WristRotate – A Personalized Motion Gesture Delimiter for Wrist-Worn Devices

Frederic Kerber
German Research Center for Artifical Intelligence (DFKI)
Saarbrücken, Germany
frederic.kerber@dfki.de

Philipp Schardt
German Research Center for Artifical Intelligence (DFKI)
Saarbrücken, Germany
philipp.schardt@dfki.de

Markus Löchtefeld
German Research Center for Artifical Intelligence (DFKI)
Saarbrücken, Germany
markus.loechtefeld@dfki.de

ABSTRACT
In this paper, we present *WristRotate*, a personalized motion gesture delimiter that enables separation of non-relevant motion from gesture input. In an extensive data collection, we acquired 435.1 hours of smartwatch acceleration data during everyday usage. We implemented a gesture recognition system based on Dynamic Time Warping to partition a stream of accelerometer readings to identify possible gestures and to classify them accordingly. Through our analysis, we were able to identify a gesture that is (1) uncommon in daily life; (2) quick and easy to execute and (3) easily and reliably detectable. The gesture is executed by simply rotating the lower arm and wrist outwards and back inwards (twice).

Author Keywords
Smartwatch; Gestural Interaction; Same-Side Interaction

CCS Concepts
•**Human-centered computing → Gestural input;**

INTRODUCTION
With the current rise of smartwatches and fitness tracking armbands, an increasing number of people are wearing an always-listening collection of motion sensors on their wrists. While today's smartwatches often allow for sophisticated input, these are not always appropriate (e.g. speech input) or possible (e.g. touch input requires the opposite hand). Fitness tracking bands such as the Fitbit[1], on the other hand, often do not have such sophisticated input techniques. Still, those devices can be used for input, as they often have a permanent connection to the user's smartphone. We envision a gesture based interaction design that allows for inconspicuous one-handed interaction. In comparison to, for example, speech input, such motion-based gestures are perceived as more socially acceptable [11].

[1] `http://www.fitbit.com/`, last accessed 22/10/2015

MUM '15, November 30-December 02, 2015, Linz, Austria

DOI: `http://dx.doi.org/10.1145/2836041.2836063`

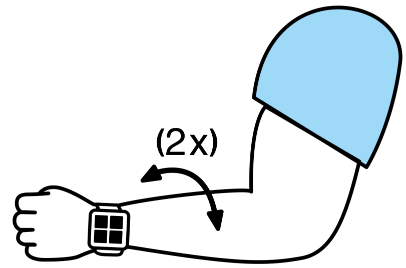
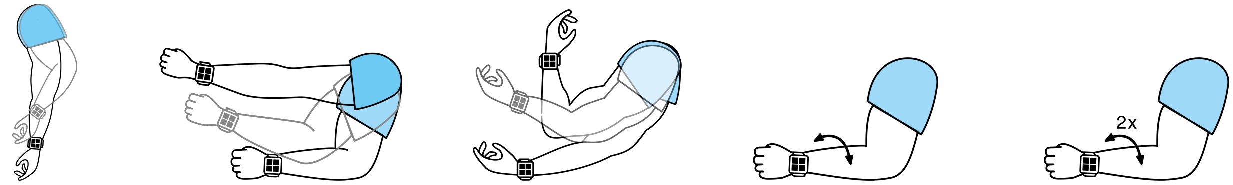


Figure 1: *WristRotate* is a gesture delimiter for wrist-worn devices which is easy to perform and uncommon in most people's everyday life.

The crucial factors for high user acceptance of a motion-gesture based system are high reliability and high recognition rates with low rates of false positives. Compared to smartphones, this is an even bigger problem for wrist-worn devices. While smartphones stay in our pockets most of the time, these devices are fastened around our arms, meaning that they will be in nearly constant motion. Especially as we do complex motions with our arms, these could easily be misinterpreted as a gesture (the so-called Midas touch problem). For the interpretation of gestures based on internal sensors, a variety of techniques exist, such as [6]. But those are not designed to cope with the frequent movement of our wrists and often include the need to press a button for activation. To avoid this, an effective and easy-to-use delimiter to separate non-relevant motion from gesture input is needed. Afterwards, the aforementioned gesture interpreter can be easily applied.

In this paper we present *WristRotate*, a gesture delimiter that makes it possible to separate non-relevant motion from gesture input done on purpose. We are following the approach of DoubleFlip presented by Ruiz and Li [12]. Through an extensive data collection we were able to identify a gesture that is (1) uncommon in most people's daily life; (2) quick and easy to execute and (3) easy and reliable to detect. The gesture is executed by simply rotating the lower arm and wrist outwards and back inwards (see Figure 1).

RELATED WORK
Various ways to detect gestural interaction with a mobile device exist, e.g. [6, 7]. As they are designed to detect gestures when the device is hand-held, the difference from a wrist-worn device should be rather small. However, they require a button to be pressed while performing the gesture, which cannot be done on a wrist-worn device with only one hand.



(a) ***pull*** **(b)** ***push*** **(c)** ***curl*** **(d)** ***rotate*** **(e)** ***doubleRotate***

Figure 2: The gesture delimiter candidates.

Therefore a delimiter – a gesture that is distinctive enough not to occur in everyday usage – is needed to initiate the gestural commands. While such a delimiter has been found for hand-held devices by Ruiz et al. [12], it is not applicable to wrist-worn devices, as they are in constant motion and therefore more error-prone. Williamson et al. mention a simple gating gesture for wrist-based interaction [13], but do not give further information about its false-positive rate or suitability during everyday interactions. To close this gap, we investigate delimiter gestures for wrist-worn devices.

The increasing variety and popularity of smartwatches underpins the trend towards a wrist-worn computing device, but until now, only a limited set of interaction techniques has existed. Most of the related work focuses on Opposite-Site Interaction (OSI) techniques, meaning they require the hand that is not wearing the wearable device to operate it. These interaction techniques range from eyes-free input through tactile landmarks on the touch-screen [2] to extending the touch-screen to the entire wristband [9]. Xiao et al. even developed a multi-degree-of-freedom, mechanical interface for smartwatches [14] which allowed for continuous 2D panning and twisting as well as binary tilt and click. But all these interaction techniques require both hands for interaction. In this paper, we are more concerned with leveraging the capabilities of such a device by only making use of the arm that is wearing the device, so called Same-Side Interaction (SSI) [5].

With GestureWrist [10], Rekimoto et al. used capacitive sensors and an accelerometer to sense wrist-shape changes and measured forearm movements. This allows the user to input commands using only one arm, but requires additional sensors which are typically not available in today's wrist-worn devices. Kerber et al. employed electromyography (EMG) using a Myo wristband, allowing use of one-handed gestures [5]. While their preliminary study (comparing it against touch input) did not find any significant difference in terms of task completion time, keeping in mind the early state of commercial EMG devices, their results demonstrated the feasibility of SSI. To the best of our knowledge, the first direct comparison of OSI and SSI was presented by Kerber et al. [4]. They found the direct-touch interaction of the static peephole (OSI) to be on average 12% faster. But this is only marginal compared to the advantage of only using one arm to interact (SSI) when using a dynamic peephole.

We strongly believe that in many everyday situations, gestural interaction can be a convincing alternative to direct touch or speech input. Therefore, we aim to enable gestural SSI by finding a gesture delimiter providing the possibility to distinguish between non-relevant motion and intended gestures.

DESIGN OF WRISTROTATE

As already outlined, we aim to support not only wrist-worn devices with sophisticated input possibilities, such as smartwatches, but also those that only provide an accelerometer. As we also want to support interactions on the go, same-side interactions (SSI) are preferable. This opens up the possibility to carry something in one hand while interacting with the wrist-worn device worn on the other hand. Thus, we aim to detect a special gesture we can use as delimiter to distinguish between non-relevant and intended interactions.

Similar to [12], we defined a set of requirements that should be met by our delimiter gesture: (1) it should not be involuntarily invoked during everyday life; (2) it should not require complicated movements of the arm or wrist; (3) it should be easy to remember and quick to perform and (4) it should be reliably detectable when done on purpose.

Compared to the set of possible gestures that can be executed with a smartphone in the user's hand, the set of possible gestures for a wrist-worn device is already limited. Furthermore, we have to dismiss gestures that are complex (with respect to either length or required movements) as well as those that are typically used in existing applications (e.g. turning one's wrist inwards to activate a smartwatch). We also have to exclude interactions that often happen involuntarily during everyday life to avoid a high number of false positives.

In the end, we identified five gestures for further examination (see Figure 2):

(a) a pull gesture, for which the arm is hanging parallel to the body and is then pulled towards the shoulder (*pull*).

(b) a push gesture, where the arm is held parallel to the ground and pushed forward like a punch (*push*)

(c) a gesture where the arm is first stretched out (palm facing upwards) and then the lower arm is bent towards the shoulder similar to a bicep curl (*curl*)

(d) an outward rotation of the lower arm and wrist, quickly followed by a rotation back inward (*rotate*)

(e) an outward rotation of the lower arm and wrist, quickly followed by a rotation back inward, executed twice in a row (*doubleRotate*)

None of these gestures requires long or complex interactions, which also makes them easy to remember and quick to perform. To check for requirements (1) and (4), we conducted an extensive data acquisition and analysis.

DATA ACQUISITION

To ensure that our proposed delimiter gestures are not involuntarily invoked, we collected a corpus of smartwatch motion data during everyday activities. We recruited six volunteers (two female, all office workers), aged 23 to 33 (M=27.8, SD=3.6) and equipped them with an off-the-shelf Pebble Smartwatch as well as an LG Nexus 5 smartphone with our own data acquisition software on both devices. We selected the smartwatch because of its battery runtime and its three-axis accelerometer, which was sampled at a rate of only 50 Hz to save battery life. The recorded data was sent to the smartphone, as the Pebble only provided limited storage.

The participants were asked to wear and use the smartwatch as a replacement for their personal watch. Four participants wore the smartwatch on the side of their non-dominant hand whereas the other two used the dominant side. During the recording, the smartwatch displayed the current time as well as the actual recording status, thereby closely mimicking a regular watch. In case of a connection loss between the two devices, the recording was paused until the Bluetooth connection could be reestablished.

In total, we collected a set of 435.1 hours of sensor recordings from our participants. Most of the data was collected during normal working hours from 7am to 6pm, but recordings outside these periods (e.g. from weekends) were also collected.

THE GESTURE RECOGNITION SYSTEM

We use a three-stage approach to detect and classify a gesture. First, the recorded acceleration data is filtered to reduce noise and eliminate the gravity portion. Afterwards, the data needs to be segmented to isolate single samples which can then be sent to the classifier, which labels them based on its available training data.

Pre-Processing

The Pebble's three-axis accelerometer provides an estimate of the acceleration along each of the axes x, y and z separately measuring the acceleration calibrated to a maximum of $\pm 4\ G$. To transfer the raw accelerometer data (including noise and gravity) into linear acceleration values reflecting the pure motion of the wrist, we applied a combination of a low-pass and a high-pass filter. A simple low-pass filter is equivalent to a smoothing function and results in a noise-reduced signal which is less dependent on quick changes. As a consequence, the low-pass filtered data corresponds to the actual gravity. To filter out the acceleration caused by gravity, we applied a high-pass filter based on the results of the low-pass filter.

Segmentation

To identify potential gestures in the continuous sensor data stream, we developed an algorithm which detects the start and stop points of gestures in a continuous time series and thereby partitions the stream into segments. Only those segments that are considered as containing a gesture have to be examined by our classifier. We use an algorithm similar to the one utilized in [3], which is based on the assumption that the movement energy of the smartwatch increases over a certain threshold when a gesture starts and decreases below a specific threshold when the gesture ends. To compute the start and end points of a potential gesture, we use a sliding window approach with two overlapping windows of size $N = 5$.

Classification

For our classification, we utilize Dynamic Time Warping (DTW) – a technique well known from the field of speech recognition but also widely used in other areas for recognizing patterns in continuous data streams [1]. DTW measures the similarity between two time sequences of different time series which may vary in time or speed by warping them non-linearly in the time dimension and figuring out the costs to match them. We do not directly compare the raw accelerometer readings, but work with a derived value – the slope of the acceleration. We use an approximate DTW implementation provided by the FastDTW project[2] which is characterized by an improved computation time and memory complexity ($O(N)$) in contrast to the original DTW algorithm ($O(N^2)$).

For the comparison based on the DTW algorithm, a set of labeled gestures has to be provided as training data. A gesture that should be classified is then compared to all training gestures and the computed matching cost value is stored in an increasingly sorted score table, i.e. the training gesture with the highest similarity is saved at the top position. However, this gesture might still be very different from the gesture to classify. We therefore consider the computed matching cost – if it is above a pre-defined threshold, we do not consider the gesture as recognized. Furthermore, we make use of an adapted k-Nearest-Neighbor approach, i.e. we check the first k entries from the score table and label the gesture to classify as the one that is most often present in these k entries. We empirically chose a threshold value of 2 and set k to 3.

RESULTS

Overall, 7501 segments or potential gestures were detected in the 435.1 hours of collected data. We classified the potential gestures to be able to make a statement regarding the false-positive (FP) rate, i.e. the number of erroneously detected gestures in relation to the total number of examined gestures. The results for all participants can be seen in Table 1.

Table 1: Results of the user-specific classification of possible gestures in the 435.1 hours of recordings of normal smartwatch usage.

	pull	*push*	*curl*	*rotate*	*dblRotate*
Detections	40	79	126	608	10
FP rate [in %]	0.53	1.05	1.68	8.11	0.13
Avg. per hour	0.09	0.18	0.29	1.40	0.02

As can be seen from the results, the *doubleRotate* gesture performs best, as only ten occurrences were detected. Although it is not obvious from the combined results, the *curl* gesture also performed quite well for all but two participants, as 124 of the detected matches originate from only two participants. A one-way ANOVA revealed a significant difference in the FP rate between the gestures ($F_{4,25} = 4.91, p < 0.01$). Post-hoc

[2] `https://code.google.com/p/fastdtw/`, last accessed 22/10/2015

comparions using the Tukey HSD test indicated a significant difference ($p < 0.01$) between *pull* (M=0.01, SD=0.01) and *rotate* (M=0.1, SD=0.05) as well as between *rotate* and *doubleRotate* (M=0.003, SD=0.003). There were no significant differences between the other pairs of gestures.

Although a low false-positive rate is unquestionably important, a high true-positive (TP) rate, i.e. the number of correctly detected gestures in relation to the actual number of executions, is also required to ensure that a gesture is correctly recognized when done on purpose. To test this aspect, we collected a separate gesture set, in which all participants did the specific gestures ten times each, and tested those with the user-specifically trained classifier. The results can be seen in Table 2.

Table 2: Results of the user-specific classification of the 300 purposely recorded gesture executions.

	pull	*push*	*curl*	*rotate*	*dblRot*
Detected segments	33	51	54	58	59
Correct detections	30	50	54	57	48
TP rate [in %]	90.9	98.0	100	98.3	81.4
Combined TP rate [in %]	50.0	83.3	90.0	95.0	80.0

The results provide two insights: First, as every participant executed each gesture ten times, it was expected that 60 segments per gesture would be detected, but actually, only a part of the segments could be found – especially for the pull gesture. An analysis of the data that was not correctly segmented showed that the gestures were executed too slowly, resulting in not reaching the required energy level to start a segment. Second, regarding the TP rate, a one-way ANOVA revealed a significant difference between the gestures ($F_{4,25} = 3.25, p < 0.05$). Post-hoc comparions using the Tukey HSD test indicated a significant difference ($p < 0.05$) between *pull* (M=5.5, SD=4.81) and *rotate* (M=9.67, SD=0.82) as well as between *pull* and *doubleRotate* (M=9.83, SD=1.33). There were no statistically significant differences between the other pairs of gestures.

We distinguish between the TP rate of the classification alone (third row of Table 2) and the one of the overall gesture recognition system (counting the non-identified segments as missed gestures as well: fourth row of Table 2). From a user's perspective, the latter is more relevant, as it characterizes the effort the user has to make to successfully trigger the gesture. A one-way ANOVA revealed a significant difference between the gestures ($F_{4,25} = 2.78, p < 0.05$). Post-hoc comparions using the Tukey HSD test indicated a significant difference ($p < 0.05$) between *pull* (M=5.0, SD=4.52) and *rotate* (M=8.0, SD=3.03). There were no statistically significant differences between the other pairs of gestures.

DISCUSSION

Our analysis of the 435.1 hours of recorded daily motion data shows that three of our proposed delimiter gestures are only rarely detected during daily motion, resulting in a low false-positive rate, i.e. it is in general not to be expected that the chosen delimiter gesture would often be activated erroneously. Although the *rotate* gesture provides a significantly higher false positive rate, it is still only activated about once an hour. For the best-performing gesture (*doubleRotate*), a single false positive is expected only every 43.5 hours.

The results for the segmentation process show that it works with good performance for four out of our five proposed gesture candidates, but they also reveal that the detection rate of segments containing the *pull* gesture is significantly worse. As a consequence, also regarding the true-positive rate of our proposed gestures, a significant difference between *pull* and *(double)Rotate* could be observed. Therefore, the *pull* gesture cannot be considered a good candidate for a delimiter gesture as it does not fulfill our fourth requirement of being reliably detectable when done on purpose.

When it comes to personal preference, we observe a clear trend towards the *rotate* and the *doubleRotate* gesture – especially as they can be executed in an unobtrusive way.

Taking all the above-mentioned points into account, we consider both the *rotate* and the *doubleRotate* gestures suitable as motion gesture delimiters. Depending on a user's preference, either the one with the particularly low FP rate (*doubleRotate*) or the one with slightly better TP rate (*rotate*) can be chosen.

As shown by Ng et al. in [8], wrist rotations are in general also feasible when walking or carrying something. In contrast to their examined use case, we do not require high accuracy when executing the gesture. The significantly different movement time is not considered problematic, as it is one of the core features of our Dynamic Time Warping-based classifier to match samples that vary in speed.

CONCLUSION AND OUTLOOK

In this paper, we presented *WristRotate*, a personalized motion gesture delimiter to separate non-relevant motion from gestures done on purpose. We implemented a gesture recognition system that is capable of detecting segments that could potentially contain gestures from a continuous stream of acceleration data as well as classifying these found segments based on a Dynamic Time Warping approach. Our examinations of five gesture delimiter candidates revealed that the *rotate* and the *doubleRotate* gesture, i.e. quickly rotating the lower arm and wrist outwards and back inwards (twice), are best suited in terms of true- and false-positive rates.

For future work, we will target a user-independent classification with the goal to eliminate the currently necessary user-specific training phase. With a user-independent implementation, the classifier can be trained beforehand and is therefore immediately operational for the user. In the end, we envision an in-the-wild study targeting aspects like executing the gesture while walking or while carrying something.

ACKNOWLEDGEMENTS

This research was partially funded by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS13015A (project SmartF-IT).

REFERENCES

1. Donald J. Berndt and James Clifford. 1994. Using Dynamic Time Warping to Find Patterns in Time Series. In *KDD Workshop*, Vol. 10. 359–370.
2. Gabor Blasko and Steven Feiner. 2004. An Interaction System for Watch Computers Using Tactile Guidance and Bidirectional Segmented Strokes. In *Proc. ISWC '04*, Vol. 1. IEEE, 120–123.
3. Kitae Hwang and Jae Moon Lee. 2013. An Implementation Experience of Accelerometer-Based Gesture Recognition with Android Smartphone. *International Journal of Advancements in Computing Technology* 5, 12 (2013), 305.
4. Frederic Kerber, Antonio Krüger, and Markus Löchtefeld. 2014. Investigating the Effectiveness of Peephole Interaction for Smartwatches in a Map Navigation Task. In *Proc. MobileHCI '14*. ACM, 291–294.
5. Frederic Kerber, Pascal Lessel, and Antonio Krüger. 2015. Same-Side Hand Interactions with Arm-Placed Devices Using EMG. In *Proc. EA CHI '15*. ACM, 1367–1372.
6. Sven Kratz and Michael Rohs. 2010. A $3 Gesture Recognizer: Simple Gesture Recognition for Devices Equipped with 3D Acceleration Sensors. In *Proc. IUI '10*. ACM, 341–344.
7. Sven Kratz, Michael Rohs, and Georg Essl. 2013. Combining Acceleration and Gyroscope Data for Motion Gesture Recognition Using Classifiers with Dimensionality Constraints. In *Proc. IUI '13*. ACM, 173–178.
8. Alexander Ng, Stephen Brewster, and Andrew Crossan. 2011. The Effects of Encumbrance on Mobile Gesture Interactions. In *Proc. Body, Movement, Gestures & Tactility in Interaction with Mobile Devices, Workshop at MobileHCI 2011*.
9. Simon T. Perrault, Eric Lecolinet, James Eagan, and Yves Guiard. 2013. WatchIt: Simple Gestures and Eyes-Free Interaction for Wristwatches and Bracelets. In *Proc. CHI '13*. ACM, 1451–1460.
10. Jun Rekimoto. 2001. GestureWrist and GesturePad: Unobtrusive Wearable Interaction Devices. In *Proc. ISWC '01*. IEEE, 21–27.
11. Julie Rico and Stephen Brewster. 2010. Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability. In *Proc. CHI '10*. ACM, 887–896.
12. Jaime Ruiz and Yang Li. 2011. DoubleFlip: A Motion Gesture Delimiter for Mobile Interaction. In *Proc. CHI '11*. ACM, 2717–2720.
13. Julie R. Wiliamson, Andrew Crossan, and Stephen Brewster. 2011. Multimodal Mobile Interactions: Usability Studies in Real World Settings. In *Proc. ICMI '11*. ACM, 361–368.
14. Robert Xiao, Gierad Laput, and Chris Harrison. 2014. Expanding the Input Expressivity of Smartwatches with Mechanical Pan, Twist, Tilt and Click. In *Proc. CHI '14*. ACM, 193–196.