GraspTracker: tracking user grab posture with built-in sensors on mobile devices

Team GraspTrackers
SeungJoo Lee, HyungJun Yoon
Korea Advanced Institute of Science and Technology (KAIST)

I. Research Problem & Literature Review

Mobile devices are used in countless user contexts, which constantly change over time. To provide the best user experience, the interaction method should be aware of the users context and adapt to it in real-time. Thus, it is important for mobile devices to grasp the user context. Especially, the hand posture of the user should be considered as a critical factor, since touching is the most dominant interaction method. iGrasp [3] devised a hand posture-aware keyboard interface and showed its effectiveness by reducing 42% of typing time in its evaluation, which reveals the importance of hand posture awareness. However, our mobile devices yet do not provide the user's hand posture context, thus cannot be adapted accordingly.

To address this problem, several works have been proposed. Lung-Pan Cheung et al. [3], Kim et al. [4], and Harrison et al. [5] used additional capacitive touch sensors to sense how the users grab mobile devices. However, they leveraged additional sensing devices that are not usually equipped with commodity mobile devices, incurring additional costs and inconveniences to users.

To overcome prior works limitation, Mayank Goel et al. [2] detected hand posture on mobile phones by combining the three features: touch size, the shape of the swipe arc, rotation of the device, which can be measured using built-in sensors. Nonetheless, it senses the hand posture based on the users already-taken action including swipe and touch; it predicts the hand posture after the user conducts five interaction steps. There would be some time gap between the users actual hand posture change and the prediction by the application, which can possibly degrade the users QoE.

In this study, we aim to solve the stated problem from the following perspectives: tracking user grab posture transition *(i) only using built-in sensors, and (ii) in real-time without time lag.*

II. Plan & Deliverables

Until the end of this semester, we will build an android application that tracks the user's real-time grab posture. GraspTracker will be evaluated using predefined grasp postures which are widely used by users as ground truth [6].

We will decide whether we have succeeded or not by the following criteria. *(i) The hand posture classification accuracy is above 85%, and (ii) whether the user is satisfied with the delay between posture change and GraspTracker detection.* The criteria performance was chosen based on state-of-the-art using built-in sensors [2].

Our plan for this semester is as follows:

10/1 $\sim$ 10/15	Study android programming Reproduce GripSense [2]
10/16 $\sim$ 10/26	Gather sensor data & determine what sensor to use
10/27 $\sim$ 11/27	Implement classifier & In-lab experiment
11/28 $\sim$ 12/5	Evaluation & user test
12/6 $\sim$ 12/9	Project demo preparation

III. Role of each member

For a literature search, demo and presentation, we will have same weight on preparation. For the other research processes, we distributed as follows.

Seungjoo Lee	Android app building Sensor data acquiring & processing
HyungJun Yoon	Machine learning classifier Evaluation & user test

IV. References

[1] How Do Users Really Hold Mobile Devices?, *UX Matters '13*

[2] Mayank Goel, Jacob O. Wobbrock, Shwetak N. Patel, GripSense: using built-in sensors to detect hand posture and pressure on commodity mobile phones, *UIST '12*

[3] Lung-Pan Cheng, Hsiang-Sheng Liang, Che-Yang Wu, Mike Y. Chen, iGrasp: Grasp-based Adaptive Keyboard for Mobile Devices, *ACM CHI '13*

[4] Kim, K.-E., Chang, W., Cho, S.-J., et al. Hand Grip Pattern Recognition for Mobile User Interfaces. *Proc. AAAI '06.*

[5] Harrison, B.L., Fishkin, K.P., Gujar, A., Mochon, C., and Want, R. Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces. *Proc. CHI '98.*

[6] Le, Huy Viet, et al., Finger placement and hand grasp during smartphone interaction, *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. ACM, 2016.*