УДК 004.67

БЛОЧНО-ВРЕМЕННОЙ АЛГОРИТМ ФИЛЬТРАЦИИ ГЕОЛОКАЦИОННЫХ ДАННЫХ

© 2013 Н.В. Бейлина[1]

В работе предложен простой и эффективный алгоритм фильтрации потоковых геолокационных данных.

Ключевые слова: геолокационные данные, потоковая обработка данных.

1. Постановка задачи

Геолокационные данные, описывающие перемещение наблюдаемого объекта, представляют собой последовательность кортежей вида $(lon, lat, time, ...)$, где lon, lat — географические координаты объекта (широта и долгота), $time$ — время получения координаты, а многоточием обозначены дополнительные данные, такие как высота над уровнем моря, мгновенная скорость и так далее.

Геолокационные данные, поступающие в информационные системы от датчиков GPS/Глонасс, зачастую избыточны: к примеру, многие датчики передают координаты один раз в секунду, тогда как для реального применения достаточно данных с точностью до минуты, а иногда существенно реже.

Предположим, что в информационную систему передаются лишь широта, долгота, штамп времени (по 64 бита), высота над уровнем моря и скорость (по 16 бит). Без учета накладных расходов каждая запись имеет размер 28 байт. Однако, если данные поступают раз в секунду, за сутки от одного наблюдаемого объекта в систему поступит около 2,3 МБайт данных, 840 Мбайт в год. Понятно, что построение различных аналитических отчетов по таким объемам может быть затруднительно для небольших организаций, не обладающих оборудованием с соответствующей вычислительной мощностью.

С учетом того, что информация, по эмпирическим подсчетам, избыточна приблизительно в 60 раз, весьма актуальным является вопрос фильтрации поступающих в информационную систему данных, по возможности осуществляемый одновременно с приемом данной информации либо с небольшой задержкой, но небольшими блоками.

Очевидно, что (lon, lat), расположенные в порядке возрастания времени, представляют собой вершины ломаной. Для упрощения ломаных линий часто используется алгоритм Рамера — Дугласа — Пекера [1; 2]. Он отличается простотой

[1]Бейлина Наталья Викторовна (natalie@samdiff.ru), кафедра высшей математики и информатики Самарского государственного технического университета, 443011, Российская Федерация, г. Самара, ул. Акад. Павлова, 1.

реализации, высокой эффективностью, а его сложность оценивается как $O(n^2)$. Именно этот алгоритм используется в большинстве геоинформационных систем для отображения конечному пользователю траектории движения наблюдаемого объекта на карте.

Формально алгоритм Рамера — Дугласа — Пекера можно применить к любой ломаной, т. е. к любой части имеющихся данных. Алгоритм сохраняет т. н. "характерные" точки ломаной, удаляя из нее те, что лежат на расстоянии, не превосходящем ε от прямой, соединяющей другие точки. Однако среди этих точек могут оказаться также и точки, несущие дополнительную смысловую нагрузку, например, точки длительного простоя наблюдаемого объекта или промежуточные точки на длинных прямолинейных участках магистралей — они являются излишними с точки зрения алгоритма Рамера — Дугласа — Пекера, но могут являться важным элементом для других бизнес-процессов предприятия, эксплуатирующего информационную систему.

Типичными "потерями" при применении алгоритма Рамера — Дугласа — Пекера (и многих других алгоритмов упрощения ломаных) к данным геолокации являются:

- потеря мест "простоя" наблюдаемого объекта, когда становится невозможно определить, как долго на самом деле находился объект в окрестности некоторой точки;
- потеря промежуточных точек наблюдаемого объекта при его движении по прямолинейному шоссе.

Это связано с тем, что алгоритм Рамера — Дугласа — Пекера учитывает лишь расстояния (в простейшем случае — на плоскости), но не учитывает время.

Кроме того, многие алгоритмы упрощения ломаных, и в частности алгоритм Рамера — Дугласа — Пекера, не являются поточными, т. е. требуют наличия всех входных данных сразу, в данном случае — всей ломаной, тогда как имеется необходимость в блочно-поточной фильтрации поступающих данных.

Цель данной работы — разработать простой блочно-временной алгоритм фильтрации геолокационных данных, который позволяет сохранять дополнительные характерные точки, такие как:

- "точки простоя" наблюдаемого объекта,
- "контрольные точки" по расстоянию и времени.

Таким образом, предлагается сначала выделить на треке "точки простоя" и "контрольные точки" и использовать их как точки разбиения исходной ломаной на подломаные, к каждой из которых уже применять классические алгоритмы упрощения ломаных, например, алгоритм Рамера — Дугласа — Пекера.

2. Алгоритм выделения "точек простоя"

"Точка простоя" характеризуется тем, что в течение некоторого промежутка времени, не менее τ, все координаты попали в окружность радиуса δ, а все более ранние и более поздние точки лежат на расстоянии не менее δ_2 от этой окружности. Всю группу точек, попавшую в эту окружность, мы будем заменять не одной, а двумя точками: самой ранней и самой поздней. Таким образом, мы сохраним

информацию о времени прибытия наблюдаемого объекта на стоянку и времени выезда со стоянки. Если же данная группа точек по времени попадает в диапазон τ, мы заменим данную группу точек не двумя, а одной. Таким образом, кроме выделения "точек простоя" данная часть алгоритма будет дополнительно фильтровать входные данные по принципу "ближайший сосед".

На вход алгоритма поступают геолокационные данные, которые накапливаются в буфере *parking*, пока радиус окрестности, описанной вокруг точек этого буфера, не превышает ε.

```
for point in input_stream:
    # Если все точки помещаются в нужну окрестность - пусть помещаются
    if circle(parking + point).radius < delta:
        parking.append(point)
    # если же новая точка не лезет, пора заканчивать
    else:
      # если прошло больше tau - то это стоянка, добавляем первую и последнюю
        if parking.last - parking.first.time > tau:
            output.append(parking.first)
            output.append(parking.last)
        else:    # время небольшое, сжимаем крайние точки в одну "среднюю"
            midpoint = (parking.last + parking.first) / 2
            output.append(midpoint)
        # очищаем парковку
        parking = []
        # и точку, которая не влезла в предыдущую парковку, кладем в новую
        parking.append(point)
```

Если использовать эффективный алгоритм нахождения радиуса описанной окружности, например [3], имеющий сложность $O(n)$, то в худшем случае сложность предлагаемого алгоритма будет $O(n^2)$.

Недостатком данного алгоритма является возможность разрастания временного буфера *parking*, например в случае, если отслеживаемый объект слишком долго находится на стоянке. Однако и этот недостаток легко устраняется — достаточно лишь добавлять в *parking* только те точки, которые приводят к росту окрестности.

Если пожертвовать точностью и вместо окружности использовать прямоугольную окрестность, алгоритм, очевидно, будет иметь сложность $O(n)$.

3. Реализация и апробация результатов

Алгоритм был реализован на языке *Python* и используется в составе программного комплекса "Кто куда" (ООО "Лаб М") для фильтрации геолокационных данных от аппаратных GPS/Глонасс–трекеров, установленных на автомобилях, передвигающихся по г. Самара и области и передающих геоданные в среднем 10 раз в минуту.

Данные поступали в фильтр, выделяющий "точки простоя", с выхода этого фильтра — в промежуточный буфер. К данным в промежуточном буфере применялся алгоритм Рамера — Дугласа — Пекера, для блока данных между "точками простоя" и "контрольными точками".

Данные записывались в базу данных со входа фильтра и с выхода фильтра. Эксплуатация показала не менее чем десятикратное снижение количества записываемых в базу данных геопозиций без ущерба для качества представления (при движении в городском режиме).

Литература

[1] Ramer Urs. An iterative procedure for the polygonal approximation of plane curves // Computer Graphics and Image Processing. 1972. № 1(3). P. 244–256. (DOI: 10.1016/S0146-664X(72)80017-0).

[2] Douglas David, Peucker Thomas. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature // The Canadian Cartographer. 1973. № 10(2). P. 112–122. (DOI: 10.3138/FM57-6770-U75U-7727).

[3] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). New Results and New Trends in Computer Science // Lecture Notes in Computer Science. 1991. V. 555. P. 359–370.

Поступила в редакцию 18/*X*/2013;
в окончательном варианте — 02/*XII*/2013.

A TIME-BLOCK ALGORITHM FOR FILTERGIN GEOLOCATION DATA

© 2013 N.V. Beilina[2]

In this paper a fast and simple algorithm for filtering geolocation data stream is considered.

Key words: geolocation data, stream data processing.

Paper received 18/*XI*/2013.
Paper accepted 02/*XII*/2013.

[2]Beilina Natalya Viktorovna (natalie@samdiff.ru), the Dept. of Mathematics and Informatics, Samara State Technical University, Samara, 443011, Russian Federation.