Петрозаводский государственный университет
Математический факультет
Кафедра информатики и математического обеспечения

Промежуточный отчет

Идентификация маршрута мобильного пользователя по набору GPS координат.

Выполнил: студент 3 курса
Сало А. Ю.

Научный руководитель: кандидат технических наук, доцент кафедры ИМО
Богоявленская О. Ю.

Представлен: ______________
2002 г.

Петрозаводск
2002

Содержание

1 Введение

В связи с развитием мобильных систем и сетей возникает необходимость в эффективных средствах для их анализа и диагностики. Одной из важнейших характеристик любой сети является ее производительность. Идентификация маршрута мобильного пользователя по набору его GPS координат — задача, предшествующая прогнозированию характеристик производительности при мобильной передаче данных в коммуникационных системах.

Задача идентификации маршрута мобильного пользователя тесно связана с задачей приближения или упрощения линии[1] в картографии. Наиболее распространенным алгоримтмом для решения этой проблемы является алгоритм Дугласа-Пекера. Однако, одним из основных его недостатков является то, что приходится хранить в массиве все точки до следующей точки поворота и выполнять упрощение для них каждый раз заново. Это происходит потому, что новые точки, отображающие перемещение мобильного пользователя, поступают через некоторые промежутки времени, т. е. они не известны все сразу. Поэтому целью этой работы является написание нового алгоритма, который будет экономно расходовать память и время и учитывать то, что все точки изначально не известны.

2 Постановка проблемы

Введем некоторые понятия. Будем называть идеальной траекторией линию $L(x) = kx + b$, которая является близкой к наблюдаемой траектории (обозначим ее $S(t)$). Введем классификацию траекторий $S(t)$:

1. Будем считать траекторию $S(t)$ *сильно регулярной*, если существует идеальная траектория, такая что угол α между любым вектором $(S(t_2); S(t_1))$, $t_2 > t_1$ и идеальной траекторией меньше, чем $\pi/4$.

2. Будем считать траекторию $S(t)$ *регулярной*, если существует идеальная траектория, т. ч. угол α между любым вектором $(S(t_2); S(t_1))$, $t_2 > t_1$ и идеальной траекторией меньше, чем $\pi/2$.

3. Если $\alpha > \pi/2$, то траектория *нерегулярная*.

Наибольший интерес представляют сильно регулярные траектории, так как маршрут мобильного пользователя в случае просто регулярной или нерегулярной траектории будет идентифицироваться "лишними" точками, которые в действительности не являются точками поворота.

[1] line simplification problem

Поэтому необходимо выполнять фильтрацию точек. Фильтрация реализуется следующим образом: если принять ширину дороги за ϵ, то расстояние между соседними полученными точками должно быть не меньше $\sqrt{2}\,\epsilon$. Если же это не так, то последняя полученная точка просто отбрасывается и ожидается поступление следующей точки.

Существует два подхода к решению задачи идентификации маршрута мобильного пользователя. В первом случае считается, что нам известна карта местности, по которой движется пользователь. Во втором случае карта является неизвестной.

2.1 Алгоритм с использованием карты

Будем считать, что нам с определенной точностью известна карта местности, по которой движется пользователь. Это следует понимать в том смысле, что задан граф, например, в виде списка, вершинами которого являются ключевые узлы на местности (при движении пользователя по городу это перекрестки), а ребрами — дороги, соединяющие эти точки (улицы). Тогда маршрут мобильного пользователя будет идентифицироваться последовательностью узловых точек.

Обозначения:

tmp — текущий узел на карте;
ver — узел, в котором может быть совершен поворот;
stv — узел, соответствующий узлу ver;
maxcos — максимальное значение косинуса.

Алгоритм:

1. Получить координаты первой точки и вычислить по ним tmp (начальный) узел.

2. Если координаты точек больше не поступают, то перейти к 9

3. Получить координаты новой точки.

4. Если траектория не сильно регулярна, то перейти к 2.

5. Вычислить maxcos для узла tmp и выбрать соответствующий ему узел ver.

6. Вычислить maxcos для узла ver и выбрать соответствующий ему узел stv.

7. Если узлы tmp и stv не совпадают, то запомнить tmp и сделать текущим узлом узел ver.

8. Перейти к 2

9. Запомнить узел tmp.

10. Конец.

2.2 Алгоритм без использования карты

Будем считать, что карта нам не задана. Тогда идентификация маршрута мобильного пользователя будет заключаться в выделении тех точек, в которых произошел поворот, из числа полученных точек.

Обозначения:
Δx, Δy — знак проекции вектора на соответствующую ось координат.
i_x, i_y — индикаторы, показывающие знак проекции предыдущего вектора на соответствующую ось координат.
c_x, c_y — счетчики шагов, в течение которых знак проекции вектора на соответствующую ось координат оставался постоянным.

Алгоритм:

1. Получить координаты первой и второй точек.

2. $\Delta x = \mathrm{sgn}(x_2 - x_1)$; $\Delta y = \mathrm{sgn}(y_2 - y_1)$.

3. Присвоить $i_x = \Delta x$; $i_y = \Delta y$; $c_x = 1$; $c_y = 1$.

4. Если точек больше нет, то перейти к 14.

5. Получить координаты n-ной точки x_n и y_n.

6. Если траектория не сильно регулярна, то перейти к 4.

7. $\Delta x = \mathrm{sgn}(x_n - x_{n-1})$; $\Delta y = \mathrm{sgn}(y_n - y_{n-1})$.

8. Если $\Delta x = i_x$, то $c_x = c_x + 1$.

9. Если $\Delta y = i_y$, то $c_y = c_y + 1$.

10. Если ($\Delta x \mathrel{!}= i_x$) и ($\Delta y \mathrel{!}= i_y$), то перейти к 4.

11. Если $\Delta x \mathrel{!}= i_x$, то {
Если $c_x > c_y$, то запомнить $(n-1)$-ю точку.
$i_x = \Delta x$ и $c_x = 1$}.

12. Если $\Delta y \mathrel{!}= i_y$, то {
Если $c_y > c_x$, то запомнить $(n-1)$-ю точку.
$i_y = \Delta y$ и $c_y = 1$}.

13. Перейти к 4.

14. Запомнить $(n-1)$-ую точку.

15. Конец.

Если используются две системы координат, то данный алгоритм применяется для каждой из них в отдельности. Координаты точки в системе координат OXY преобразуются в координаты точки в системе координат OX^1Y^1 так:

$$\begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix},$$

где φ — угол поворота системы координат OX^1Y^1 относительно системы координат OXY. В частности, при повороте OX^1Y^1 на угол $-\frac{\pi}{4}$ относительно OXY (ось OY^1 лежит в первой четверти системы координат OXY) получим:

$$\begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} \cos(-\frac{\pi}{4}) & \sin(-\frac{\pi}{4}) \\ -\sin(-\frac{\pi}{4}) & \cos(-\frac{\pi}{4}) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

Если хотя бы в одной системе координат произошел поворот, то точка запоминается.

Список литературы

[1] Olga I. Bogoiavlenskaia. Several approaches to the algorithmic analysis of the problems of predicting quality-of-service for nomadic user workstation.