Towards Accurate Automatic Segmentation of IMU-Tracked Motion Gestures

Sven Kratz
FX Palo Alto Laboratory
3174 Porter Drive
Palo Alto, CA 94304, USA
kratz@fxpal.com

Maribeth Back
FX Palo Alto Laboratory
3174 Porter Drive
Palo Alto, CA 94304, USA
back@fxpal.com

Abstract

We present our ongoing research on automatic segmentation of motion gestures tracked by IMUs. We postulate that by recognizing gesture execution phases from motion data that we may be able to auto-delimit user gesture entries. We demonstrate that machine learning classifiers can be trained to recognize three distinct phases of gesture entry: the start, middle and end of a gesture motion. We further demonstrate that this type of classification can be done at the level of individual gestures. Furthermore, we describe how we captured a new data set for data exploration and discuss a tool we developed to allow manual annotations of gesture phase information. Initial results we obtained using the new data set annotated with our tool show a precision of 0.95 for recognition of the gesture phase and a precision of 0.93 for simultaneous recognition of the gesture phase and the gesture type.

Author Keywords

gesture segmentation; gesture annotation; gesture recognition; IMU; motion gestures; motion data

ACM Classification Keywords

H.5.2 [Input devices and strategies (e.g., mouse, touchscreen)].

Introduction and Related Work

At present, almost all smartphones as well as certain smart watches [8] are equipped with highly-sensitive inertial measurement units (IMUs). These sensors usually incorporate an accelerometer, a gyroscope and a magnetometer. This triplet of sensors allows the device to provide information about its current acceleration, rate of rotation and absolute orientation. Motion information from a device's IMU can not only be used for simple mode switching (such as changing the device's screen orientation from landscape to portrait), but also to recognize complex motion gestures [3, 5, 7].

One significant issue facing designers of user interfaces with motion gesture recognition is gesture segmentation, i.e., the segmentation of the sensor's data stream into non-gesture and gesture segments. To solve this issue, previous approaches [3, 5, 4] have mostly relied on a *push-to-gesture* mechanism, i.e., requiring some sort of user action to delimit the start and end of an input gesture. Another approach [6] used a predefined *delimiter gesture* to mark the start of a subsequent input gesture. We believe that the usability of these methods suffers due to the additional delimitation steps, and that truly fluid gestural interaction via motion gestures can only be accomplished when gestures are delimited automatically.

Ashbrook suggests using a threshold on the variance of the last N samples as decision criterium for gesture segmentation [1]. We believe that this approach has two drawbacks: (1) although we might be able to segment gesture data from inactivity, we have no further information about the gesture the user is inputing (we believe that the gesture ID can be detected with some certainty from the start phase of a gesture), and (2), this method might not perform well when deployed in the field and used in "noisy" environments where the user is subjected to motion, e.g., when using public transport.

In this abstract, we present our ongoing work on improving automatic gesture segmentation. We approach this problem by analyzing the execution phases of motion gestures more closely: instead of simply generating machine learning classifiers for entire motion gestures, we analyze the *start*, *middle* and *end* phases of gesture execution. Our preliminary results indicate that reliable classifiers can be trained for these phases of gesture execution.

In the following, we briefly outline some initial explorations we conducted into the analysis of gesture execution phases, describe a new motion data gathering experiment, a tool created to generate manual gesture phase annotations for training classifiers and present initial results of gesture phase recognition accuracy.

Exploratory Analysis of Gesture Segmentation using Machine Learning

To test the initial assumption that the gesture execution phases that we previously described can be recognized by a classifier, we analyzed a motion gesture data set that was used in a previous work [2]. When gathering this data set, we asked 25 test users to perform at least 20 repetitions of 6 different motion gestures. The data (temporal sequences of 3-axis acceleration, rotation rate and absolute orientation) was gathered on an iPhone 4 at a frequency of 100 Hz. The motion gestures were manually segmented using a *push-to-gesture* button, delimiting the start (pushing the button and holding it) and end of a gesture (releasing the button). In this way, we recorded a total of 3507 gesture entries.

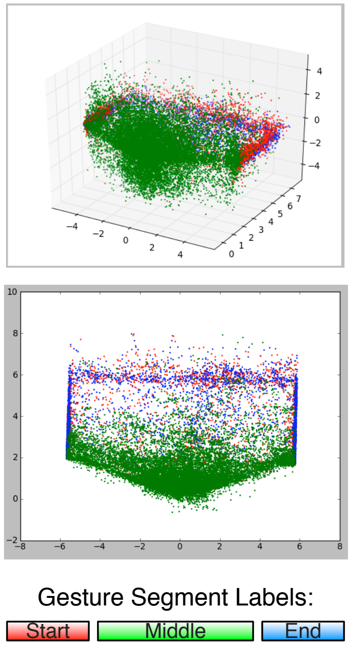

Figure 1: 2D and 3D PCA plots of the segmented gesture data. The plots provide a visual indication that the *start* and *end* gesture segment classes can be separated by a classifier from *middle*.

As we only recorded motion data when the *push-to-gesture* button was pushed and held, we did not know the exact length of the proposed gesture *start*, *middle* and *end* phases. For this reason, we conducted an exploratory statistical analysis on the number of data samples per gesture entry in our data set (Table 1):

average	median	st. dev.	max	min
286	266	109	1766	24

Table 1: A set of exploratory statistics on the number of data samples per gesture entry.

Using the empirical data on gesture sample lengths we computed the phase lengths for segmenting the gesture into the three proposed phases using a value L_{cutoff}, with:

$$L_{\text{cutoff}} = \mu - 2\sigma \quad (1)$$

Figure 2 shows how we used L_{cutoff} to segment the data for a complete gesture into three phases. We also used L_{cutoff} to filter out gestures that are unusually short, i.e., we considered only gestures with a length of L_{cutoff} or greater to build a segmentation model.

Segmenting the gesture data like this, we obtain three labelled data classes, *start*, *middle* and *end*. To obtain an indication that the data would be separable, we conducted a Principle Component Analysis (PCA) on the labeled data. Figure 1 shows 2D and 3D plots of the PCA with data dimensionality reduced to two and three dimensions, respectively. Visually, Figure 1 indicates the possibility of separating *start* and *end* from *middle*, as those points are spatially well separated. At this projection dimensionality, there appears, however, to be a substantial intermixing of *start* and *end* (red and blue dots), which may pose a problem for a classifier.

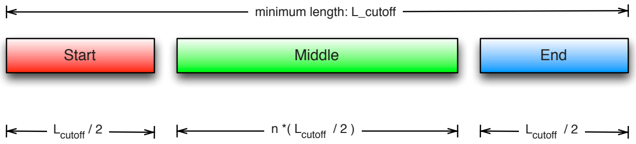

Figure 2: The proposed segmentation strategy for our existing data set is to build a classifier for the start, middle and end phases of a motion gesture. We used empirical measurements of gesture lengths to determine L_{cutoff} in order to set the correct number of sample points for each of the gesture phases.

To obtain an initial classification result, we trained a Support Vector Machine (SVM) classifier with a Radial Basis Function (RBF) kernel with the segmented and labeled data. Using 6-fold cross validation, the average accuracy for classifying the gesture phases was 88.1%.

Discussion of the Preliminary Results

Our preliminary results show that it is indeed possible to train a classifier to recognize distinct phases of gesture entry. However our initial approach has the following shortcomings:

1. The data set used in the preliminary study only includes motion samples between pressing and releasing the *push-to-gesture* button. It may, however, be of interest to also consider motion samples shortly before and after gesture execution, as these could contain important motion information that could be improve the classification accuracy of the gesture phases.
2. Since we also want to delimit gestures from other motion, we will need to train a classifier for a further label, *noise*. This will allow us to delimit a legitimate gesture entry from uninteresting motions of the mobile device.

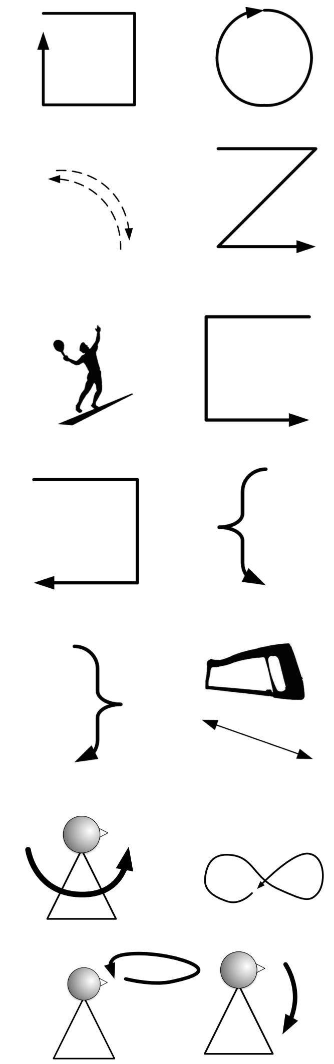

Figure 3: The 14 gestures of the newly recorded gesture data set.

Recording of an Improved Gesture Data Set

To address the issues mentioned previously, we conducted a new gesture sampling round to record an improved data set. Our goal was to capture the complete data stream from each user session, so that "noise" samples would be included. In addition to gestures manually delimited with a *push-to-gesture* button, we also captured free-form gestures from the users without any active delimitation.

For the gesture recording we recruited a total of 10 participants (3 female, 1 left-handed), all staff members of an industrial research lab. The majority of the participants were between 30 and 45 years old. On average, the participants reported that they moderate experience (2.7 on a 5-point Likert Scale, 5=*very experienced*) with motion gestures.

For the data set, we captured a set of 14 different gestures with a minimum of 15 repetitions per gesture for each user. The gesture set was chosen from gestures used previously in the literature [2, 3, 4, 7]. Figure 3 shows an overview of the gestures we used. Users were shown these images as a guide for performing each gesture type.

We used a Bluetooth YEI 3-Space IMU [9] to record the gesture data. We recorded the following 3-axis motion information at a rate of 120 Hz: absolute orientation, rotation rate, acceleration and "linearized acceleration".[1]

To support manual segmentation of the motion data into distinct gesture phases, we also captured all user gesture entries on video from the front and from the side of the user using a pair of cameras. We wrote a short script to juxtapose the two video views and record them to a single video file for each user and gesture type. The users found it easy to translate the graphical gesture representations of the gestures to motions—after data sampling was completed, they rated the ease of this process with an average of 4.4 on a 5-point Likert Scale (5=*very easy*).

Gesture Phase Annotation Tool

Because one half of the recorded gesture data purposely does not contain any delimiter events, we developed a custom tool application that allows us an annotator to manually edit the gesture data and add labels for *noise* and the gesture *start*, *mid* and *end* phases.

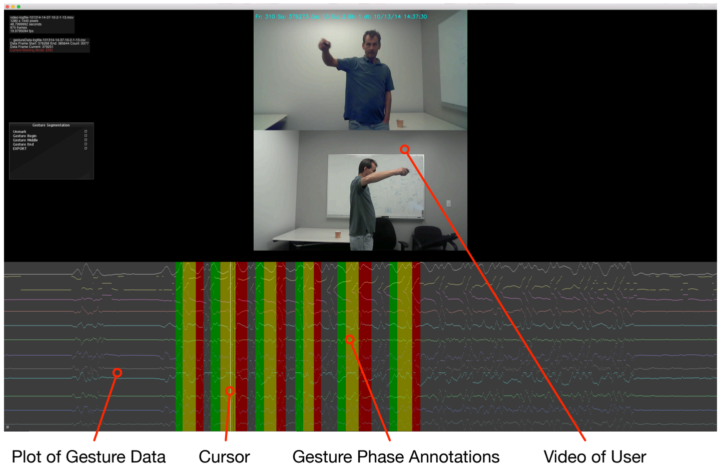

Figure 4: A screenshot of the tool used to manually annotate gesture phase information.

For each recording of a gesture type, the tool displays a plot the 12 captured data values as well as the video of the gesture recording. The tool displays a cursor that is that is synchronized to the playing video, such that the position of the cursor on the gesture data plot corresponds to the current displayed video frame. The

[1] This is the orientation-compensated acceleration of the device, with the gravity acceleration component removed [9].

tool allows the annotator to play back the video at full frame rate or, for finer analysis, step through the video frame by frame. Furthermore, the annotator can use the tool to mark portions of the gesture data with the corresponding phase labels. After annotation is completed, the tool outputs an annotated data set that contains the additional gesture phase labels.

Initial Results with New Dataset

Our experience shows us that manual gesture phase annotation is a relatively time-consuming process, with about 10 minutes needed to annotate the 15 gesture entries per gesture type per each user. Thus we estimate that to completely annotate the entire 10 user data set a proficient annotator will need about 23 h. An additional 23 h will also be needed if, for consistency, we wish to also manually annotate the part data where the *push-to-gesture* button has been used.

The process of annotating the data set is ongoing and, for the purposes of this work-in-progress, we used our tool to annotate the gesture entries of the first three users. In the following, present some initial machine learning results.

Gesture Data Preprocessing and Sampling

Our initial results indicate that it is possible to train a classifier to distinguish between the start, middle and end phases of gesture entries. This previous classifier was trained generally over all different gesture type. For our manually-annotated gesture data, we trained classifiers for each phase of each gesture. Thus, we intended to find out if we can classify the gesture type just by looking at the start phase, for instance.

To enable classifier training using the WEKA toolkit[2], we preprocessed the gestures as follows:

All gesture data was mean-shifted and normalized to a $[-1, 1]$ interval. Then, to obtain feature vectors of homogenous length we subsampled each marked gesture phase segment to contain 10 samples of the 12 data points provided by the sensor. Thus each feature vector has a length of 120. We used linear interpolation to subsample the data evenly from the source annotations.

We trained a multi-class SVM classifier (with a Radial Basis Function kernel, $C = 4.0$ and $\gamma = 0.5$) on training data labeled with the gesture ID and gesture phase (i.e., a total of 42 different classes). Our results using 10-fold cross validation show an average precision of 0.93 (and F1 score = 0.921) with a minimum precision of 0.78 and a maximum accuracy of 1.0. This results indicate that we can not only classify the *type* of gesture being performed but also the current *phase* of gesture performance.

For comparison with our first data set, we also trained a multi-class SVM classifier (again with a Radial Basis Function kernel, $C = 4.0$ and $\gamma = 0.5$) on the training data labeled just with the gesture phase for each gesture entry (i.e., three different classes). Using 10-fold cross-validation we obtained an average accuracy of 0.95 (and F2 value of 0.95). This result provides us with a verification that our new data set is on par or better than our initial data set. We also get an indication that manual segmentation, taking part of the lead-in and lead out of a gesture into consideration, yields improved recognition accuracy for gesture-phase classification.

[2] http://www.cs.waikato.ac.nz/ml/weka/

Discussion and Future Work

Our current results look promising: it is appears to be possible not only to recognize the gesture execution phase (start, middle and end of a gesture), but our results also show that it is possible to simultaneously classify what type of gesture is being performed (i.e., simultaneous classification of gesture ID and phase). Apart from using this information for auto-delimitation, we believe the classification of gesture phase could be used to train more accurate gesture classifiers, for instance it could be imaginable to use a hierarchy of classifiers with classifiers first detecting the start and (possible) ID of a gesture, and then using a more specialized classifier to verify the gesture ID. More expressive, multi-part gestures, e.g., with multiple repetitions of a certain segment are also a possibility.

One aspect of the new data set that we did not factor into the current work is the *noise* label. On our agenda for future work we intend to use classifiers more suitable to dynamic time-based data, such as Hidden Markov Models (HMMs). HMMs can be trained with a *garbage* state to recognize the data labeled as *noise*. Ultimately, we wish to develop and evaluate a live demonstrator for automatic gesture segmentation. In this case, HMMs will be more suitable to process a live stream of sensor data to generate live gesture recognition events.

References

[1] Ashbrook, D. *Enabling Mobile Microinteractions*. PhD thesis, Georgia Institute of Technology, 2010.

[2] Guse, D. Gesture-Based User Authentication on Mobile Devices using Accelerometer and Gyroscope (Master's Thesis). *Quality and Usability Group, Deutsche Telekom Laboratories, TU Berlin* (2011).

[3] Hoffman, M., and Varcholik, P. Breaking the status quo: Improving 3d gesture recognition with spatially convenient input devices. In *IEEE Virtual Reality Conference (VR)*, IEEE (Waltham, Massachusetts, USA, 2010), 59–66.

[4] Kratz, S., and Rohs, M. A $3 Gesture Recognizer - Simple Gesture Recognition for Devices Equipped with 3D Acceleration Sensors. In *Proceedings of the 14th international conference on Intelligent user interfaces.* (Hong Kong, China, Feb. 2010).

[5] Kuehnel, C., Westermann, T., Hemmert, F., Kratz, S., Müller, A., and Moeller, S. I'm home: defining and evaluating a gesture set for smar-home control. *International Journal of Human-Computer Studies 69*, 11 (2011), 1071–5819.

[6] Ruiz, J., and Li, Y. DoubleFlip: a motion gesture delimiter for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, ACM (2011), 2717–2720.

[7] Schloemer, T., Poppinga, B., Henze, N., and Boll, S. Gesture recognition with a Wii controller. In *Proc. TEI '08*, ACM (New York, NY, USA, 2008), 11–14.

[8] Wikipedia Editors. Apple Watch (retrieved 01/05/2015). `http://en.wikipedia.org/wiki/Apple_Watch`.

[9] YEI Technology. *3-Space Sensor User's Manual*. `http://www.yeitechnology.com/sites/default/files/YEI_TSS_Users_Manual_3.0_r1_4Nov2014.pdf`, 2014.