

# Распознавание движения мобильного устройства

Пестов Е. А.

**Аннотация** — Вслед за развитием рынка мобильных устройств и увеличением функциональности, увеличивается и круг задач, которые можно решить с их помощью. В статье рассмотрена задача распознавания движения мобильного устройства, то есть детектирования определённого физического перемещения устройства в пространстве. В статье приводится обзор алгоритма распознавания, фильтрация данных акселерометра, а также визуализации базовых движений.

**Ключевые слова** — акселерометр, мобильное устройство, распознавание движения, фильтрация данных.

## I. ВВЕДЕНИЕ

Распознавание движения - это большая задача компьютерной науки, которая имеет множество вариаций, в каждой из которых существует огромное количество подзадач и различных подходов к решению. Вариативность задачи зависит от контекста, в котором движение необходимо распознавать: от распознавания жеста сделанного мышью, до управления телом и руками персонажами в играх на приставке. В контексте мобильных платформ, понятие «распознавание движения» сводится, как правило, к распознаванию жестов пальцами на сенсорном экране для управления программным обеспечением. Эта задача в современных устройствах решена на уровне операционной системы, именно на ОС возложена функция распознавания базовых жестов.

Современные мобильные устройства обладают не только высокими вычислительными характеристиками, но и обилием различного рода датчиков, позволяющих узнать как различные свойства окружения, так и некоторые свойства самого устройства в этом окружении. С появлением таких датчиков появился ещё один аспект задачи «распознавания движения» - это задача по распознаванию движения самого устройства. Возможность детектировать манипуляции устройством открывает массу вариантов использования этих манипуляций. Например, становится возможным передать контактные данные другому человеку, просто совершив движение устройства по направлению к нему, или реализовать виртуальную дуэль на «волшебных палочках» (роль которых будут играть мобильные устройства), где определенные «заклинания» будут инициироваться определенным движением устройства. Словом, открывается широкий спектр интересных возможностей.

Е. А. Пестов студент магистратуры Московского Государственного Университета имени М. В. Ломоносова, факультета Вычислительной Математики и Кибернетики (e-mail: eapestov@yandex.ru).

Для определения характеристик устройства в пространстве, в большинстве случаев используются три датчика, устанавливаемых на большинство современных устройств – это акселерометр, компас и гироскоп. Последние два позволяют определить ориентацию и наклон устройства в пространстве. В задаче распознавания движения устройства, применение гироскопа и компаса ограничено, обрабатывая данные с этих датчиков можно определить изменение углов наклона устройства, но нельзя детектировать перемещение. Изменение положения устройства позволяет определить первый упомянутый датчик – акселерометр.

С развитием технологий акселерометром стали оснащаться практически все мобильные устройства. Основным назначением этого датчика является предоставление информации о текущем ускорении устройства, точнее разности ускорения устройства и ускорения свободного падения. В состоянии покоя показания датчика совпадают с вектором ускорения свободного падения.

Первое мобильное устройство с акселерометром было выпущено в 2005 году компанией Nokia, с тех пор акселерометр стал активно использоваться в устройствах других производителей. Основную популярность этот датчик получил в мобильных игровых приложениях, позволяя отказаться от неудобного, в мобильном контексте, «клавишного» управления в пользу интуитивного управления ориентацией устройства.

Как правило, способ использования данных акселерометра достаточно тривиален – снимаются показания датчика, и подсчитывается отклонение вектора ускорения от начального положения. Получая, таким образом, показания датчика с определенной частотой, можно использовать их, например, для управления автомобилем в гоночной игре. Вектор отклонился - автомобиль поворачивает в соответствующую сторону.

Более сложная обработка данных, предоставляемых акселерометром, позволяет определить и анализировать движение самого мобильного устройства в пространстве.

Под словосочетанием «движение устройства» в контексте данной работы следует понимать физическое перемещение устройства в пространстве. Под термином «жест» - заранее предопределенное движение (например, движение по кругу по часовой стрелке), которое подлежит детектированию.

В первой части работы рассмотрен один из алгоритмов распознавания движения устройства [1][2].

Во втором разделе описаны способы сглаживания и фильтрации данных, полученных с акселерометра.

В третьем разделе описаны и графически представлены показания датчика при выполнении базовых движений.

В заключении подведены итоги исследования.

## II. ОБЗОР АЛГОРИТМА 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 и собрали библиотеку данных воспроизведения (свыше 4000 записей) этих базовых жестов разными людьми в течение продолжительного периода.

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

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.

Ядром алгоритма uWave является другой алгоритм в английской литературе называющийся «Dynamic Time Warping» (Динамическая Трансформация Шкалы Времени, *перев. авт.*) [4][5]. 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 алгоритмах 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  $d(S_1, S_2) = D(M, N)$ , где

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

Сложность алгоритма DTW по времени и по использованию памяти составляет  $O(M * N)$ , поэтому правильное квантование данных является серьезной оптимизацией 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. По результатам проведенного исследования 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, заменяет собой старый эталонный шаблон.

## III. 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.

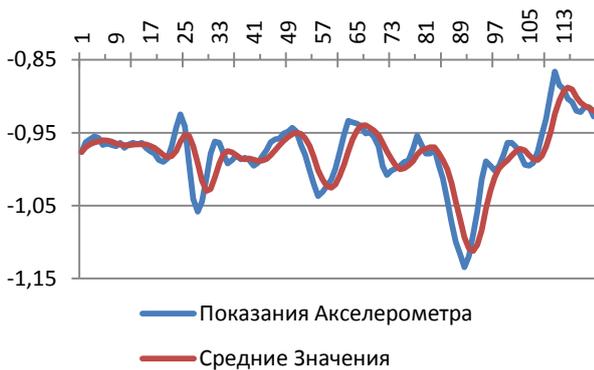
В качестве данных для тестирования взяты измерения акселерометра по одной оси.



### A. 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.



### В. 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.

#### 1) Фильтр нижних 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 нижних 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),  $\alpha$  – коэффициент фильтрации, принимающий значения от 0 до 1. При  $\alpha$  равном 1, выходные значения совпадают с входными.

В качестве примера, ниже приводятся два варианта отфильтрованных 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) Модифицированный фильтр нижних частот

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



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

#### 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  $\hat{x}_{n|m}$  будем понимать оценку истинного вектора  $X$  в момент  $n$  с учетом измерений с момента начала работы и по момент  $m$  включительно.

Состояние фильтра задается двумя переменными:

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

$P_{k|k}$  — 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  $\hat{x}_{k|k-1}$  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  $\hat{x}_k$ . 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.

#### а) Этап экстраполяции

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

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_{k-1}$$

Ковариационная матрица для экстраполированного вектора состояния:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_{k-1}$$

#### б) Этап коррекции

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

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1}$$

Ковариационная матрица для вектора отклонения (вектора ошибки):

$$S_k = H_k P_{k|k-1} H_k^T + R_k$$

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

$$K_k = P_{k|k-1} H_k^T S_k^{-1}$$

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

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

Расчет ковариационной матрицы оценки вектора состояния системы:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1}$$

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.

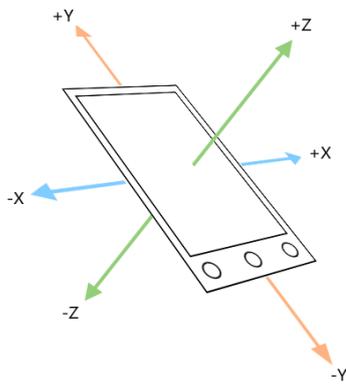
$H$  — матрица, определяющая отношение между измерениями и состоянием системы, в случае с данными с акселерометра примет единичный вид.

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



## IV. КАТЕГОРИИ ДВИЖЕНИЙ

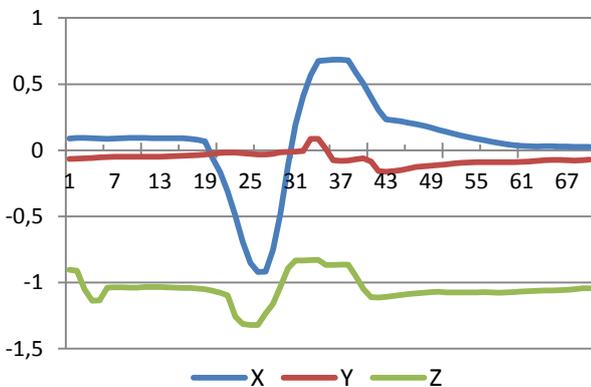
В этом разделе рассматриваются категории простых движений и их отображение данными акселерометра. В качестве устройства был взят смартфон Nokia Lumia 900, управляемый операционной системой Windows Phone 7[6]. Оси акселерометра располагаются как показано на рисунке:



**Рис. 1** Направления осей акселерометра.

*А. Движение по прямой*

Первое рассматриваемое элементарное движение – достаточно резкое движение устройством по горизонтальной прямой слева направо.

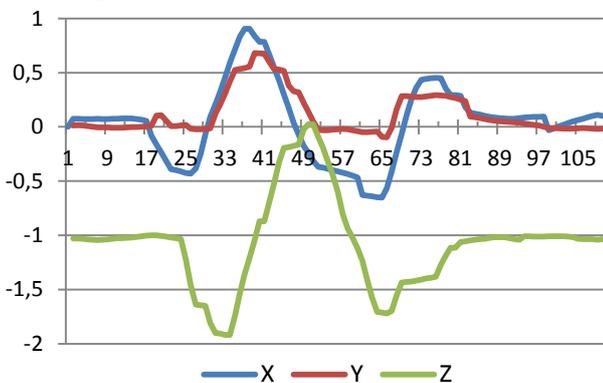


На графике ярко выделяются два всплеска по оси X, первый – начало движения, второй – конец. Значения на остальных осях существенно не меняются.

Аналогично, всплески значений на осях Y и Z образуются при движении в соответствующем направлении.

*В. Движение по окружности*

Движение по окружности проводилось против часовой стрелки, начальной точкой движения являлась нижняя точка окружности.

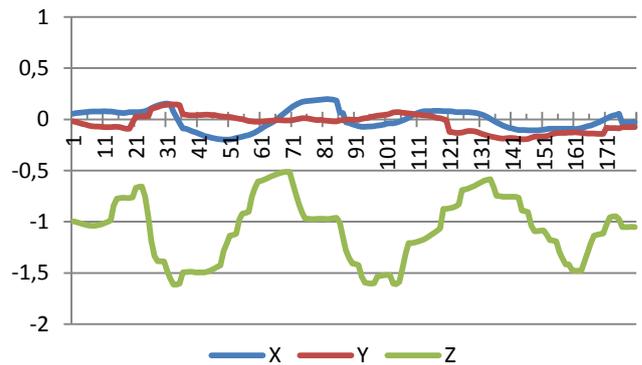


Графики изменений значений по осям X и Z практически совпадают, именно по этим осям

«рисовалась» окружность. Естественно имеет место фазовый сдвиг, так как горизонтальное изменение началось раньше, чем вертикальное. Рассмотрим подробнее график показаний акселерометра по вертикальной оси Z. График имеет синусоидальный вид, так как начало движения находилось в нижней точке окружности, где sin принимает значение -1, то ожидается увидеть полную фазу синусоиды от первой нижней точки до второй, что и можно наблюдать на графике. Спад до первой нижней точки и подъем от второй, характеризуют начало движения (разгон) и завершение (торможение).

*С. Движение «Восьмерка»*

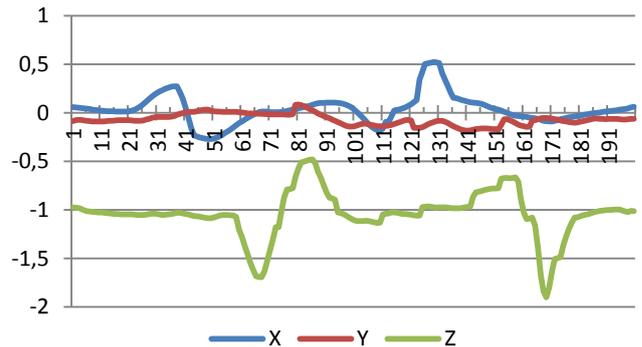
Движение проводится по горизонтальной восьмерке с началом из центра фигуры справа налево.



Ожидаемые пять длинных колебаний по вертикальной оси хорошо различимы на графике. Также вполне выделяемы две смены направления движения по горизонтали.

*Д. Движение «Квадрат»*

Движение производится по часовой стрелке, начиная с правого нижнего угла, с остановкой в каждой вершине квадрата.



Колебания по осям X и Z означают начало и остановку движения по соответствующему ребру, поэтому перепады характеризуются резкостью. Также хорошо прослеживается тот факт, что во время колебаний значений по одной оси, значения по другой оси не меняются. Действительно, во время «рисования» ребра квадрата значения меняются только по одной из осей, иначе фигура сводится к окружности.

## V. ЗАКЛЮЧЕНИЕ

В работе было проведено исследование существующего алгоритма, позволяющего частично решить задачу распознавания движения устройством с акселерометром – алгоритм *uWave*. Был выявлен существенный аспект, нерассмотренный авторами алгоритма – подавление шума датчика. Решение этой задачи позволит улучшить принятый в *uWave* подход. Также возможно усовершенствовать схему адаптации шаблонов, так как в исследуемой работе подход был описан лишь примерно, реализация же сокрыта.

Природа акселерометра в мобильных устройствах подразумевает наличие шума и погрешности измерений, поэтому в работе был проведен анализ способов сглаживания и фильтрации данных акселерометра [3]. Рассмотренные подходы оценивались по двум характеристикам – степень сглаживания измерений и скорость реакции на сильные колебания.

Кроме того, в работе был проведен анализ поведения акселерометра на примере базовых движений. Были визуализированы показания датчика [7] по трем осям измерения для каждого базового движения; также была кратко объяснена взаимосвязь между фигурой движения и показаниями акселерометра.

## БИБЛИОГРАФИЯ

- [1] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan, "*uWave: Accelerometer-based personalized gesture recognition and its applications*," in *Pervasive and Mobile Computing*, vol. 5, issue 6, pp. 657-675, December 2009.
- [2] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan, "*User evaluation of lightweight user authentication with a single tri-axis accelerometer*," in *Proc. ACM Int. Conf. Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, September 2009.
- [3] I. J. Jang and W. B. Park, "*Signal processing of the accelerometer for gesture awareness on handheld devices*," in *Proc. IEEE Int. Wkshp. Robot and Human Interactive Communication*, W. B. Park, Ed., 2003.
- [4] Donald J. Bemdt, James Clifford. "*Using Dynamic Time Warping to Find Patterns in Time Series*," Information Systems Department Stern School of Business New York University, New York, April 26, 1994.
- [5] Pavel Senin. "*Dynamic Time Warping Algorithm Review*," Information and Computer Science Department, University of Hawaii at Manoa Honolulu, USA, December 2008.
- [6] D.Edson. "*Using the Accelerometer on Windows Phone 7*", in The Windows Phone Developers Blog, [http://windowsteamblog.com/windows\\_phone/b/wpdev/archive/2010/09/08/using-the-accelerometer-on-windows-phone-7.aspx](http://windowsteamblog.com/windows_phone/b/wpdev/archive/2010/09/08/using-the-accelerometer-on-windows-phone-7.aspx)
- [7] Gabor Paller. "*Motion Recognition with Android devices*", Sfonge Ltd., <http://www.slideshare.net/paller/motion-recognition-with-android-devices>