



Петрозаводский государственный университет
Кафедра информатики и математического обеспечения



Владислав Викторович Клименко

Реализация мобильного приложения интеллектуального персонифицированного геолокационного трекера пользователя

Отчет о научно-исследовательской работе

Научный руководитель: преподаватель В. М. Димитров

Предметная область

Трекер - это мобильное приложение, позволяющее отслеживать перемещения пользователя в пространстве при помощи спутниковых систем (например, GPS). Также трекер может предоставлять различную аналитическую информацию по перемещениям (список перемещений, графики скорости, высот рельефа, ускорений и т.д.).

Точка - это положение пользователя в пространстве и времени, характеризующееся четырьмя составляющими: координата широты, координата долготы, высота над уровнем моря, метка времени.

Трек - набор точек перемещения пользователя, логически объединённых при перемещении из стартовой точки в конечную и отсортированных по метке времени.



Цель и задачи

Цель работы

Разработать энергоэффективный трекер, фиксирующий местоположение пользователя и самостоятельно формирующий треки.

Задачи

- Реализовать демо приложение для выбора оптимальной частоты опроса.
- Разработать алгоритм разбиения набора точек на треки.
- Реализовать трекер-приложение



Обзор аналогов

Наиболее популярные геотрекеры вынуждают пользователя вручную запускать запись трека:

- A-GPS Tracker
- Sportractive GPS Running Cycling Distance Tracker
- Geo Tracker - GPS tracker
- GPS Sports Tracker App: running, walking, cycling
- Strava Training: Track Running, Cycling And Swimming
- Runtastic Running App



Уровень энергопотребления

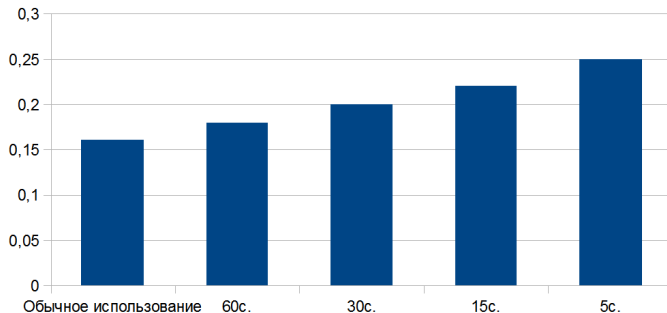
Исследования проводились при следующих условий:

- Устройство - ZTE Blade X3
- ОС - Android 5.1
- Google Play Services 49
- Количество экспериментов: 6
- Продолжительность одного: 10 часов



Уровень энергопотребления

Энергопотребление за 10 часов работы телефона



Алгоритм разбиения на треки

Код для определения точки остановки

CheckSpeed:

```
dist = sqrt(111.1111111111 * ((x1 - x2) ^ 2 +  
    cos((x1 + x2) / 2) * (y1 - y2) ^ 2))  
time = to_hourse(t1 - t2)  
return dist / time <= 2 or  
    dist / time >= 150
```



Алгоритм разбиения на треки

Модифицированный алгоритм для определения точки остановки

```
checkSpeed:
    dist = 0;
    time = 0;
    for i in len(points):
        x1, y1, t1= points.get(i)
        x2, y2, t2 = points.get(i % len(points))
        dist += sqrt(111.111111111 * ((x1 - x2) ^ 2 +
            cos((x1 + x2) / 2) * (y1 - y2) ^ 2))
        time += to_hourse(t1 - t2)
    return dist / time <= 2 or
        dist / time >= 150
```



Описание алгоритмов

Первый алгоритм вычисляет среднюю скорость между текущей и предыдущей точкой. Если их скорость не входит в интервал от 2 до 150, то это точка остановки.

Второй алгоритм работает аналогичным образом, однако средняя скорость вычисляется для большего числа точек. При вычислении используется 5 предыдущих точек и текущая. Если скорость не входит в интервал от 2 до 150, то текущая точка является точкой остановки.

Второй алгоритм выдаёт большее количество треков. При их применении на один и тот же набор точек, полученный за 24 часа первый алгоритм определил 349 треков, а второй показал 583.



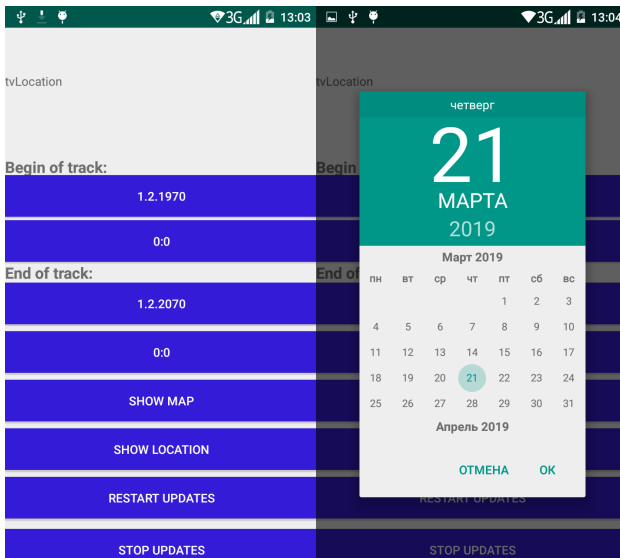
Заключение

Полученные результаты

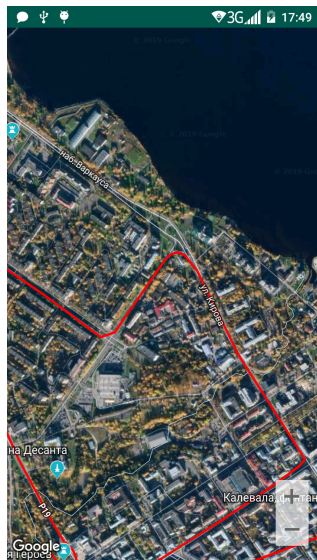
- Найдена оптимальная частота опроса
- Разработан алгоритм разбиения координат на треки
- Написано основное приложение



Скриншот приложения



Скриншот приложения



Спасибо за внимание!

