



Рис. 3 – Модуль скорости сенсора во время удара

После нахождения границ активностей полученные данные можно использовать для классификации движений, определения качества удара, вычисления необходимых метрик, например, точных скорости и перемещения. Скорость, вычисляемую с помощью фильтра высоких частот, некорректно использовать для метрик движения, так как HPF может удалять из сигнала важную информацию, однако такие данные упрощают разработку алгоритмов для определения границ активностей.

Список использованных источников:

1. PIQ – Homepage [Электронный ресурс]. – Режим доступа: <https://www.piq.com>. – Дата доступа: 23.03.2017
2. Селиванов, И. А. Применение датчиков мобильных устройств для обработки информации о состоянии окружающей среды : дис. ... маг. техн. наук : 1-40 80 05 / И. А. Селиванов. – Мн.: БГУИР, 2015. – 64 л.
3. High-pass\_filter [Электронный ресурс]. – Режим доступа: [http://www.en.wikipedia.org/wiki/High-pass\\_filter](http://www.en.wikipedia.org/wiki/High-pass_filter). – Дата доступа: 23.03.2017

## АЛГОРИТМ СКАЧКООБРАЗНОГО СЛИЯНИЯ ОТНОШЕНИЙ В РЕЛЯЦИОННЫХ БАЗАХ ДАННЫХ

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Скачихин А.А.

Бранцевич П.Ю. – канд. техн. наук, доцент

Одним из ключевых факторов, влияющих на производительность реляционных баз данных, является эффективность алгоритмов объединения отношений [1]. Рассматривается алгоритм слияния отношений, который является улучшением алгоритма слияния сортировкой, позволяющий осуществлять слияние  $N$  отношений со сложностью  $O(N_{min} * \log(N_{max}/N_{min}))$ . В работе проведён сравнительный анализ алгоритмов слияния отношений. Особое внимание уделено алгоритму скачкообразного слияния, как эффективной альтернативе существующим алгоритмам.

Алгоритм скачкообразного слияния - это алгоритм слияния отношений для запросов заданных логикой первого порядка без использования универсальных кванторов (за исключением отрицательного квантора существования). Алгоритм скачкообразного является улучшением алгоритма слияния сортировкой, которое позволяет одновременно производить слияние нескольких унарных отношений  $A_1(x) \dots A_k(x)$  за время пропорциональное размеру наименьшего из отношений. Алгоритм был впервые применён в рамках коммерческой системы, основанной на Datalog.

В качестве начальных условий, алгоритм требует, чтобы отношения  $A_i \subseteq N$  были отсортированы. Каждое из отношений представляется в виде линейного итератора со следующим интерфейсом:

key()	Возвращает ключ отношения в текущий позиции итератора
next()	Переводит итератор к следующему ключу
seek(key)	Позиционирует итератор на наименьшем ключе поле key ( $currentKey \geq key$ ) или передвигает итератор в конец, если такой ключ не существует.
hasNext()	Возвращает true если итератор достиг конца.

Требуется, чтобы все методы итератора для отношения  $A_i$  имели временную сложность не более чем  $O(\log N)$ , где  $N = |A|$ . Более того, если  $m$  ключей посещены строго по возрастанию, сложность должна составлять  $O(1 + \log_2(N/m))$ , что легко достижимо с применением сбалансированных древовидных структур данных [2]. В случае, если все ключи были посещены по возрастанию, сложность составит  $O(1)$ .

Сам алгоритм скачкообразного слияния, тоже реализован как итератор для пересечения  $A_1 \cap \dots \cap A_k$ . Алгоритм использует массив итераторов для каждого из отношений. В ходе работы, алгоритм отслеживает наибольший и наименьший ключи всех итераторов и двигает итератор, стоящий в наименьшем ключе, к наибольшему ключу, пока все итераторы не будут позиционированы на одном ключе. Если все итераторы позиционированы на одном и том же ключе, значение ключа помещается в результирующее отношение.

Временная сложность алгоритма скачкообразного слияния зависит от размера наименьшего из отношений, т.к. именно наименьшее отношение будет способствовать быстрому продвижению итераторов. В отличие от алгоритма слияния, основанного на хешировании, алгоритм скачкообразного слияния может применяться для слияния больших списков. Существуют улучшения алгоритма, которые позволяют осуществлять слияние списков для любых конъюнктивных запросов и не ограничены унарными отношениями. Все подобные алгоритмы используют алгоритм скачкообразного слияния в качестве основного строительного блока.

Преимущества алгоритма:

- 1) в определённых ситуациях алгоритм скачкообразного слияния позволяет добиться лучших результатов по сравнению с алгоритмом попарного слияния;
- 2) Время исполнения пропорционально размеру наименьшего отношения;
- 3) алгоритм хорошо подходит для слияния больших отношений.

Недостатки алгоритма:

- 1) необходимость в предварительной сортировке отношений, накладные расходы на сортировку могут быть высокими;
- 2) требует больше памяти по сравнению с другими алгоритмами слияния.

Список использованных источников:

1. Chaudhuri, S. An overview of query optimization in relational systems / S. Chaudhuri. - In Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), 1998. - pages 34–43.
2. Abiteboul, S. Foundations of Databases / S. Abiteboul, R. Hull, V. Vianu. - Addison-Wesley, 1995.

## ИСПОЛЬЗОВАНИЕ BLUETOOTH-МАЯКОВ ДЛЯ НАВИГАЦИИ В ПОМЕЩЕНИЯХ

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Соколов И.А.*

*Парамонов А.И. – канд. техн. наук, доцент*

Задача определения местоположения объектов в закрытых помещениях на сегодняшний день остается актуальной и востребованной. Это обусловлено спецификой современной архитектурой и ростом технологической оснащенности инфраструктуры помещений. Известные подходы с применением традиционных средств позиционирования, в том числе GPS, не дает удовлетворяющего результата либо недоступно. Одним из решений задачи определения местоположения в закрытых помещениях - использование маяков вместе с программным обеспечением (ПО).

iBeacon - это протокол, во основе которого лежит использование Bluetooth-маяков, относящихся к категории BLE (Bluetooth Low Energy) [1]. Маяк представляет собой простой передатчик, который с заданной периодичностью посылает сигнал с определенным в протоколе iBeacon данными. Устройство имеет размер в пределах 10 сантиметров, питание происходит от батареи.

Структура данных протокола представлена в стандартном описании, которое включает:

- а) Beaconprefix – префикс пакета данных, позволяет определить, что устройство относится к маякам;
- б) ProximityUUID – идентификатор группы маяков (16 байт). Идентификатор позволяет различать принадлежность маяков, работая с определенной группой маяков используется одинаковый идентификатор;
- в) Major – идентификатор подгруппы маяков (2 байта). Позволяет различать определенный набор маяков среди всех остальных, например: группа маяков в отделе музея;
- г) Minor – номер (2 байта), идентифицирующий маяков внутри подгруппы;
- д) TXPower – значение мощности сигнала между маяком и принимающим устройством (2 байта), на основании которого можно вычислить расстояние [2].

Сочетание рассмотренных параметров позволяет однозначно определить, какой маяк находится в зоне приема, а также расстояние до него.

На сегодняшний день рынок поставщиков оборудования (маяков) широко представлен, среди наиболее