

РАЗРАБОТКА СЕРВИСА ДЛЯ ФИКСАЦИИ ОТКЛОНЕНИЙ ОТ СХЕМЫ МАРШРУТА ДОРОЖНОГО ТРАНСПОРТА

Аннотация. В данной статье представлена реализация алгоритма для корректировки данных GPS/Глонасс, поступающих с навигационного оборудования, установленного в маршрутном транспорте. Полученные скорректированные данные будут использоваться для фиксации отклонений от запланированной схемы движения и графика маршрутного транспорта.

Ключевые слова: дорожный граф, GPS/Глонасс-трек, схема маршрута.

Введение

Одной из функций информационной системы РНИС “Автовокзал” является контроль за движением по схеме и графику маршрутного транспорта.

Для фиксации отклонений от схемы необходимо представить GPS/Глонасс-трек транспортного средства и схему маршрута как наборы ребер дорожного графа, разница которых и будет являться отклонениями от схемы маршрута.

Для каждой точки трека требуется получить проекцию на геометрию ребра дорожного графа, представляющего из себя ломаную линию, и вычислить к какому из ребер относится полученная проекция.

Фактический пробег транспорта в рамках схемы маршрута в соответствии с запланированным графиком считается как сумма длины линии пройденных ребер, за исключением ребер, не входящих в схему маршрута, в каждом рейсе графика.

Для получения GPS/Глонасс-трека автомобильного транспорта в виде набора связанных ребер дорожного графа был реализован алгоритм корректировки данных трека.

Описание исходных данных

В качестве дорожного графа используется база данных *Corrector*, созданная на основе данных *OpenStreetMap (OSM)*. Дорожный граф является ориентированным графом, вершины которого представлены строками в таблице “*road_segments*”, а ребра графа в столбце “*related*” каждой строки. В таблице 1 представлено описание полей вершины дорожного графа.

Таблица 1. Описание полей таблицы “*roads_segments*”

Имя	Тип	Описание
id	bigint	Идентификатор вершины
highway	varchar(255)	Тип дороги
distance	double precision	Протяженность геометрии в км
osm_id	bigint	Идентификатор OSM
oneway	smallint	Одностороннее движение (0 или 1)
related	bigint[]	Массив идентификаторов вершин, с которыми вершина имеет связь
geometry	geometry	Геометрия вершины в формате Well-known binary [2]
azimuths	double precision[]	Направления сегмента в разных точках геометрии
reverses	double precision[]	Обратные направления сегмента в разных точках геометрии

Навигационные данные о движении транспорта РНИС “Автовокзал” хранятся в базе данных *Receiver*, в таблице “*marks*”, основные поля которой приведены в таблице 2.

Треком называется выборка из таблицы “*marks*” по уникальному идентификатору навигационного прибора и временному промежутку.

Таблица 2. Описание таблицы “marks”

Имя	Тип	Описание
id	bigint	Идентификатор отметки
datetime	timestamp with time zone	Дата и время формирования отметки
device_id	integer	Идентификатор навигационного прибора
latitude	double precision	Широта
longitude	double precision	Долгота
received_time	timestamp with time zone	Время доставки отметки на сервер
satcount	smallint	Количество спутников, с помощью которых формировалась отметка
speed	numeric(7,2)	Скорость движения
hdop	integer	Снижение точности в горизонтальной плоскости

Каждый элемент трека, помимо информации о местоположении, содержит в себе характеристики, позволяющие оценить качество позиционирования, такие как скорость движения, количество спутников, относительно которых происходит позиционирование, и качественный показатель HDOP (Horizontal Dilution of Precision).

HDOP – показатель снижения точности в горизонтальной плоскости. Чем меньше значение данного показателя, тем выше точность позиционирования. HDOP оценочный показатель, поэтому его нельзя перевести в физические величины, такие как, например, метры.

Опираясь на результаты исследования, приведенные в статье Stopher P.R. [1], можно использовать соответствия показателя HDOP и радиуса возможного отклонения от фактического местоположения, представленные в таблице 3.

Таблица 3. Зависимость отклонения от значения HDOP

Показатель HDOP	Радиус отклонения, м
1-3	менее 5
3-5	от 5 до 10
5-10	от 10 до 100
10 и более	более 100

На основе этих данных отметки с показания HDOP более 10 не следует учитывать в процессе коррекции трека.

Алгоритм привязки GPS-данных к дорожному графу

При разработке приложения была использована оценочная формула для инкрементального алгоритма привязки данных на основе оценки расстояния (S_d) и направления (S_α) [2].

1. Составляющая S_d учитывает взвешенное расстояние от точки трека до ребра:

$$S_d(p, c) = \mu_d - \alpha * d(p, c)^{n_d},$$

где $d(p, c)$ – расстояние от точки p до ребра c $\{ d(p, c) > 0 \}$,

μ_d, n_d, α – эмпирические константы $\{ \mu_d > 0, 0 \leq \alpha \leq 360^\circ, n_d > 0 \}$

На рисунке 1 представлен график функции S_d для различных значений констант (μ_d, n_d, α). При увеличении значения расстояния d от точки трека до линии ребра оценка S_d уменьшается. Параметр n_d влияет на чувствительность функции к изменениям значения d ($n_d = 1$ – линейная зависимость, $n_d = 2$ – квадратичная).

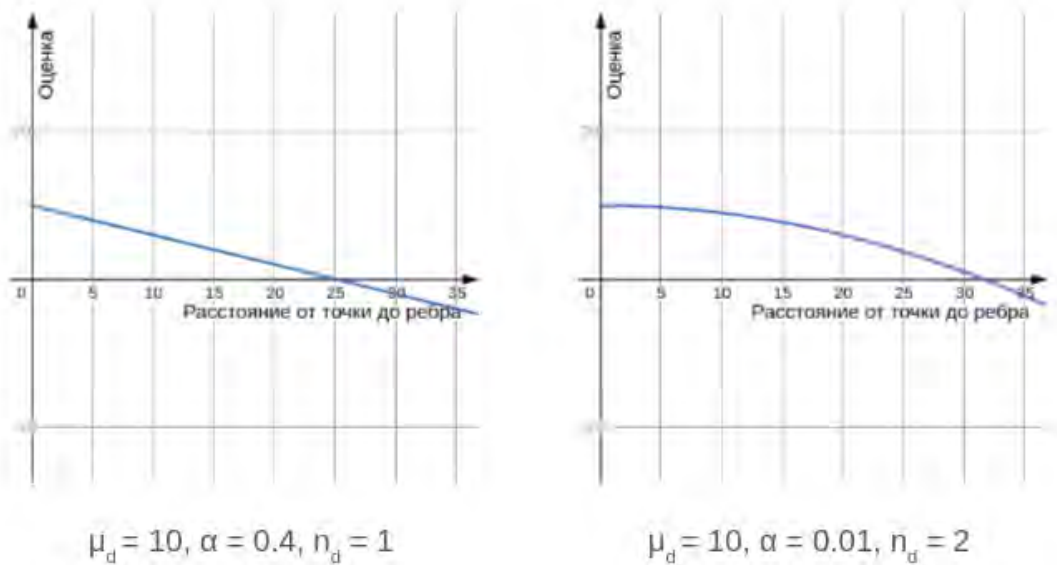


Рис. 1. График функции составляющей S_d

2. Составляющая S_α учитывает угол между направлением линии ребра и направлением вектора, соединяющего соседние точки трека:

$$S_\alpha(p, c) = \mu_\alpha * \cos(a)^{n_\alpha},$$

где μ_α, n_α – эмпирические константы $\{\mu_\alpha > 0, n_\alpha > 0\}$,

a – разность значений направления линии ребра и направления вектора, соединяющего соседние точки трека $\{0 \leq a \leq 360^\circ\}$.

На рисунке 2 представлен график функции S_α для различных значений констант (μ_α, n_α) . Параметр n_α влияет на чувствительность функции к изменениям значения разности значений a . При приближении a к значению 90° оценка S_α уменьшается. Для нечетного значения n_α разница направлений от 90° до 270° даст результат с отрицательным показателем оценки S_α , это можно использовать в ситуациях, когда по дороге доступно движение лишь в одном направлении.

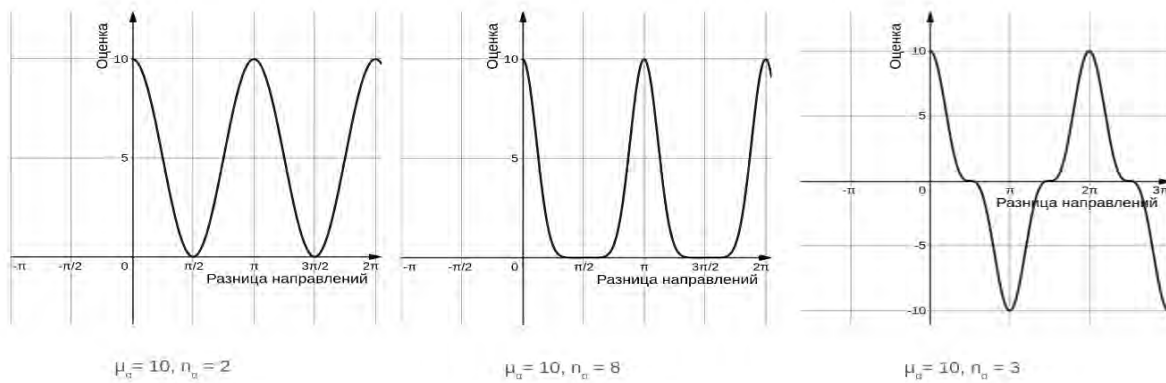


Рис. 2. График функции составляющей S_α

Для алгоритма важны значения параметров μ_d и μ_α относительно друг друга. В случае, когда $\mu_d > \mu_\alpha$, расстояние до ребра будет иметь больший вес, чем разница направлений, т. е. имеется возможность варьировать вес оценки S_d (расстояния) и S_α (направления).

Итоговая метрика:

$$S(p, c) = S_d(p, c) + S_\alpha(p, c).$$

Для увеличения точности можно получить оценку наборов всех возможных вариантов последовательности ребер на глубину $depth$ и выбрать набор с максимальным значением оценки S :

$$S(p_i, c_j) = \sum_{k,l=0}^{depth} S(p_{i+k}, c_{j,l}), \quad (1)$$

где p_i – i -ая точка трека, c_j – j -ое ребро графа дорог.

Итоговый алгоритм

1. Выбрать все ребра графа с линиями, пересекающими буферную зону первой точки трека;
2. Получить наборы связных ребер глубиной $depth$;
3. Оценить все полученные наборы с помощью формулы (1);
4. В наборе с наибольшей оценкой выбрать ребро, на которое попала проекция точки, а также все, предшествующие ему. Сделать это ребро текущим и добавить его к готовому маршруту;
5. Перейти к пункту 2.

Разработка сервиса

Для реализации сервиса были использованы:

- Язык программирования Python версии 3.8;
- Библиотека Shapely [3], для работы с линиями ребер и точками трека;
- Библиотека psycopg2 [4] для работы с базами данных.

При реализации алгоритма возникла необходимость представления дорожного графа и трека автомобиля в оперативной памяти, так как постоянные переходы между ребрами графа и точками трека вынуждали обращаться к базе данных, из-за чего процесс обработки занимал много времени. Для этого были реализованы классы, диаграмма которых представлена на рисунке 3.

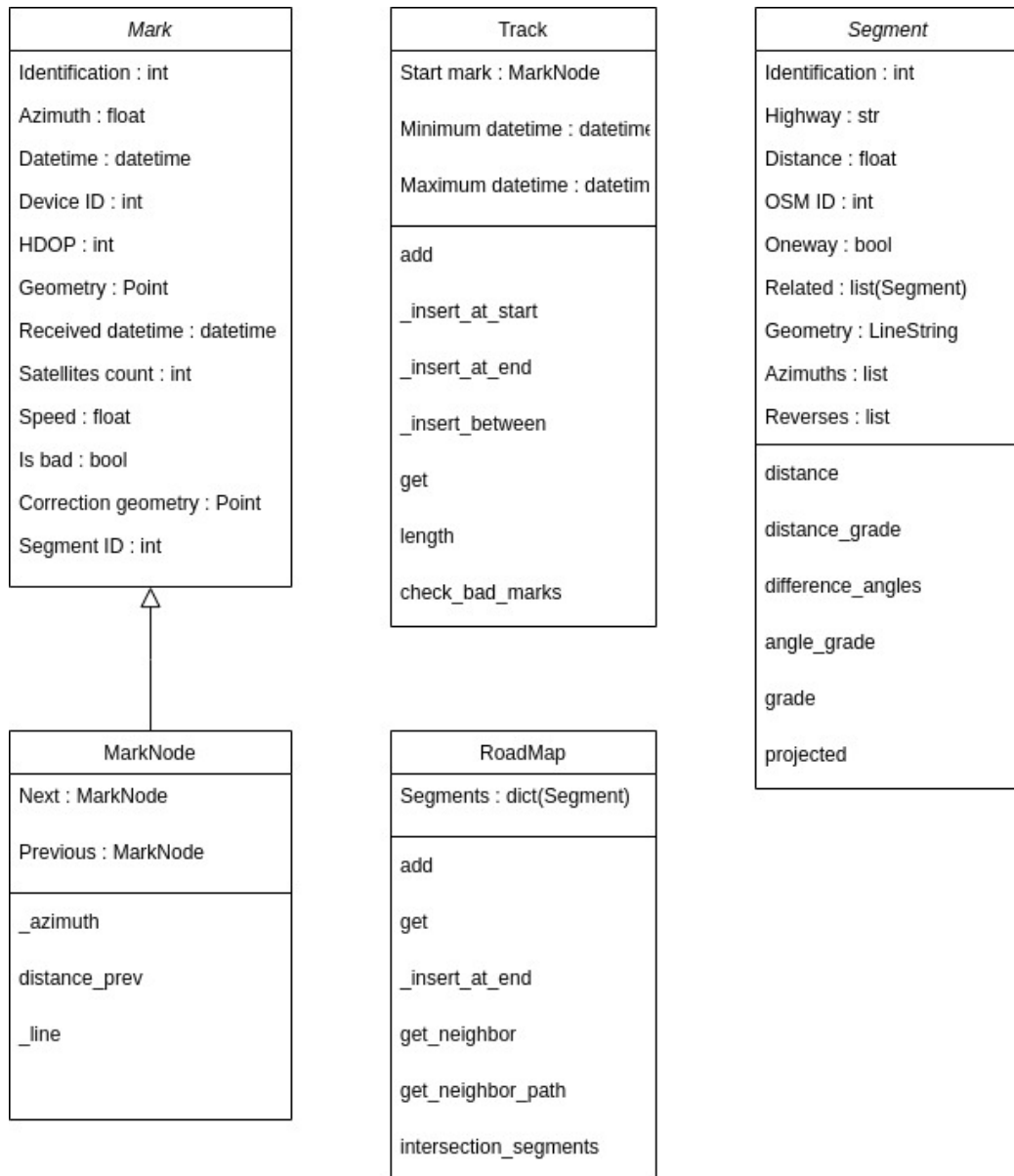


Рис. 3. Диаграмма классов приложения коррекции отметок

Трек автомобиля представлен как двусвязный список (класс `Track`), это позволяет контролировать порядок отметок по времени с помощью методов `_insert_at_start`, `_insert_at_end`, `_insert_between`, а также легко обращаться к предыдущей и следующей отметке.

Дорожный граф (класс `RoadMap`) представлен как словарь, ключами которого являются идентификаторы ребер, а в параметре “related” в массиве хранятся ссылки на связанные ребра.

Метод `intersection_segments` позволяет получить список сегментов, пересекающих буферную зону, заданную точкой и радиусом, это необходимо для обработки первой точки трека.

Метод `get_neighbor` – получение списка связанных с указанным ребром ребер.

Метод `get_neighbor_path` – рекурсивный метод для получения наборов вариантов движения по дорожному графу.

Оценочная формула (1) была реализована в классе `Segment`, в методах `distance_grade` – оценка S_d , `angle_grade` – оценка S_α , `grade` – общая оценка S .

Результат работы алгоритма

На рисунке 4 приведен пример работы алгоритма, в котором точка проецируется на ближайшее ребро графа, на рисунке 5 пример работы алгоритма с использованием оценочной формулы (1).

На рисунках видно, что первый подход неэффективен, так как:

1. В итоговом наборе ребер будут содержаться ребра, по которым транспорт не проезжал. Это произошло по причине того, что поиск ближайшего ребра не учитывает корректировку для последующих точек трека, а также направление движения транспорта и ребра.
2. Имеются пропущенные ребра. Поиск ближайшего ребра осуществляется с помощью буферной зоны диаметром 100 метров для каждой точки трека, поэтому не учитывается связь с предыдущим полученным ребром дорожного графа.

Данные проблемы решаются благодаря алгоритму привязки данных с использованием оценочной формулы (1).

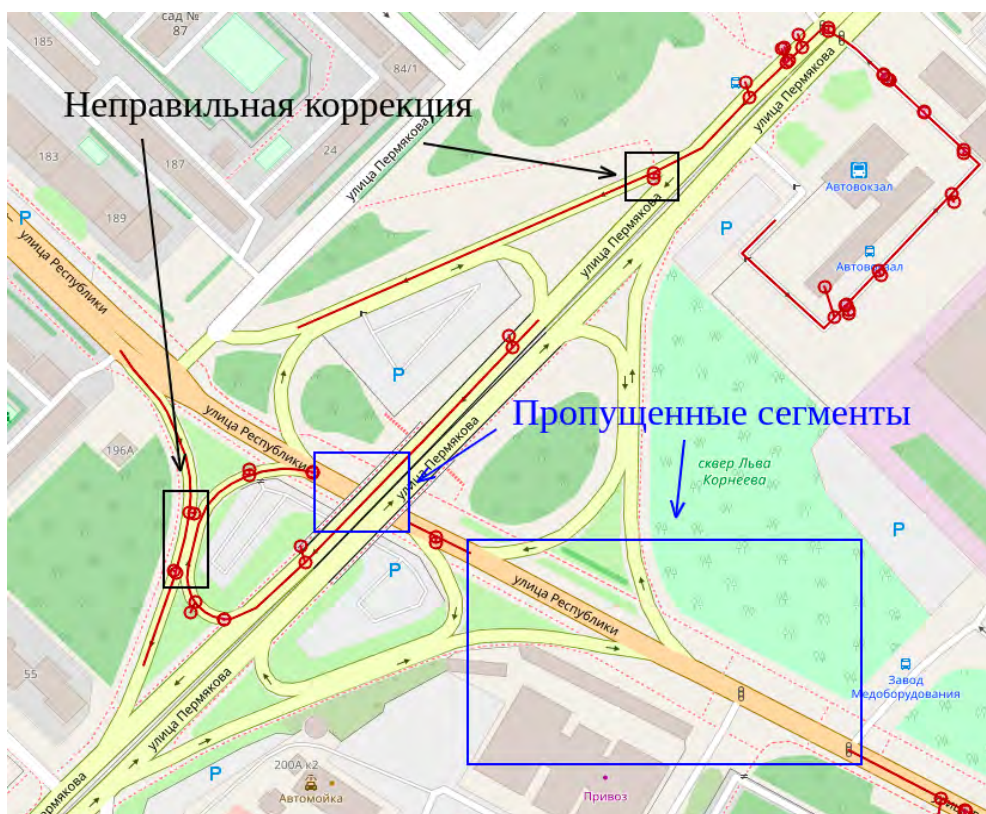


Рис. 4. Работа алгоритма поиска ближайшего ребра



Рис. 5. Работа алгоритма с использованием оценочной формулы (1)

СПИСОК ЛИТЕРАТУРЫ

1. Stopher P.R., Daigler V., Griffith S. Smartphone app versus GPS Logger: A comparative study // Transportation Research Procedia. – 2018. – Т. 32. – С. 135-145.
2. Brakatsoulas S., Pfoser D., Salas R., Wenk C. On map-matching vehicle tracking data // Very large data bases (VLDB '05). – 2005. – Т. 31. – С. 853-864.
3. Документация библиотеки Shapely [Электронный ресурс]. – URL: <https://shapely.readthedocs.io/> (Дата обращения: 13.05.2021, режим доступа: свободный)
4. Документация библиотеки psycorg2 [Электронный ресурс]. – URL: <https://www.psycorg.org/docs/> (Дата обращения: 10.04.2021, режим доступа: свободный)