

Раздел 1. Обзор существующих решений

В свободном доступе сети Интернет работ по распознаванию движения мобильным устройством не так много. Поиск существующих решений производился в Интернете. Помимо этого был произведен поиск приложений хотя бы частично решающих поставленную задачу, а также в магазине приложений. Действительно полезных примеров среди приложений найдено не было, однако автором приложения, рассмотренного во второй части раздела, пришлось в значительной степени заниматься анализом данных полученных с акселерометра.

§1.1. uWave

uWave – алгоритм распознавания движения разработанный ЧиЯнгом Ли, Чен Вангом и Лин Чонгом, сотрудниками Инженерного Департамента университета в Хьюстоне. Алгоритм использует только показания акселерометра, поэтому может быть реализован на любом устройстве с этим датчиком. Подход описанный в статье [1], по словам авторов алгоритма позволяет с точностью до 98% детектировать совершённый жест. Авторы выделили восемь жестов (Рис.1.) и собрали библиотеку данных воспроизведения (свыше 4000 записей) этих базовых жестов разными людьми в течение продолжительного периода.

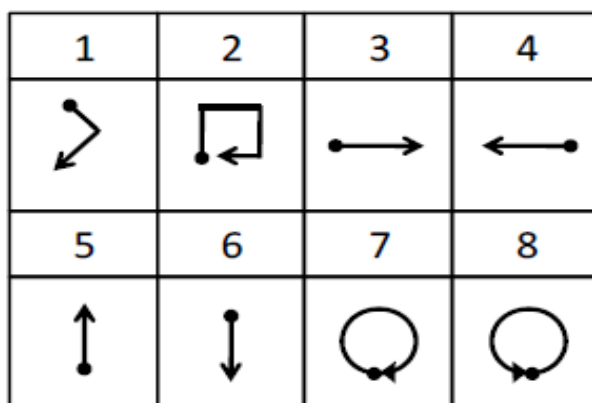


Рис. 1 Жесты, выбранные разработчиками uWave в качестве основных.

§1.1.1. Описание работы алгоритма

Работа алгоритм uWave состоит из трех этапов: квантование данных акселерометра, поиск соответствующего шаблона движения и адаптация шаблонов (Рис. 2.).

Работа алгоритма начинается с этапа квантования, в пределах которого данные акселерометра подвергаются фильтрации и квантованию. Таким образом, резко уменьшается количество обрабатываемых данных, что позволяет производить вычисления в условиях ограниченных ресурсов, как в случае с мобильным устройством. Авторы используют 32 уровня квантования, причем применена нелинейная схема распределения уровней. Так для колебаний в диапазоне от 0 до g выделено 10 уровней квантования, для диапазона от g до $2g$ выделено 5 уровней, значения большие $2g$ находятся на одном уровне. Такое распределение авторы объясняют высокой частотой возникновения значений акселерометра в диапазоне от 0 до g и очень низкой частотой появления значений в районе $2g$ и больших.

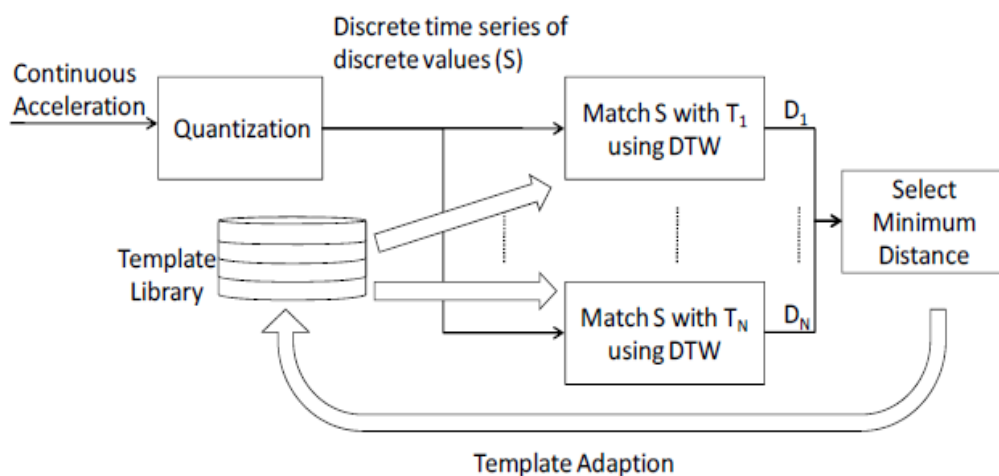


Рис. 2 Схема работы алгоритма uWave.

Ядром алгоритма uWave является другой алгоритм в английской литературе называющийся «Dynamic Time Warping» (Динамическая Трансформация Шкалы Времени, *перев. авт.*). Данный алгоритм позволяет измерить степень похожести двух последовательностей данных, которые могут отличаться друг от друга скоростью изменения данных. С помощью алгоритма DTW можно выявить схожесть между

видеозаписями, в одной из которых человек шагает с одной скоростью, а в другой – в два раза большей. DTW часто используется в распознавании речи, в обработке видео, аудио и другой информации, представимой в виде последовательности сравнимых объектов.

Алгоритм DTW в своей реализации находит расстояние Левенштейна двух последовательностей, в данном случае показателей акселерометра для эталонного жеста и показателей акселерометра для воспроизведенного жеста.

Расстояние Левенштейна находится следующим образом: Пусть S_1 и S_2 — две строки (длиной M и N соответственно) над некоторым алфавитом, расстояние Левенштейна $d(S_1, S_2)$ можно подсчитать по рекуррентной формуле

$$d(S_1, S_2) = D(M, N), \text{ где}$$

$$D(i, j) = \begin{cases} 0 & ; i = 0, j = 0 \\ i & ; j = 0, i > 0 \\ j & ; i = 0, j > 0 \\ \min(& \\ \quad D(i, j - 1) + 1, & \\ \quad D(i - 1, j) + 1, & ; j > 0, i > 0 \\ \quad D(i - 1, j - 1) + m(S_1[i], S_2[j]) & \\) & \end{cases}$$

Сложность алгоритма DTW по времени и по использованию памяти составляет $O(M*N)$, поэтому квантование данных играет серьезную роль в производительности алгоритма.

Последним этапом алгоритма является адаптация шаблонов жестов. Способ выполнения одного и того же жеста одним человеком может меняться от времени к времени – человек не может в точности повторить два движения, проведя исследование авторы алгоритма решили применить адаптацию эталонных шаблонов к способу исполнения жеста пользователем. Адаптация происходит достаточно простым способом. Для каждого эталона хранится два варианта воспроизведения, каждый новый пользовательский способ воспроизведения жеста сравнивается с двумя хранимыми вариантами и, если пользовательский жест был успешно распознан, он заменяет собой старый эталонный шаблон.

§1.2. AccelPaint

AccelPaint – мобильное приложение, интересным образом реализующее работу с акселерометром. С помощью акселерометра данное приложение позволяет рисовать. Тривиальная на первый взгляд задача, является довольно сложной в своей реализации. Действительно, имея только показания акселерометра, то есть измерений ускорения устройства, автор приложения решил, точнее, попытался решить, задачу восстановления изменений координат устройства из данных об ускорении. Рисовать в программе AccelPaint, хоть и непросто, но правила, по которым происходит рисование в зависимости от движения, очевидны. Рисующий маркер практически следует траектории, описываемой устройством во время рисования. К сожалению, автор приложения не применил фильтрации к данным акселерометра, отчего становятся заметными шумы и неточности в результирующем рисунке.

Раздел 2. Устранение погрешностей акселерометра

Показания акселерометра на мобильных устройствах подвержены достаточно сильному шуму. Зашумленность порой достигает 0.08g, вследствие этого возникает острая необходимость борьбы с шумом. Ниже будут рассмотрены несколько подходов к сглаживанию и фильтрации данных акселерометра.

В качестве данных возьмем для тестирования измерения акселерометра по одной оси



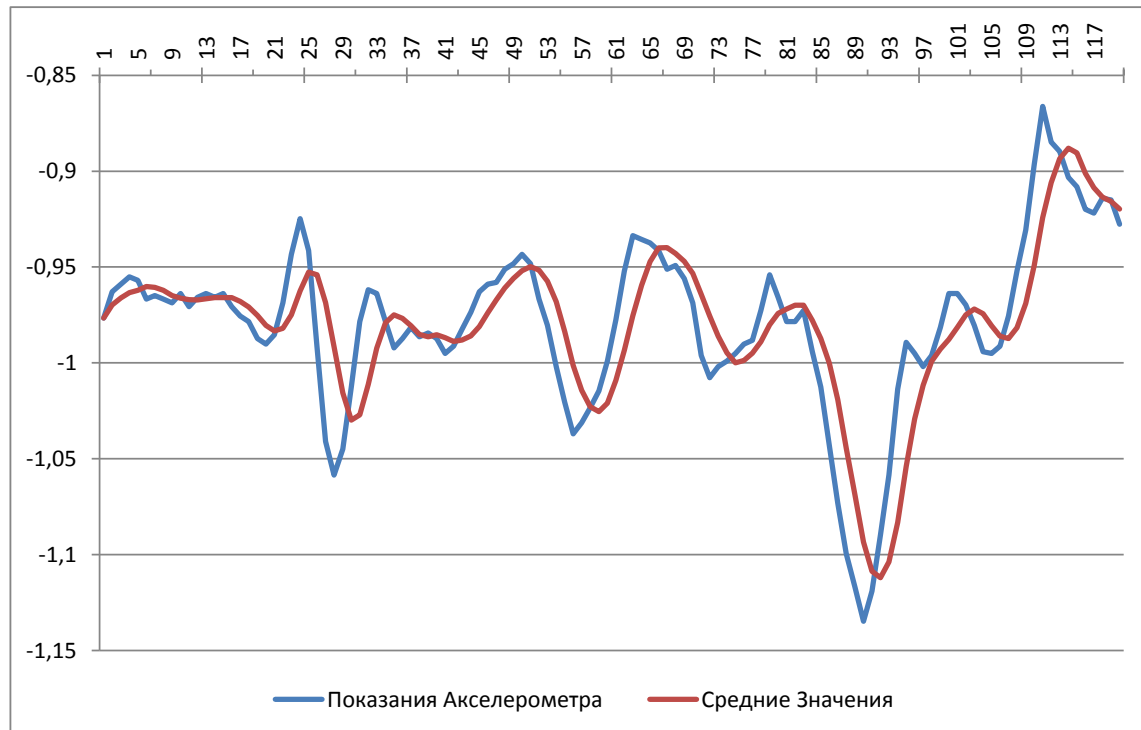
§2.1. Метод средних значений

Метод средних значений один из самых простых методов фильтрации шума, суть его такова: на каждом шаге k , значение v_k вычисляется как среднее из n предыдущих

значений акселерометра, то есть $v_k = \frac{\sum_{i=0}^n a_{k-i}}{n}$. Такой метод дает, при средних

значениях n , неплохое сглаживание, но имеет один существенный недостаток –

достаточно большую задержку в значениях.



§2.2. Фильтрация данных

Одним из способов борьбы с зашумленными данными является применение фильтра. Задача распознавания движения устройством накладывает одно существенное требование к фильтру – требование производительности достаточной для того, чтобы использовать фильтр в режиме реального времени с минимальными задержками. Конечно, большим плюсом фильтра является приближенность значений к начальным.

§2.2.1. Фильтр низких частот

Фильтры нижних частот – группа фильтров общей характеристикой которых является способность фильтровать сигналы выше указанной частоты, то есть фильтры пропускает сигналы низкой частоты, что позволяет избавиться от шумовых помех сигнала, в нашем случае показаний акселерометра.

Самый простой фильтр низких частот описывается следующей формулой:

$O_n = O_{n-1} + \alpha(I_n - O_{n-1})$, где O – выходное значение сигнала (отфильтрованное), I_n –

входные значения (нефильтрованные), α - коэффициент фильтрации, принимающий значения от 0 до 1. При α равном 1, выходные значения совпадают с входными.

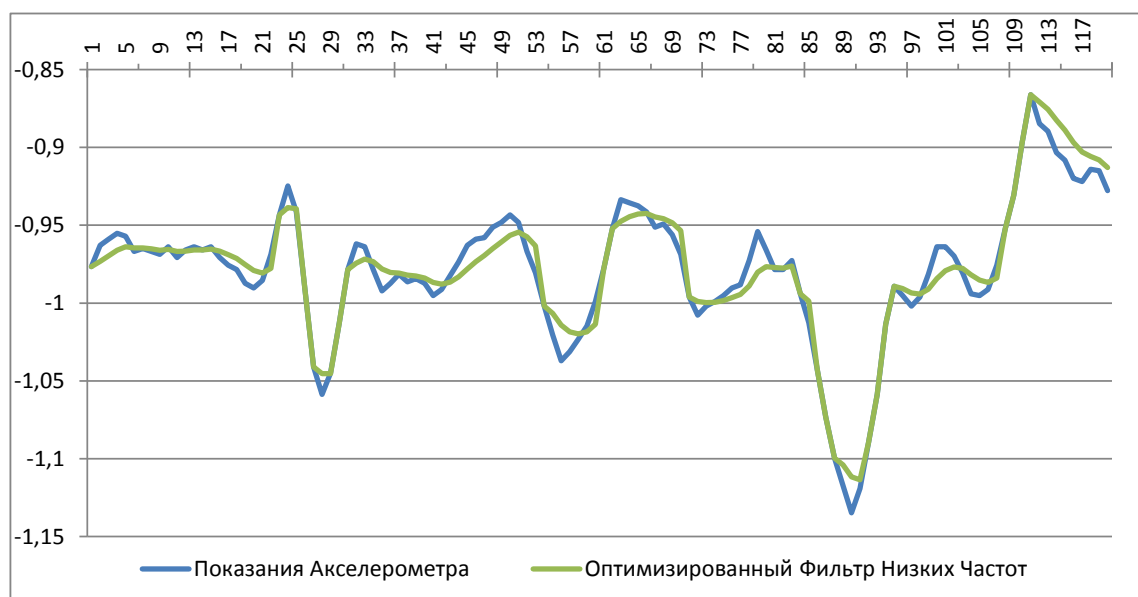
В качестве примера, ниже приводятся два варианта фильтрованных с помощью фильтра нижних частот данных с коэффициентом $\alpha = 0.5$ и $\alpha = 0.25$, соответственно.



Как видно из примеров меньший коэффициент дает более гладкий результат. Результирующий сигнал достаточно гладкий, но, так же как и при сглаживании методом средних значений, присутствует заметная задержка, особенно при резком колебании значений.

§2.2.2. Модифицированный фильтр низких частот

Для уменьшения задержки сглаживания при резких колебаниях необходимо ввести зависимость фильтра от приращения n -го и $n-1$ -го значения сигнала. Пусть ε - пороговое значение, такое что, если $|I_n - I_{n-1}| < \varepsilon$, то значение фильтруется вышеописанным фильтром низких частот, иначе – возвращается входное значение I_n .



Как видно из представленного графика – на резких перепадах, задержка сведена к минимуму, при этом сглаживание вполне неплохое.

§2.2.3. Фильтр Калмана

Фильтр Калмана часто используется для фильтрации значений различного рода сигналов, его можно встретить в GPS-приемниках, обработчиках показаний датчиков и т.д. Фильтр Калмана является разновидностью рекурсивных фильтров. Для вычисления оценки состояния системы на текущий шаг работы ему необходима оценка состояния (в виде оценки состояния системы и оценки погрешности определения этого состояния) на

предыдущем шаге работы и измерения на текущем шаге. Далее под записью $\hat{\mathbf{x}}_{n|m}$ будем понимать оценку истинного вектора \mathbf{x} в момент n с учетом измерений с момента начала работы и по момент m включительно.

Состояние фильтра задается двумя переменными:

$\hat{\mathbf{x}}_{k|k}$ — апостериорная оценка состояния объекта в момент k полученная по результатам наблюдений вплоть до момента k включительно;

$\mathbf{P}_{k|k}$ — апостериорная ковариационная матрица ошибок, задающая оценку точности полученной оценки вектора состояния и включающая в себя оценку дисперсий погрешности вычисленного состояния и ковариации, показывающие выявленные взаимосвязи между параметрами состояния системы.

Итерации фильтра Калмана делятся на две фазы: экстраполяция и коррекция. Во время экстраполяции фильтр получает предварительную оценку состояния системы $\hat{\mathbf{x}}_{k|k-1}$ на текущий шаг по итоговой оценке состояния с предыдущего шага. Эту предварительную оценку также называют априорной оценкой состояния, так как для её получения не используются наблюдения соответствующего шага. В фазе коррекции априорная экстраполяция дополняется соответствующими текущими измерениями для коррекции оценки. Скорректированная оценка также называется апостериорной оценкой состояния, либо оценкой вектора состояния $\hat{\mathbf{x}}_k$. Обычно эти две фазы чередуются: экстраполяция производится по результатам коррекции до следующего наблюдения, а коррекция производится совместно с доступными на следующем шаге наблюдениями, и т. д.

§2.2.2.1. Этап экстраполяции

Экстраполяция (предсказание) вектора состояния системы по оценке вектора состояния и примененному вектору управления с шага $(k-1)$ на шаг k :

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_{k-1}$$

Ковариационная матрица для экстраполированного вектора состояния:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T + \mathbf{Q}_{k-1}$$

§2.2.2.2. Этап коррекции

Отклонение полученного на шаге k наблюдения от наблюдения, ожидаемого при произведенной экстраполяции:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$$

Ковариационная матрица для вектора отклонения (вектора ошибки):

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k$$

Оптимальная по Калману матрица коэффициентов усиления, формирующаяся на основании ковариационных матриц имеющейся экстраполяции вектора состояния и полученных измерений (посредством ковариационной матрицы вектора отклонения):

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1}$$

Коррекция ранее полученной экстраполяции вектора состояния — получение оценки вектора состояния системы:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k$$

Расчет ковариационной матрицы оценки вектора состояния системы:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1}$$

Выражение для ковариационной матрицы оценки вектора состояния системы справедливо только при использовании приведенного оптимального вектора коэффициентов.

В случае с одной переменной матрицы вырождаются в скалярные значения. Рассмотрим значения переменных фильтра применительно к нашему случаю:

F – переменной, описывающей динамику системы присвоится значение 1, что означает, что экстраполируемое значение будет равно значению на предыдущем шаге.

B – переменная, определяющая применение управляющего воздействия, примет значение 0, ввиду отсутствия такового.

H – матрица, определяющая отношение между измерениями и состоянием системы, в случае с данными с акселерометра примет единичный вид.

На графике изображен один из вариантов фильтра Калмана примененный к данным полученным акселерометра.

