2010. — . — Vol. 6, no. 2. — P. 13:1–13:27. — URL: <http://doi.acm.org/10.1145/1689239.1689243>.