Prevention of Unintentional Input While Using Wrist Rotation for Device Configuration

Han Joo Chae[1] **Jeong-in Hwang**[2] **Yuri Choi**[3]
Yieun Kim[4] **Kyle Koh**[5] **Jinwook Seo**[6]

[1,2,4,5,6]Seoul National University, [3]Samsung Electronics, Seoul, Korea
{[1]hanjoo, [2]jihwang, [4]yekim, [5]kyle}@hcil.snu.ac.kr, [3]yuri82.choi@samsung.com, [6]jseo@snu.ac.kr

ABSTRACT
We describe the design of the safeguard interface that helps users avoid unintentional input while using wrist rotation. When configuring the parameters of various devices, our interface helps reduce the chance of making accidental changes by delaying the result of input and allowing the users to make deliberate attempt to change the parameters to their desired value. We evaluated our methods with a set of user experience and found that our methods were more preferred when the end-results of configurational changes of the devices become more critical and can cause irreversible damage.

Author Keywords
Internet of Things; Radial Meter; Wrist Rotation

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

INTRODUCTION
With an imminent era of ubiquitous Internet of Things(IoT) and Augmented Reality(AR) in our everyday environment, it becomes necessary for the users to configure the devices like the volume of speakers, the brightness of light bulbs, or the temperature of heaters in quick and efficient ways, without having to physically get closer or use dashboards on smartphones. For such input modals, wrist rotation has been vastly explored and demonstrated to be robust in selecting menu items and changing parameters [2]. Also, it does not always require users to uphold their hands in the air causing fatigue [1] and can be detected via the wearable devices attached to their body at any given time, resulting in more seamless experience [3].

However, the lack of reference point can cause wrist rotation angle to be at dangerous points. For example, if the wrist rotation angle was mapped to the volume of a speaker, users may accidently raise the volume to its maximum setting by rotating the wrist to its maximum twist position, which can cause damage to ears. Similar scenarios are conceivable with light bulbs or gas stoves. Many techniques such as 'slide to unlock' found in smartphones [4] have been suggested in order to alleviate the issue of the accidental inputs. In this paper, we present the design of the prototype that is resilient against the unintentional inputs and their safety issues unique to the wrist-rotation based interfaces.

SELECTION MECHANISM
We adopted a dragging-like selection mechanism where a user is required to touch-and-hold while making changes using wrist rotation. As a touching gesture demonstrates a clear intention for adjustment, unnecessary noise prior and after the wrist rotation can be eliminated. In our prototype, we used a ring-type capacitive sensor to detect touch gestures and a smart watch for wrist rotation (Fig. 1).

IMMEDIATE VS. NON-IMMEDIATE UPDATE
When a setting is being changed through a wrist rotation, there are two different ways to update the changes to the object: immediate and non-immediate. Immediate update can be useful for observing volume or brightness changes, while annoying when accidental changes are made. On the other hand, non-immediate update is error-safe and more adequate when there is no need for a closed-loop feedback.

SAFEGUARD FEATURES
As shown in Fig. 2a, the GUI consists of three components: a cursor, a gauge and a target, which indicate the current position of a user's wrist, the current value of a setting, and the goal where the gauge needs to be moved (Fig 2b), respectively. Such visual feedback is important for users to understand the range of motion [2] and can be rendered on built-in displays or be directly shown over the devices with AR techniques. In addition to the baseline method with no safeguard feature, we present four methods specifically designed for preventing unintentional inputs.

Absolute Angle (*AA*) - Baseline
A user's wrist angle (-65° to 60°) [2] is absolutely mapped to the cursor of the radial meter and the gauge always moves together with the cursor so that any changes on the wrist angle (cursor) are applied immediately. As no safeguard

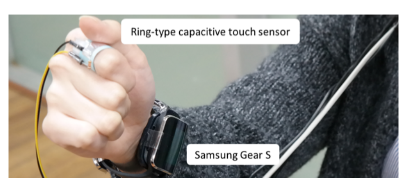

Figure 1. Samsung Gear S detects wrist rotation and ring-type capacitive sensor detects touch gestures.

UIST'16 Adjunct, October 16-19, 2016, Tokyo, Japan
ACM 978-1-4503-4531-6/16/10.
http://dx.doi.org/10.1145/2984751.2984771

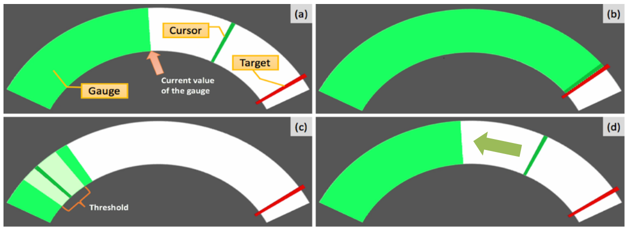

Figure 2. Overview of the user interface.

feature is provided in *AA*, accidental changes may occur when the user's initial wrist position is different from the current state.

Absolute Angle-Threshold (*ATH*)
We added a threshold to the *AA* by requiring a user to move a certain angle greater than the threshold, in order to start making changes. This would give the users a room to cancel before making any changes in cases of an accidental touch input or an unintended cursor position. The width of the threshold can be seen in the radial meter (Fig. 2c).

Absolute Angle-Magnet (*AM*)
AM requires a user to move the cursor to the current value of the gauge first (Fig. 2d, along the arrow), then the gauge will follow the cursor like *AA*. As *AM* forces the user to move the cursor to the current value before the gauge moves, it prevents any sudden changes in the actual value.

Absolute Angle-Trail (*AT*)
Similar to *AM*, *AT* requires a user to move the cursor to the current value of the gauge before it can be freely changed. However, until the cursor and the gauge are met, the gauge slowly moves towards the cursor automatically. This reduces the distance between them, while preventing any sudden changes. Also, instead of snatching the gauge, users can hold the cursor on top of the target and wait for the gauge to follow as if we hold the +/- button on the remote and wait for the volume to reach the desired level.

Relative Angle (*R*)
Unlike other methods, *R* uses a relative mapping. The value of the gauge changes only for the amount the wrist has rotated regardless of the wrist's starting position.

USER STUDY
We conducted a user study to evaluate the four interfaces against the baseline AA. 12 participants were recruited and asked to wear a smart watch and a ring-type capacitive sensor (Fig. 1). Each performed tasks to adjust the volume to the randomly given target (Fig. 2a) using all five different interfaces whose order was counterbalanced via Latin Square design. Interviews were conducted afterwards.

RESULTS AND DISCUSSION
Participants spent on Avg. of 1.99 (σ=.60), 2.28 (σ=.76), 2.75 (σ=.67), 2.82 (σ=.75), 3.27 (σ=1.18) seconds for *AA*, *ATH*, *AM*, *AT*, and *R* respectively to complete a task. One-way ANOVA and Tukey HSD post-hoc test have revealed that *AA* and *ATH* performed significantly faster than the others ($p<.001$) and *R* significantly slower than the others ($p<.001$). The difference between AA(Fastest) and R(Slowest) was 1.28 seconds, which were the maximum tradeoffs between the speed and the safety.

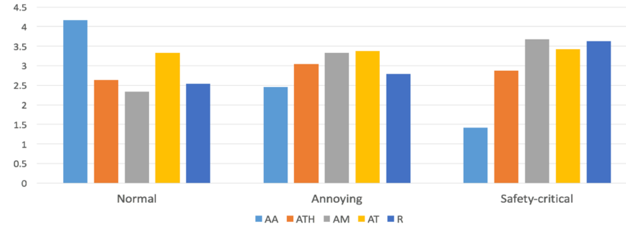

Figure 3. User rankings among interfaces for normal, annoying and safety-critical situations.

During the interview, participants were asked to rank the interfaces in three different situations: *normal*, *annoying*, and *safety-critical*. For *normal*, they were asked to rank solely based on their preferences after using the interfaces. For *annoying*, they ranked considering the cases where unintentional change could be annoying (e.g. sudden change in volume). For *safety-critical*, they ranked considering the cases where unintentional change could cause irreversible damage (e.g. sudden burst of flame on a stove). Since *AT* was selected as top 2 in *normal* and top 1 in *annoying* (Fig. 3), we found that *AT* had a balance between the preferences among users and safety. In real scenarios, as *normal* and *annoying* are the most common, it can be argued that AT fits the best for general purposes. "(*AT*) prevents unnecessary popping…it works well with volume control (P6)."

CONCLUSION
In this paper, we present the design and prototype that helps users avoid unintentional inputs that may cause safety issue, while using wrist rotation as input. From an evaluation, we found that people preferred to have such safeguard methods in the *annoying* or *safety-critical* situations, which means people were willing to trade off the time performance for the safety. In the future, we want to explore the ways to expand our interfaces into other input modals, such as touch or mid-air gestures.

ACKNOWLEDGMENTS
This work was sponsored by Samsung Electronics. Jinwook Seo is the corresponding author.

REFERENCES
1. Hincapié-Ramos, Juan David, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. Proc. CHI'14. 1063-1072. 2014.
2. Rahman, Mahfuz, Sean Gustafson, Pourang Irani, and Sriram Subramanian. Tilt techniques: investigating the dexterity of wrist-based input. Proc. CHI'09. 1943-1952. 2009.
3. Rekimoto, Jun. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. Proc. ISWC'01. 21-27. 2001.
4. Truong, Khai N., Thariq Shihipar, and Daniel J. Wigdor. Slide to X: unlocking the potential of smartphone unlocking. Proc. CHI'14. 3635-3644. 2014