MyoKey: Surface Electromyography and Inertial Motion Sensing-based Text Entry in AR

Young D. Kwon*, Kirill A. Shatilov*, Lik-Hang Lee‡, Serkan Kumyol*, Kit-Yung Lam *,
Yui-Pan Yau *, and Pan Hui*†
*Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR
†Department of Computer Science, The University of Helsinki, Finland
‡Center for Ubiquitous Computing, The University of Oulu, Finland
Email: {ydkwon, kshatilov, lhleeac, skumyol, kylambd, ypyau, panhui}@cse.ust.hk

***Abstract*—The seamless textual input in Augmented Reality (AR) is very challenging and essential for enabling user-friendly AR applications. Existing approaches such as speech input and vision-based gesture recognition suffer from environmental obstacles and the large default keyboard size, sacrificing the majority of the screen's real estate in AR. In this paper, we propose MyoKey, a system that enables users to effectively and unobtrusively input text in a constrained environment of AR by jointly leveraging surface Electromyography (sEMG) and Inertial Motion Unit (IMU) signals transmitted by wearable sensors on a user's forearm. MyoKey adopts a deep learning-based classifier to infer hand gestures using sEMG. In order to show the feasibility of our approach, we implement a mobile AR application using the Unity application building framework. We present novel interaction and system designs to incorporate information of hand gestures from sEMG and arm motions from IMU to provide seamless text entry solution. We demonstrate the applicability of MyoKey by conducting a series of experiments achieving the accuracy of 0.91 on identifying five gestures in real-time (Inference time: 97.43 ms).**

***Index Terms*—Textual Input, Augmented Reality, EMG, IMU, Deep Learning**

I. Introduction

The advancing mobile hardware has facilitated the development of Augmented Reality (AR) applications [1]. AR overlays virtual content directly on top of the user's physical surroundings. The virtual content can take diverse forms such as icons, menus, windows, and keyboards. The default interaction design (e.g., controlling a cursor with a mini-touchpad wired to the smartglasses or vision-based hand gesture recognition) enables users to interact with virtual content. In many AR applications, large-size virtual content is easy to locate. However, the default approaches are significantly inefficient for textual input, which involves small-sized content selection. More specifically, selecting character keys on a virtual keyboard is error-prone and inefficient [2] since choosing small character keys for a highly repetitive task is difficult [3].

There are various approaches to achieve seamless user experiences for text input in AR. First, speech recognition can be used to input words by recognizing the voice of users. However, it is limited by several major drawbacks [4]: (1) Privacy, where the user may disclose sensitive information to people around; (2) Noise, where the environmental noise can cause unintended textual input or corrupt the intended one. Second, vision-based approaches can detect hand to input character keys. However, it has the following limitations: (1) The virtual QWERTY keyboard, such as the one in Microsoft Hololens, occupies the large surface of the screen's center area; (2) Computer vision is susceptible to occlusion and lighting conditions, and without additional markers, vision-based tracking of hands at arbitrary orientations over a large area is challenging [5]. Lastly, existing mid-air taps methods such as HIBEY system [3] and LEAP Motion sensors [6] can be a solution for portable devices, but lifting the user's arm in prolonged time with such approaches suffer from hand fatigue [7] and limited device accessibility in mobile scenarios [8].

Accordingly, we propose the **MyoKey**, an unobtrusive solution to input characters and words in AR. MyoKey can recognize various hand gestures and arm motions by utilizing external sensors on the user's forearm, collecting two essential information: (a) surface Electromyography (sEMG), recognizing hand gestures through measuring the electrical potential from muscle cells. (b) Inertial Measurement Unit (IMU), tracking the motion and orientation of the user's arm. MyoKey demonstrates three advantages over the prior works. First, sEMG collects more robust signals than computer-vision based methods whose performance varies by the occlusion and lighting conditions. Second, it enables off-hand interaction with the AR environment instead of performing hand gestures in front of the facial area, causing arm fatigue [7]. Third, the MYO armband device is non-invasive and socially acceptable [8], where users can perform gestures naturally in their waist level. MyoKey also minimizes the keyboard interface to reserve the majority of the screen space for applications in the holographic environment. The contributions of our work include as follows: (1) We present a novel interaction and system design to enhance text inputs in the constrained AR environment by leveraging sEMG and IMU signals. (2) We show the preliminary results to demonstrate the possibility of our system by achieving a high accuracy of 0.91 on recognizing five gestures in real-time (Inference time: 97.43 ms). (3) The natural gestures in our design can support off-handed interactions with text entry interfaces of many contexts

(a) Five Gestures (b) Left and Right Motion

Figure 1: Hand gestures and arm motions used for the interaction

(in-text or keyboard cursors/ recommended words/ character deletion).

II. Related Work

We discuss the most relevant works on character selection in AR, as well as deep learning models and mobile AR driven by sEMG and IMU.

A. Textual Input in AR

On-device interaction [8] enables character selection, for example, on a sensible surface of a device: Taps and Swipes [9] on the spectacle frame of smartglasses, and an addendum ring surface allowing cursor pointing to characters within finger space [10] for subtle inputs [11]. The body-center interaction method is based on an interface attached to the user's body. An infrared sensor-based study, Palmtype [12], used a wrist band with multiple sensors to create a virtual palm keyboard using a visual display. In MyoKey, we explore the IMU-induced and sEMG-driven interaction for character inputs. The freehand interaction-based approach [13] relies on visual recognition of hand movements to obtain textual input. HIBEY [3] explores the 1-line keyboard configuration but neglects the interaction with in-text characters, while MyoKey addresses both the in-text (selection/deletion of typed characters) and keyboard interaction (selecting new characters/ recommended words).

B. Deep Learning and Mobile AR using sEMG and IMU

Convolutional neural networks (CNNs) is a member of the artificial neural network (ANN) family, which is widely applied on mobile AR applications [14]. Researchers utilize deep learning on user behavior analytics [15] and mobile AR system for object recognition and context-aware tracking [16]. Surface Electromyography (sEMG) is a non-invasive method to quantitatively measure the electrical potential differences between muscle and ground electrodes. Several studies use sEMG to tackle gesture recognition tasks [17]. Another study [18] attempts to identify in-hand objects using IMU and sEMG sensors in wearable devices. In contrast, MyoKey leverages IMU and sEMG sensors to achieve robust and off-hands gestural inputs in AR.

III. Design of Text Entry Interface

This section focuses on the design hand gestures and arm motions that map with the 1-line text entry layout.

Table I: Hand gestures and arm motions with associated functions in MyoKey

Signals	Interaction	Function in the context
sEMG (Hand Gesture)	One	move cursor left on the text
	Two	move cursor right on the text
	Three	delete a character that cursor is on
	Four	select a recommended word
	Fist	select a character from the keyboard
IMU (Arm Motion)	Left	move keyboard cursor left
	Right	move keyboard cursor right

A. Hand Gestures

The set of gestures we used as input to our system are derived from the internationally recognized well-known number of gestures due to their usage in American Sign Language (ASL), with the following motivation. User familiarity with ASL leads to better sEMG signals. Also, standard gestures improve replicability. As shown in Figure 1, we use an ASL gesture subset where the numbers '1', '2', '3', '4' map to the ASL number gestures while the gesture fist represents the number '0'). We did not take the sEMG signal resolution of each gesture into account while choosing the gesture set. Thus, we keep the gesture set simple and easily recognizable by any learning model. Each gesture serves to represent a selection mode in the text entry interface. Table I lists the functions of five gestures. Gestures '1' and '2' govern the left and right movements of the in-text cursor (i-cursor), while gesture '3' enables the character deletion at the i-cursor position. Gesture '0' (Fist) and '4' activates the IMU-induced keyboard cursor (k-cursor) driven by the user's arm motions, and the selection of recommended words, respectively. When the user holds the gesture '0' to locate the k-cursor at a character more than 500 ms dwell time, the character will be selected into the text. Multiple characters can be chosen until the gesture '0' being released. Besides, the recommended word at the k-cursor position is selected once the user releases the respective gesture.

B. Arm Motions

Users with MyoKey control the k-cursor with arm movements through capturing the IMU data at 60 Hz from the 3-axis MYO armband. In the 1-line keyboard layout, we neglect other axis data to avoid unnecessary cursor movements in the layout and hence relieve the users' physical loading in the repetitive task of character inputs. The k-cursor movement corresponds to the x-axis data ranging from 0°to 110°linearly mapping to the 27 characters in the 1-line layout.

C. 1-line Layout Configuration

Inspired by a prior work designing the minimized interfaces [3], we arrange the 27 characters (i.e., the 26 Roman alphabets (a – z) and the white space '_') in a horizontal formation in the alphabetical order (we position the white space after the alphabet). The alphabetical layout helps the user learning of the visible layout intuitively and brings

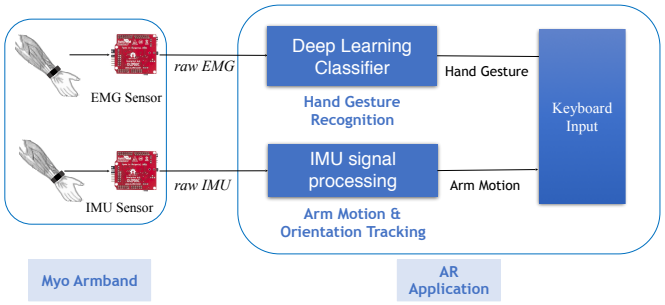

Figure 2: The system overview of MyoKey.

benefits to MyoKey such as performance improvement [19] and better usability to novice users [20]. Besides, prior works on 1-line layouts demonstrated that the alphabetical order outperforms the QWERTY and ENBUD layouts [21]. The 1-line characters and recommended words locate at the upper edge area to reserve the limit-size screen real estate.

IV. System Design

In this section, we present the system design of MyoKey, including the EMG, Classification model, an AR application.

A. EMG

EMG is measured from electrical signals generated by the muscle tissues by using electrodes on the human skin. There are two types of electrodes as the skin surface electrode (non-invasive) located near this field and the needle electrode (invasive) inserted in [22]. We use a skin surface based electrode which sends signals based on the action potentials of the muscle, which leads voltage with both positive and negative peaks. EMG devices can collect and amplify the signals generated by the human muscles, process and transmit them to other devices. The mapping of the collected signals to the user's gesture is highly dependent on both the fidelity of the collected signals and the selected classification method. We employed an MYO armband to collect signals generated by the arm muscles. In MyoKey, the MYO armband connects to the mobile companion via Bluetooth.

B. Classification

For the classification of sEMG signals, we used a convolutional neural network (CNN). Eight channels of temporal data that are streamed by sEMG sensors withhold multiple patterns recognizable by the employed CNN. Generally, recognition is sensitive to the sensor positioning as a temporal picture might shift dramatically. Employed CNN consists of 5 convolutional layers, one fully connected (with a dropout rate of 50%), as well as the softmax layer defining the output. During the training phase kernels of each hierarchical convolutional layers learn patterns within one or multiple sEMG channels: the first layer consists of 25 $[1 \times 10]$ filters; the second - 25 $[2 \times 25]$ filters; the third - 50 $[10 \times 25]$ filters; the fourth - 100 $[10 \times 50]$ filters; the fifth - 200 $[10 \times 100]$.

Table II: The average accuracy and inference time (standard deviation) for the discussed gestures.

Average Accuracy (SD)	Average Inference Time (SD)
0.91 (0.040)	97.43 ms (1.424)

C. AR Application

We present a system overview of MyoKey consisting of a Myo armband and AR application based on the Unity engine. As shown in Figure 2, MYO armband is responsible for collecting the raw sEMG and IMUs data and sending them to the AR application. After having established the CNN model for hand gesture recognition and the IMU signal processing unit, our AR application features the following elements. First, the application displays to users the 1-line keyboard layout with both character keys typed words at the top edge. Second, full AR-headset supports using the HoloKit (https://holokit.io/), which is designed to provide users with mixed reality experience at a low cost.

V. Experiments and Evaluation

In this section, we evaluate the performance of the classifier for the discussed gesture set in Section III-A.

Dataset. We first employ the data collected in [23]. To estimate the classifier's performance, we randomly select five subjects. Then, we conduct our experiments on those participants who are all right-handed without any muscular condition or skin allergy reported. Their ages range from 23 to 34 years old. The average time of train data recording was around 5 minutes for each subject. Within the experiment, participants performed gestures with the armband on, following the instructions. Three 30 seconds long records are done per gesture for each subject on the 200Hz frequency. The first two records are used to establish train sets. The last record is used to construct test trials. Each record is then divided into 28 separate trials of 200 samples each. Note that there are 30 trials per 30 seconds, but first and the last second are trimmed because they may not have any essential information. The trials are further trimmed according to the selected time window of a current experiment. We trained the classifier and ran tests on collected static data. Classifier training is performed on the laptop, and takes approximately 30 minutes, averaged on 750 epochs per single subject.

Performance. Table II shows the average accuracy and standard deviation of the models for all subjects. Our CNN model achieves a high accuracy of 0.91 with a small standard deviation, which can satisfy multiple contexts in the text input interfaces. Figure 3 shows the confidence matrix of cumulative accuracy across all subjects. From the confusion matrix, we can observe that it is challenging for our CNN model to distinguish between Gesture '2', '3', and '4'. In other words, results of all subjects (1 – 5) show higher error rates with Gestures '2', '3' and '4' than other gestures, while Gestures '0' (Fist) and '1' have less error prune. It is because those confusing gestures are physically similar to each

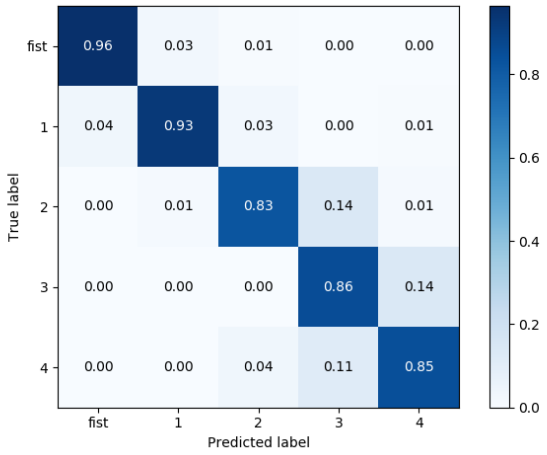

Figure 3: Confusion matrix for the subject-gesture experiment.

other. We believe that the larger size of training data or data augmentation methods can be of help to further improve the performance of the model since the per-subject data is severely limited. Also, different gesture sets can improve accuracy.

Inference Time. The inference time indicates how much the model spends to predict using one data sample. As shown in Table II, the average inference time (i.e., latency) of our model is around 97.43 ms on average, which is small enough to provide continuous classification in real-life application [17]. For the time measurement, we use a 2.5 GHz Intel Core i7 with two cores and two 8GB LPDDR3 DRAM.

VI. Conclusion and Future Work

In this paper, we proposed MyoKey, a sEMG and IMU sensing-based system to effectively and unobtrusively input texts in the constrained screen real estate of AR, with two prominent designs: (1) Hand gesture recognition using sEMG signals; (2) Novel interaction design enabling users to naturally select character keys based on IMU signals and 1-line keyboard layout with off-hand posture. Our preliminary results demonstrate CNN's capability of finding patterns from noisy signals and hence identifying hand gestures with an alternative modality. The multiple gestures with a reasonable recognition rate can satisfy the multiple contexts in the text input interfaces. For future works, we will investigate the combination of gestures to the design of minimalist interfaces, for example, splitting the 1-line character layouts by two gestures for less ambiguous IMU-induced k-cursor pointing. Accordingly, we will build language models for different splitting configurations. Finally, we will evaluate the performance of these configurations in terms of typing speed and accuracy.

Acknowledgments

This research has been supported in part by project 16214817 from the Research Grants Council of Hong Kong, and the 5GEAR project and the FIT project from the Academy of Finland.

References

[1] K. Y. Lam, L. H. Lee, T. Braud, and P. Hui, "M2a: A Framework for Visualizing Information from Mobile Web to Mobile Augmented Reality," in *Proc. of PerCom '19*, Mar. 2019, pp. 1–10.

[2] J. D. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani, "Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions," in *Proc. of CHI '14*. NY, USA: ACM, 2014, pp. 1063–72, t.O., Canada.

[3] L. H. Lee, K. Yung Lam, Y. P. Yau, T. Braud, and P. Hui, "HIBEY: Hide the Keyboard in Augmented Reality," in *Proc. of PerCom '19*, Mar. 2019, pp. 1–10.

[4] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser, "Whose move is it anyway? Authenticating smart wearable devices using unique head movement patterns," in *Proc. of PerCom '16*, Mar. 2016, pp. 1–9.

[5] F. Haque, M. Nancel, and D. Vogel, "Myopoint: Pointing and Clicking Using Forearm Mounted Electromyography and Inertial Motion Sensors," in *Proc. of CHI '15*. NY, USA: ACM, 2015, pp. 3653–3656, seoul, Republic of Korea.

[6] X. Yi, C. Yu, M. Zhang, S. Gao, K. Sun, and Y. Shi, "ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3d Hand Tracking Data," in *Proc. of UIST '15*. NY, USA: ACM, 2015, pp. 539–548, nC, USA.

[7] Y.-C. Tung, C.-Y. Hsu, H.-Y. Wang, S. Chyou, J.-W. Lin, P.-J. Wu, A. Valstar, and M. Y. Chen, "User-Defined Game Input for Smart Glasses in Public Space," in *Proc. of CHI '15*. NY, USA: ACM, 2015, pp. 3327–3336, seoul, Republic of Korea.

[8] L. Lee and P. Hui, "Interaction Methods for Smart Glasses: A Survey," *IEEE Access*, vol. 6, pp. 28 712–28 732, 2018.

[9] C. Yu, K. Sun, M. Zhong, X. Li, P. Zhao, and Y. Shi, "One-dimensional handwriting: Inputting letters and words on smart glasses," in *Proc. of CHI '16*. NY, USA: ACM, 2016, pp. 71–82.

[10] S. Nirjon, J. Gummeson, D. Gelb, and K.-H. Kim, "Typingring: A wearable ring platform for text input," in *Proc. of MobiSys '15*. ACM, 2015, pp. 227–239.

[11] L. H. Lee, K. Y. Lam, T. Li, T. Braud, X. Su, and P. Hui, "Quadmetric optimized thumb-to-finger interaction for force assisted one-handed text entry on mobile headsets," *Proc. of IMWUT*, vol. 3, no. 3, pp. 94–121, Sep. 2019.

[12] C.-Y. Wang, W.-C. Chu, P.-T. Chiu, M.-C. Hsiu, Y.-H. Chiang, and M. Y. Chen, "Palmtype: Using palms as keyboards for smart glasses," in *Proc. of MobileHCI '15*. ACM, 2015, pp. 153–160.

[13] H. Huang and S. Lin, "Toothbrushing Monitoring Using Wrist Watch," in *Proc. of SenSys '16*. NY, USA: ACM, 2016, pp. 202–215, cA, USA.

[14] X. Ran, H. Chen, Z. Liu, and J. Chen, "Delivering deep learning to mobile devices via offloading," in *Proc. of the Workshop on VR/AR Network '17*. ACM, 2017, pp. 42–47.

[15] Y. D. Kwon, D. Chatzopoulos, E. ul Haq, R. C.-W. Wong, and P. Hui, "GeoLifecycle: User Engagement of Geographical Exploration and Churn Prediction in LBSNs," *Proc. of IMWUT*, vol. 3, no. 3, pp. 92:1–92:29, Sep. 2019.

[16] W. Zhang, B. Han, and P. Hui, "Jaguar: Low latency mobile augmented reality with flexible tracking," in *Proc. of MM '18*. NY, USA: ACM, 2018, pp. 355–363.

[17] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, "Self-Recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control Based on Convolutional Neural Network," *Frontiers in Neurosci.*, vol. 11, 2017.

[18] M. Theiss, P. M. Scholl, and K. Van Laerhoven, "Predicting Grasps with a Wearable Inertial and EMG Sensing Unit for Low-Power Detection of In-Hand Objects," in *Proc. of AH '16*. NY, USA: ACM, 2016, pp. 4:1–4:8, geneva, Switzerland.

[19] S. Zhai and B. A. Smith, "Alphabetically Biased Virtual Keyboards Are Easier to Use: Layout Does Matter," in *CHI EA '01*. NY, USA: ACM, 2001, pp. 321–322, seattle, Washington.

[20] J. Gong and P. Tarasewich, "Alphabetically Constrained Keypad Designs for Text Entry on Mobile Devices," in *Proc. of CHI '05*. NY, USA: ACM, 2005, pp. 211–220, portland, Oregon, USA.

[21] M. Zhong, C. Yu, Q. Wang, X. Xu, and Y. Shi, "ForceBoard: Subtle Text Entry Leveraging Pressure," in *Proc. of CHI '18*. NY, USA: ACM, 2018, pp. 528:1–528:10, montreal QC, Canada.

[22] M. B I Raez, M. S Hussain, and F. Mohd-Yasin, "Techniques of emg signal analysis: Detection, processing, classification and applications," *Biological procedures online*, vol. 8, pp. 11–35, 02 2006.

[23] K. A. Shatilov, D. Chatzopoulos, A. W. T. Hang, and P. Hui, "Using Deep Learning and Mobile Offloading to Control a 3d-printed Prosthetic Hand," *Proc. of IMWUT*, vol. 3, no. 3, pp. 102:1–102:19, Sep. 2019.