gRmobile: A Framework for Touch and Accelerometer Gesture Recognition for Mobile Games

Mark Joselli, Esteban Clua
MediaLab, IC-UFF
{mjoselli — esteban}@ic.uff.br

***Abstract*—Mobile phone games are usually design to be able to play using the traditional number pads of the handsets. This is stressfully difficult for the user interaction and consequently for the game design. Because of that, one of the most desired features of a mobile games is the usage of few buttons as possible. Nowadays, with the evolution of the mobile phones, more types of user interaction are appearing, like touch and accelerometer input. With these features, game developers have new forms of exploring the user input, being necessary to adapt or create new kinds of game play. With mobile phones equipped with 3D accelerometers, developers can use the simple motion of the device to control the game or use complex accelerated gestures. And with mobile phones equipped with the touch feature, they can use a simple touch or a complex touch gesture recognitions. For the gesture to be recognized one can use different methods like simple brute force gestures, that only works well on simple gestures, or more complex pattern recognition techniques like hidden Markov fields, fuzzy logic and neural networks. This work presents a novel framework for touch/accelerometer gesture recognition that uses hidden Markov model for recognition of the gestures. This framework can also be used for the development of mobile application with the use of gestures.**

***Index Terms*—Mobile Games, Gesture Recognition, Motion Sensors, Touch Phones,Tangible User Interfaces.**

I. Introduction

Digital games are defined as real-time multimedia applications that have time constraints to run their tasks. If the game is not able to execute its processing under some time threshold, it will fail [1]. Mobile games are also real-time multimedia application that runs on mobile phones that have time constraints and many others constraints [2], when compared to PC or console games, like: hardware constraints (processing power and screen size); user input, (buttons, voice, touch screen and accelerometers); and different operating systems, like Android, iPhone OS, Symbian and Windows Mobile. This makes streamilly difficult for the design and development of mobile games.

On the other hand, mobile games can have unique characteristics, making unique type of games: location based games [3], [4], voice based games [5], accelerometer based games [6], camera based games [7] and touch based games [8]. In order to develop good mobile games, they must be design to take advantages of such unique characteristics into gameplay [9].

Mobile game phones are a growing market [10] and in 2010 the sales of smartphones is expected to suppress the laptops sales [11]. More than 10 million of people worldwide play games on mobile phones and handheld devices [12] and the worldwide mobile gaming revenue is expected to reach $9.6 billion by 2011 [13]. These are important motivations for game developers and designer to create blockbusters games.

One special characteristic of most mobile phones is that the user interaction is made mostly through number input [6], [14]. Because of that, the design of games must deal with this fact and design the game to use as few buttons as possible, like just one button games [15] and no-buttons at all games [14].

The evolution of mobile phones increase the processing power of such devices and also new forms of input, like touch screen devices and devices equipped with accelerometers. With the development of touch phones, like Motorola a1200, Htc Diamond, Sony Ericson w960i, Samsung Ultra

978-0-7695-3963-8/09 $26.00 © 2009 IEEE
DOI 10.1109/SBGAMES.2009.24

Smart F520 and Nokia N810, new forms of user interaction has appeared through the use of the finger or pen. This innovation has led to change the way users interact with the operation system and with games.

With the popularization of the use of accelerometer by the Nintendo Wiimote [16], the major mobile phone manufactures had also equipped their hardware with accelerometer, like Nokia N95, Sony ericson F305, Samsung Omnia and Motorola W7, among others. But this new form of user interaction has not led to major change on the interaction. This is mostly because programs/games only uses the accelerometer data as orientation.

The iPhone was one of the first devices that is equipped with touch screen and accelerometer that has mostly of the user input made though touch or motion, soon others companies followed this tendency, like: RIM Blackberry Storm, Nokia 5800, LG Arena, and many others. They basically use touch for user interaction, and the use of the accelerometer data is restricted for orientation, just like the others phones equipped with accelerometer. This paper tries to fulfill a gap on user interaction by providing a framework for gesture recognition though touch input or motion input, that can be used for games or programs.

The gesture recognition is a type of pattern recognition and can be made by different ways like: brute force [17], fuzzy logic [18], Gabor wavelet transform [19], hidden Markov model [20], Support Vector Machine [21] and neural networks [22], [23]. This work has developed a framework that can be used for gesture recognition using hidden Makov model. In order to generate and recognize the gestures database the proposed framework is divided in two parts: one for database constriction and another for the gesture recognition.

Summarizing, this work provides the following contributions:

- A novel architecture for touch/motion gesture recognition on mobile phones;
- Presentation of performance and tests of the framework showing that it can be used in real-time;
- Recognition test showing a high accuracy rate;

The paper is organized as follows: Section 2 presents some related works on the mobile development and gesture recognition on devices equipped with touch screen and devices equipped with accelerometers. Section 3 presents and explain the gesture recognition framework. Section 4 present and discuss some results of the use of the framework. Section 5 presents the conclusions and future works.

II. Related Work

Since the gRmobile has two kinds of gestures recognition (thought touch input or accelerometer motion input), this section is divided in two subsections: one for touch screen devices were user interaction and gesture recognition for this kind of device works; and another for accelerometer devices presenting user interaction works and gesture recognition approaches.

This work does not cover the related work on mobile game development. For this purpose the authors suggest the works [24], [2] which covers state of the art for this topic.

A. *Touch Devices*

Nowadays, more and more devices are coming with touch screen, and most of this is because of the decreased in the respective price [25]. Touch screen phone devices has the characteristics of having very few buttons and most of its users input interfaces are made through touch by finger or pen. For example the Blackberry Storm has about only 8 buttons and almost all of its user interaction is made by touch.

Most touch screen devices can have two kinds of input: dragged and pressed. The first is used when the user touches softly and can be used as a mouse being dragged. The second is when the user press hard on the device, and can be used as a mouse buttons pressed. Also modern devices has multi-touch screen devices like iPhone, Android T-G1 and Blackberry Storm, among others.

Some of these devices have used some of this types of input as gestures to enable friendly user interaction: like dragging for changing the web page, zooming options on photo view and many others features. But in third-party mobile software and games this use is very restrict.

Narayanaswamy et al [26] shows an implementation of handwrite recognition on PDAs using Hidden Markov Model. Wei et al. [27] presents a study about using pen gestures instead of buttons in a mobile FPS game. They showed that users have little preference in using buttons over gestures and sometimes prefer the use of gestures. The gRmobile also could be used for handwrite recognition and gesture for games but a gesture database must be constructed in order to have the this functionality in the framework (in the case of handwrite recognition all the alphabet must be in the database).

Since the touch screens have similar behavior as the mouse, mouse gesture recognizer [28], [29], [30] could be adapted in order to be used by touch screen devices. But this could be very hard since they are not adapted for the low processing power and memory constraints of such devices. The gRmobile is very adapted to such devices having a very good performance as will be showed in the performance evaluation section.

Even tough touch devices enables much more freedom when compared with buttons based phones, the input error by those devices are higher, as shows by Hoggan et al. [31] using keyboard input tests.

B. Accelerometer Devices

Accelerometer devices are getting more and more popular. This allows the interaction though the form of gestures recognition, being the Nintendo Wii game console the most prominent example of this new form of interaction. This approach allows users to become more engaged to video games [32], whose experience is not only affected by button pressing and timing but also by movement. Also Sony's Playstation 3 has a controller that is equipped with accelerometers.

Nowadays, most smartphones, and also some mobile phones, come equipped with accelerometer. This allows the use of motion and gestures as user input, but very little has been done in the field. Most mobile operating systems [11] only uses the motion to choose the screen orientation. And also most games only uses the "tilt" of the screen and not gestures as input like the works [14], [6], [33].

There are many relevant work related to gestures recognition for accelerometer devices. With the usage of the Wiimote can be highlighted the works [34], [35]. These works use Hidden Markov Models (HMM) as the recognition algorithm and the [34] presents a lower recognition of 66 % and [35] shows a lower recognition rate of 84 %. This work also uses HMM as the recognition algorithm, but it is adapted for the lower processing power of mobile devices.

Accelerometer gesture recognition requires an intensive task to be achieved on a mobile phone. The work by Choi et al. [36] presents a Bayesian network algorithm with its computation performed in the PC to recognize numbers written on the air by accelerometer mobile phones with an average recognition rate of 97 %. Also [37] that shows a HMM recognition algorithm also implemented on the PC with the gestures done by mobile phones with a recognition rate of more that 99 %. There are also some work in development like [21], [38] shows two recognition algorithms, a HMM and a Support Vector Machine, which can have an average recognition rate of 96 %.

Without the use of the PC, MobiToss [39] presents the use of mobile phone to use simple gestures to interact with large public displays with all the processing made on the phone.

III. Framework Overview

The framework is build using Java language. The choice to use Java was to achieve a higher number of devices with the same framework, allowing its usage by any accelerated/touch device that can handle the acceleration data and/or touch data, and that has a Java virtual machine, like many devices from different manufacturers: Blackberries, Nokias, Sony Ericson, Motorola, Android, LG and Samsung. Also this framework can be used in a PC with a touch device like Microsoft surface or an accelerometer device like the Wiimote.

Gesture recognition with touch/accelerated devices are represented by their patterns of the input data. The recognition is made by comparing the input pattern with the database pattern, checking if they match. In order to extract the pattern from the input data stream and comparing with the database pattern, this data must be prepared and analyzed. This work uses Hidden Markov Models in order to fulfill that need. In order to build a database, the framework needs to train and saves a set of gestures, which also used the mobile phone for this need.

The proposed framework has the following steps:

- Segmentation: is used to determine when the gesture begins and when it ends;
- Filtering: is used in order to eliminate some parts of the data stream that do not contribute to the gesture;
- Quantitizer: is used to approximate the stream of input data into a smaller set of values;
- Model: is used to compute likelihood of analyzed gestures.
- Classifier: is used in order to identify the input gesture accordingly to the database.

All the steps from the mobile phone to the user feedback are illustrated in figure 1. It is possible to notice that the framework has two distinct modes: one for gesture training, i.e, build the database, and one for gesture recognition, i.e, comparing the input gesture with the database.

A. Segmentation

Segmentation is used mainly to automatic determinate the begin and end of the gesture. For identifying the begin and end of touch gestures is very easy since the begin of the gesture is when the user first touch the screen and it ends when the user release the screen. For the accelerometer gestures the segmentation is more difficult since the data is continues in a frequency normally that varies from 20 Hz to 80 Hz. The data that comes from the accelerometer is 3 floats representing the acceleration in three axis, as figure 2 illustrate, that ranges from -3G to 3G (G meaning the gravity).

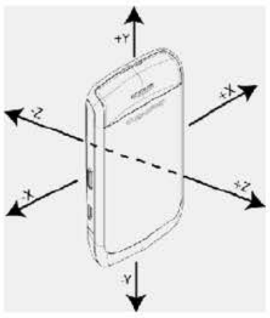

Fig. 2. The axis accelerometer

There are some accelerometer gesture recognition systems, like [35], that does the segmentation by using a button based segmentation, i.e, the user needs to press a button in order to sign the beginning and the end of the gesture. This seems like an easy approach, since it avoids computations to determine the begin and the end of a gesture like automatic segmentation must have. But on the other hand, the interaction is worst than no-button segmentation. In a mobile phones sometimes the pressing of buttons in a game is normally hard since it was designed for number dialing. This comes as another reason why this work has decided to do a no-button segmentation. This work does a similar approach as the works [40], [21].

In order to correct segmentate an accelerometer gesture, a definition of this kind of gesture is needed. This definition is made during the observation of the accelerated data and the movement of user during the recognition of different gestures. Normally gestures begins with a fast acceleration, a continuous direction change during the gesture, and it ends with a stop of the movement. In this work, the authors have observed that normally, a good gesture needs a duration of more than 0.6 seconds and less than 2 seconds.

To correct segmentate the accelerometer gesture based on the definitions some preprocessing of the accelerated data is needed. This work uses a simple mechanism. It checks the size of a vector made by the sums of the derivative, which is the difference between the axis float and the last axis float.

$$D = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2 + (z_k - z_{k-1})^2} \quad (1)$$

If the D value is bigger than 0.3, which was chosen during an extensively empirical study of gestures, the segmentation begins. And if the gesture is happening and this values drops to bellow 0.1 the gesture ends, i.e, it is assumed that the accelerometer device is in an idle position.

B. Filtering

This pass is used in order to eliminate some parts of the data stream that do not contribute to the gesture. This work uses two kinds of filters in order to eliminate noises and data that are very similar.

When a gesture is made, the data stream of the gesture may contain errors that if are sent to the HMM, some errors on the recognition may occur. In order to avoid such errors, a low pass filter is

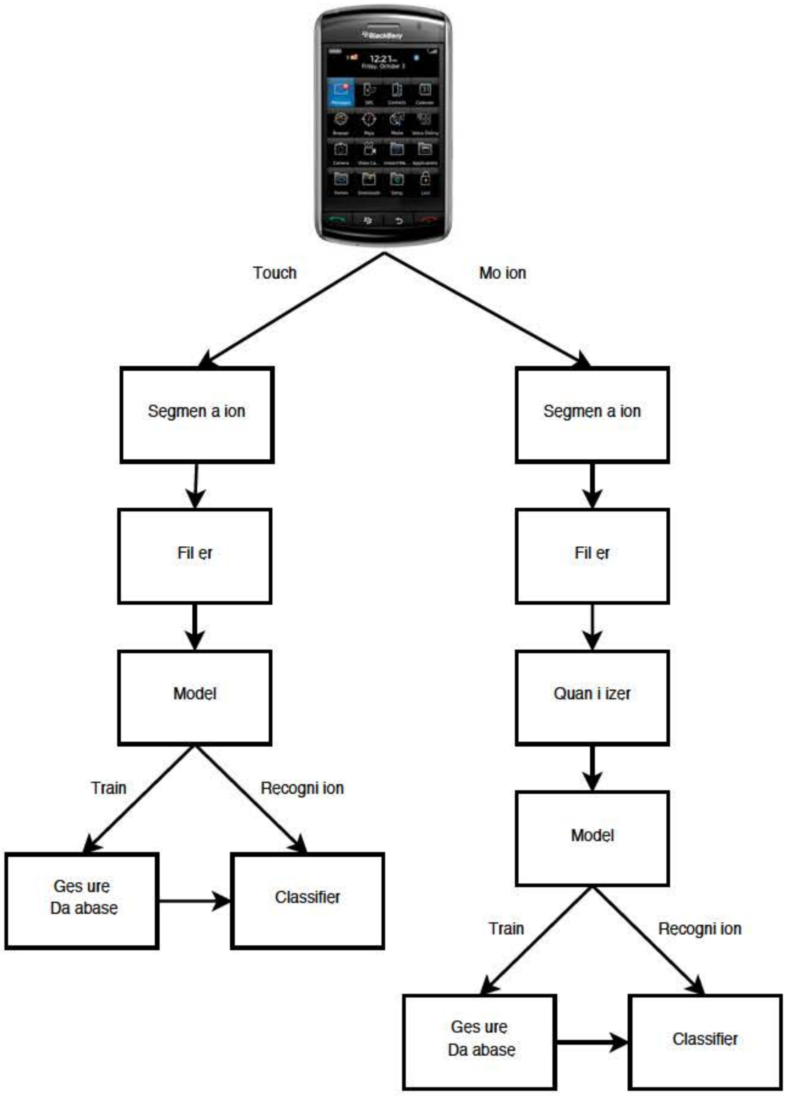

Fig. 1. System block diagram with major components of the framework.

applied which is a very common filter used for noise remove.

When the gesture is made, there are a lot of data on the data stream that does not contribute to the overall characteristic of the gesture. In order to diminished the data passed to the HMM, this work uses an idle threshold filter. This uses the same equation 1, and if D value is less than 0.2, it is not included in the gesture data stream.

C. *Quantitizer*

This step is only used for accelerated gestures. Because the accelerometer continues sends its data to the processor, the amount of data may be to big to for handling into a single HMM. Also, since the amount of RAM memory of mobile phones are not so big, the use of a quantitizer keeps less information in the gesture database. This work uses a k-mean algorithm which is a method of cluster analyzer. The algorithm aims to partition n observation

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase® Smarter legal research.