BoldMove: Enabling IoT Device Control on Ubiquitous Touch Interfaces by Semantic Mapping and Sequential Selection

Tengxiang Zhang
Beijing Key Laboratory of Mobile Computing and Pervasive Device
Institute of Computing Technology, Chinese Academy of Sciences, UCAS
Beijing, China

Xin Zeng
Beijing Key Laboratory of Mobile Computing and Pervasive Device
Institute of Computing Technology, Chinese Academy of Sciences, UCAS
Beijing, China

Yinshuai Zhang
Dingdao Intelligent Technology
Beijing, China

Xin Jiang
Lenovo Research
Beijing, China

Xuhai Xu
Information School
University of Washington, Seattle
Seattle, Washington, United States

Anind K Dey
Information School
University of Washington, Seattle
Seattle, Washington, United States

Yiqiang Chen
Beijing Key Laboratory of Mobile Computing and Pervasive Device
Institute of Computing Technology, Chinese Academy of Sciences, UCAS
Beijing, China

ABSTRACT

Recent advances in ultra-low-power ubiquitous touch interfaces make touch inputs possible anytime, anywhere. However, their functions are usually pre-determined, *i.e.*, one button is only associated with one fixed function. *BoldMove* enables spontaneous and efficient association of touch inputs and IoT device functions with semantic-based function filtering and a wait-confirm sequential selection strategy. In this way, such touch interfaces become ubiquitous IoT device controllers. We proposed the semantic-based IoT function filtering to improve control efficiency, then designed the sequential selection mechanism for interfaces with constrained input and output resources. We implemented BoldMove on a custom-built touch interface with capacitive button inputs and a smartwatch display. We then conducted a user study to determine the design parameters for the sequential selection method. At last, we validated that BoldMove only takes 3.25 seconds to complete a selection task if the target function appears within the Top-3 displayed item. Even if the assumption is relaxed to Top-10, BoldMove is still estimated to be more efficient than the conventional selection method with device-based filtering and menu-navigated selection.

CHI '22 Extended Abstracts, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9156-6/22/04…$15.00
https://doi.org/10.1145/3491101.3519805

CCS CONCEPTS

• **Human-centered computing → Interaction design process and methods**; *Gestural input*.

KEYWORDS

Semantics, touch interface, sequential selection, IoT control

ACM Reference Format:
Tengxiang Zhang, Xin Zeng, Yinshuai Zhang, Xin Jiang, Xuhai Xu, Anind K Dey, and Yiqiang Chen. 2022. BoldMove: Enabling IoT Device Control on Ubiquitous Touch Interfaces by Semantic Mapping and Sequential Selection. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3491101.3519805

1 BACKGROUND AND INTRODUCTION

Ubiquitous deployed interfaces enable users to interact with the digital world anytime, anywhere. Users can click or slide on various surfaces for input, such as objects [7, 11, 16–18], hands [15, 19], and fingers [8, 9]. Recent advances in ultra-low-power touch interfaces reduce the power consumption to several milli-Watts [13] and even micro-Watts[1]. Such extremely low power consumption enables them to be continuously powered by ambient energy sources (*e.g.*, in-door light, radio waves), thus eliminating energy maintenance efforts and being suitable for ubiquitous deployment. Compared with voice control interfaces, controlling IoT devices using ubiquitous touch interfaces is subtler and more privacy-preserving. This is especially true under scenarios when voice commands are intrusive, socially awkward, or privacy-invasive.

However, such ubiquitous touch interfaces only support input from a few buttons due to power and cost constraints. For example, a single passive BitID [16] sensor only supports button-like

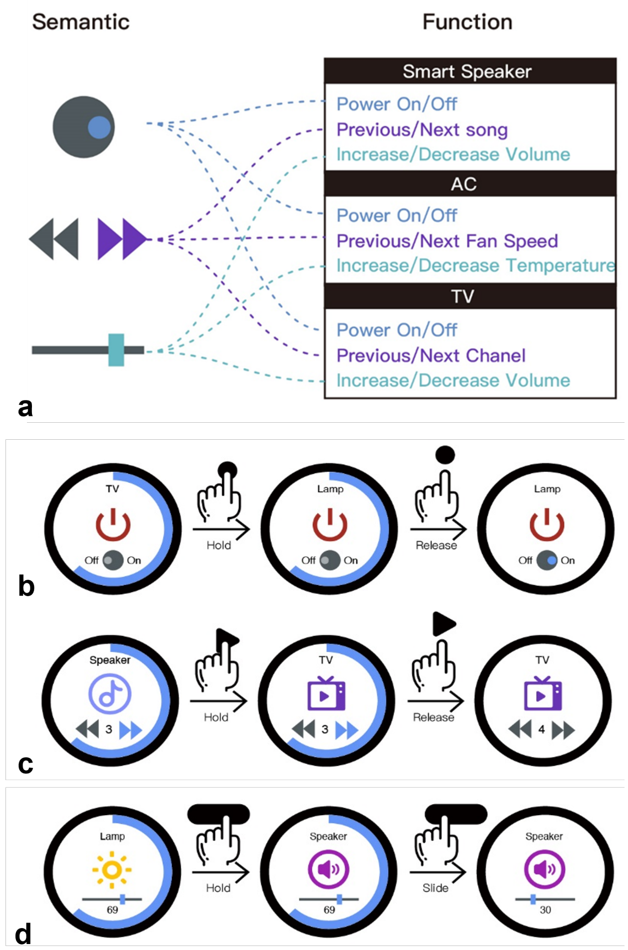

Figure 1: The mapping of three semantics (a) and their corresponding interaction design for *Toggle* (b), *Switch* (c), and *Adjust* (d).

binary inputs. The functions of such buttons are usually fixed to control particularly pre-determined objects like light bulbs. Also, they either use a local screen with a low resolution and refresh rate (*e.g.*, E-ink [21]) or reuse a remote display of a nearby device (*e.g.*, smartwatch screen) to save power. The limited input and output make it exceptionally inefficient to control multiple different IoT devices on such interfaces: The small and slow screen can only display one or two functions simultaneously. To make it worse, the input lag of simple buttons reduces the interaction efficiency, especially when users need to navigate among IoT device functions through many clicks.

To unleash the potential of the above mentioned ubiquitous interfaces for IoT device control, we propose *BoldMove*, a 'one-click' IoT device control method that enables spontaneous and efficient control on resource-constrained touch interfaces. BoldMove first filters IoT functions by mapping the semantics of both the function and the touch widget, then adopts a wait-confirm sequential selection mechanism to select from the matching functions. Inspired by Mayer et al. [10], we define three semantics in this paper (Figure 1a): *Toggle*, *Switch*, *Adjust*, which correspond to three categories of functions: those that only have binary states (*e.g.*, power on/off), those have several discreet states (*e.g.*, AC modes), and those have continuous states (*e.g.*, lamp brightness). Three simple touch widgets can represent the semantics: a button to toggle, two arrow buttons to switch, and a slider to adjust.

Figure 1 shows the control procedure. Users first express the semantic by 'boldly' pressing one of the three touch widgets before any function is shown. Only functions matching the widget semantic are displayed for further selection. The user holds the press to display the remaining functions one after one on a small screen. When the target function appears, the user either lifts (Figure 1b-c) or starts sliding the finger (Figure 1d) to confirm the selection.

BoldMove achieves 'one-click' IoT device control by trading spatial movements to temporal dwells, which significantly reduces the sensing resource demands of the interface. The 'one-click' wait-confirm selection strategy avoids input lags introduced by multiple clicks, which can be large on ubiquitous touch interfaces. The efficiency of the wait-confirm selection is further improved by the command-oriented [5] semantic-based function filtering, which can be easily expressed with extremely limited input resources. The traditional object-oriented device-based filtering (*i.e.*, select the device first, then select the target function), on the other hand, require advanced hardware like IR transceivers, cameras [3], or speakers/microphones [2] for device selection. They cannot be efficiently input on such interfaces, which we will show in later studies. Moreover, the interaction mechanism of BoldMove is also compatible with the traditional interaction mechanism – 'multiple-click' device control – with appropriate temporal threshold design, *e.g.*, quick click for navigation/confirmation and long click for semantic selection. As an initial exploration of this new technique, we first focus on the 'one-click' version of BoldMove in this paper.

We built a Bluetooth-compatible touch interface to validate the concept. The touch interface encodes capacitive button inputs into Bluetooth advertisements, which are received by a smartwatch worn on the interacting hand. The smartwatch's screen then displays functions based on the inputs. We built such a high-fidelity prototype to mimic real ubiquitous interfaces. The input of such interfaces usually lacks proper haptic feedback; the display is small or shared with another device opportunistically; the lag between input and display is large and noticeable. In this way, the results of this paper can be readily generalized to other ubiquitous touch interfaces with similar inputs and displays (*e.g.*, E-ink screen).

We first explain the design of our touch interface prototype in detail, then provide an analytical model of the sequential selection time. We conduct a user study to determine the design parameters of the sequential selection, which show that users prefer the target item to appear within the Top 3 items with a 2 seconds item display duration. We then evaluate the design under three scenarios of IoT device control. We implement the traditional menu-navigated selection method for a direct comparison despite not being suitable for such resource-constrained interfaces. The results show that BoldMove only took 3.25 seconds on average to select a function if it appears within the first three displayed items. Without the assumption, BoldMove is estimated to take 5.25 seconds on average, which is still much shorter than that of the conventional method (10.22 seconds). Analytical time estimation shows that BoldMove still outperforms the menu-navigated method as long as the target function is within the first ten items. In the end, we discuss the possibilities for BoldMove to scale for more functions and generalize to more input modalities.

The contributions of the article can be summarized as follows:

(1) A novel IoT device control method that combines semantic-based IoT function filtering with sequential selection to achieve efficient and spontaneous control on resource-constrained ubiquitous interfaces;
(2) A high-fidelity resource-constrained touch interface prototype system with capacitive touch buttons as inputs and a commercial smartwatch screen as output;
(3) An analytical model to estimate sequential selection time with respect to the appearance order of the target function;
(4) Two user studies to determine BoldMove design parameters and validate its performance under different scenarios.

Together, they make IoT device control possible on ubiquitous touch interfaces.

2 TOUCH INTERFACE DESIGN AND IMPLEMENTATION

We built a ubiquitous interface with capacitive touch inputs and a smartwatch display. The input side consists of a Bluetooth transceiver module (Ebyte E73-2G4M04S1B with an NRF52832 chip inside) and copper-made touchpads. The touchpads are connected with the analog input pins of the module for on-chip self-capacitive sensing. In this way, they can be easily deployed on surfaces of different objects. The *Toggle* semantic is mapped to a circular button, the *Switch* semantics are mapped to two arrow buttons that point left and right respectively, and the *Adjust* semantic is mapped to a 4-element slider (see Figure 2). A CR2032 coin battery powers up the whole interface. An average current of 0.3 mA is estimated using the the Nordic Online Power Profiler[1], which translates to a working lifetime of 700 hours when using a 210mAh coin battery.

The module sends out Bluetooth advertisement packets every 20ms to communicate touch events with a commercial smartwatch. The smartwatch continuously scans for the advertisement and decodes the touch status of each widget. The functions are matched and displayed on the smartwatch screen accordingly. As an example, we implement the display of our system using the screen of a smartwatch in our prototype. Our touch interface can also be used with other Bluetooth-compatible displays like TV, laptop, smartphone, and smart speaker screens. This makes the interface highly flexible in terms of deployment locations and usage scenarios.

Aside from the 20ms advertisement interval, the self-capacitive touch detection, the smartwatch processing and display refresh, and the user's reaction time of input confirmation from the display changes will also introduce additional input lags. The input lag would be even larger for low-power E-ink displays since the refresh time is usually more than 400ms.

[1] https://devzone.nordicsemi.com/power/w/opp/2/online-power-profiler-for-bluetooth-le

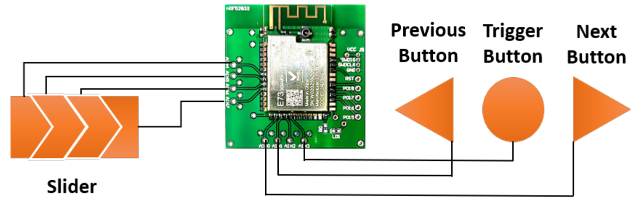

Figure 2: Our prototype touch interface uses a nRF52832 Bluetooth module and is powered by a CR2032 battery.

3 THEORETICAL ANALYSIS FOR SEQUENTIAL SELECTION TIME

The sequential selection time consists of hold time (*i.e.*, waiting for the target function to appear) and reaction time (*i.e.*, selecting the function when it appears). The system determines the hold time (*i.e.*, the appearance sequence) while the user determines the reaction time. If the target appears at the i^{th} item during a task, the function selection time for the task is

$$T_i = t_d(i-1) + t_r \tag{1}$$

in which t_d is the display duration for each item, and t_r is the reaction time. We make two assumptions: 1) t_r does not change with i; 2) the target function will appear within the first n displayed items with an equal probability. Then the expectation of the selection time is

$$E[T] = \sum_{i=1}^{n} \frac{(i-1)t_d}{n} + t_r = \frac{(n-1)t_d}{2} + t_r \tag{2}$$

We can see **the expected selection time will increase by** $t_d/2$ **if** n **increase by one.** It is possible to predict the probability distribution of the semantic-matching functions based on relevant context (*e.g.*, user orientation, location) and control history (*e.g.*, previous task). The target function will then have a higher probability of appearing earlier. So the uniform probability distribution assumption in Equation 2 makes the estimated time the **upper bound** of the selection time, which indicates the selection time of BoldMove is determined by t_d, t_r, and n.

So we first conduct User Study 1 to determine the preferred t_d and acceptable n and understand different configurations' impacts on t_r. The study results show that users prefer the display duration to be 2 seconds and the target function appears within the Top-3 items. This configuration is validated in User Study 2, which shows that BoldMove outperforms the conventional device-based menu-navigated control method on our prototype interface.

4 USER STUDY 1

We design four simple control tasks for the study: 1. Turn on the TV; 2. Turn off all lights; 3. Switch to the next TV channel; 4. Decrease the TV volume. The four tasks correspond to toggle one device, toggle multiple devices, switch on one device, and adjust on one device. We had two experimental settings: 1) The target function will appear as the 1^{st}, 3^{rd}, 5^{th} item (*i.e.*, $i = 1, 3, 5$); 2) The display duration for each item is set to 1s, 2s, and 3s (*i.e.*, $t_d = 1, 2, 3s$). This makes a total of nine display configurations for each task.

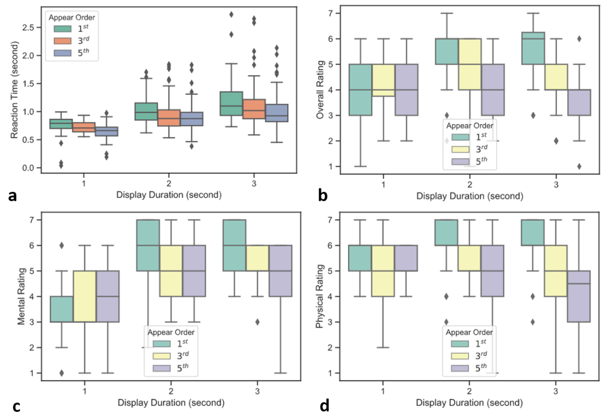

Figure 3: Boxplot of Reaction time vs. display duration (a). Boxplots for overall (b), mental (c), and physical (d) ratings under different display configurations.

We recruited 8 participants (4 Females) from a local institution with ages ranging from 23 to 31 (M=26.6, SD=3.0). They all had smart home control experiences. All participants were asked to sanitize their hands before and after the experiment. All participants wore the smartwatch on their right hand and touched the button using the same hand. Each participant completed two repetitive sessions of the experiment. Each session had four blocks, each corresponding to one control task; each block had nine trials, each corresponding to one display configuration. After each trial, the user rated each trial's overall preference ("highly undesirable" to "highly undesirable"), mental, and physical efforts ("Very High" to "Very Low") on a 7-point Likert scale (the higher the better). The sequence of blocks and trials are randomized.

4.1 Results Analysis

There is a total of 8 participants × 2 Sessions × 4 Blocks × 9 Trials = 576 trials, 64 for each display configuration. Surprisingly, Friedman test results indicate that different types of control tasks do not significantly impact overall preference ratings (Figure 3b). We speculate that our prototype requires similar human efforts (touch and release from the widget) when completing different types of control tasks.

The median reaction time (Figure 3a) is 0.7 seconds, 0.9 seconds, and 1 second for $t_d = 1, 2, 3$ *seconds* respectively. Two-way RM-ANOVA results with Greenhouse-Geisser correction show that **the reaction time for later appeared target function is significantly smaller** ($F_{2,14} = 13.0, p < .01$). For example, participants select 6.2% ($t_d = 1s$), 9.4%($t_d = 2s$), and 7.0%($t_d = 2s$) faster for $i = 3$ compared to those when $i = 1$. Variance analysis of Aligned Rank Transformed data shows that both the appear order ($F_{2,56} = 6.68, p < .01$) and display duration ($F_{2,56} = 17.2, p < .001$) have significant impacts on subjective ratings. Specifically, the display duration has a significant effect on the mental effort rating ($F_{2,56} = 40.5, p < .001$), but not on the physical effort ($F_{2,56} = 1.83, p = .17$); the appearance order (how many items need to be viewed before the correct item is displayed), on the other hand, has a significant effect on the physical effort rating ($F_{2,56} = 20.1, p < .001$), but not on the mental effort ($F_{2,56} = 1.04, p = .36$). The results indicate that **mental efforts are mainly due to the limited reaction time, while the physical efforts are mainly due to the extended finger pressing period**. The participants felt that a 1-second display duration is too short of selecting, especially when the target item appears as the first item. They are neutral (MEDIAN = 4) for the 1-second display duration for all three appearance orders. Obviously, they preferred the target function to appear as early as possible. For a more practical evaluation, we set the target function to appear within the Top 3 items with a display duration of 2 seconds because such a configuration still has a positive overall rating (MEDIAN = 5).

5 USER STUDY 2: EVALUATION

We evaluated our prototype system under three scenarios: 1) Working, the user presents slides on a laptop with colleagues on the phone. The touch interface is deployed on the tabletop; 2) Reading, the user turns off the TV and starts to read a book in the living room. Two touch interfaces are deployed on a coffee cup and a book's back cover; 3) Returning Home, the user completes a daily routine after returning home. The touch interface is deployed on the user's palm. Table 1 shows the detailed devices, functions, and task sequences. Aside from BoldMove, we also implemented a conventional IoT control method with device-based filtering and menu-navigated selection as a baseline. The user navigates through the device and function menus using the two switch buttons and confirms selection using the toggle button. An example is shown in Figure 4a. Even though such a method is not efficient on our interface, we were still interested to understand the quantitative differences between the two methods.

We recruited 7 participants from a technology company. Their ages range from 19 to 32 (M=27.9, SD=5.71). They all had IoT device control experiences. The study lasted about 1 hour. Each participant was compensated with two $7 coffee coupons. To achieve a more realistic control experience, an experimenter observed user behavior and controlled the devices using remote controllers. All participants were asked to sanitize their hands before and after the experiment.

Figure 4b-d shows the experiment settings. For each scenario, the participant was asked to complete the same task sequence twice using BoldMove and the menu-based selection method respectively. The sequence of the scenarios was randomized. After each scenario, the participant rated BoldMove in terms of mental and physical efforts, as well as overall experience using a 7-point Likert scale (the higher, the better).

5.1 Results Analysis

We collected data for $7 \ participants \times 3 \ scenarios \times 7 \ tasks = 147$ tasks. The overall average selection time for each task across the three scenarios is 3.25 seconds (MEDIAN = 3.10 seconds, SD = 2.34 seconds). Note that this is with a strong assumption that the target function appears within the Top-3 items ($n = 3$). n is determined by the number of matched functions for each semantic. In this study, $n_{max} = 5$. Based on the analysis in Section 3, the maximal estimate task selection time $T_{max} = 3.25 + 2 \times t_d/2 =$

Table 1: The IoT devices, their corresponding functions, and task sequences for the three tested scenarios.

Devices and Functions	Task Sequence
Scenario 1: Working **Laptop**: *Toggle Sleep*, *Switch Slide*, *Toggle Mute*, *Adjust Brightness*, *Adjust Volume*, *Switch Desktop*; **Conference Phone**: *Toggle Answer*, *Toggle Mute*, *Toggle Redial*, *Adjust Volume.*	1 *Redial* the phone. 2 *Decrease Volume* of the phone. 3 *Unmute Microphone* of the phone. 4 *Increase Volume* of the laptop. 5 Turn to *Previous Slide* 1 time on the laptop. 6 Turn to *Next Slide* 1 time on the laptop. 7 *Mute Speaker* the laptop. 8 *Hang Up* the phone.
Scenario 2: Reading **TV**: *Toggle Power*, *Switch Episode*, *Adjust Volume*; **Lamp**: *Toggle Power*, *Switch Mode*, *Adjust Brightness*; **Smart Speaker**: *Toggle Play*, *Switch Song*, *Adjust Volume.*	1 *Previous Channel* from the cup. 2 *Power Off* TV from the cup. 3 *Power On* Lamp from the cup. 4 *Increase Lightness* of Lamp from the book. 5 *Start Play* of the smart speaker from the book. 6 *Next song* from the book. 7 *Decrease Volume* of the speaker from the book. 8 *Power Off* Lamp from the book.
Scenario 3: Returning Home **Light**: *Toggle Power*, *Adjust Brightness*; **Curtain**: *Toggle Open*, *Adjust Position*; **TV**: *Toggle Power*, *Adjust Volume*; **AC**: *Toggle Power*, *Switch Fan Speed*; **Air Purifier**: *Toggle Power*, *Switch Fan Speed*	1 *Power On* the light. 2 *Close* the curtain. 3 *Switch to Next Fan Speed* of the Air Purifier. 4 *Switch to Previous Fan Speed* of the AC. 5 *Open* the TV. 6 *Increase Volume* of the TV. 7 *Decrease Lightness* of the Light. 8 *Close* the TV.

5.25 seconds. This is still much smaller than that of the baseline method, which consumed 10.22 seconds on average (MEDIAN = 8.67 seconds, SD=5.66 seconds) to complete a task. We suspect such a difference is mainly due to two reasons: **1) The semantic-based filtering happens *in-situ* with the finger press while the device-based flitering requires explicit device selection; 2) The menu-navigated selection methods involve too many button clicks while BoldMove only requries one "long click". Each click will introduce noticable lagging because of touch sensing, data communication, and display feedback on such a resource-constrained touch interface.**

The subjective ratings are shown in Figure 4e. We can see that ratings of BoldMove improve with the number of devices. Three participants complained that the baseline method involves too many clicks. P4 said "*The new method (*BoldMove*) is intuitive and straightforward. The many clicks of the conventional method suddenly feel redundant.*" Two participants mentioned that BoldMove demands higher mental effort since the countdown timer made them nervous. P6 (a machine learning engineer) commented that it should be easy to implement a recommendation algorithm for BoldMove since "*there is more information*" and "*the optimization goal is clearly defined*".

6 DISCUSSION, LIMITATION, AND FUTURE WORK

Study 1 shows that users are reluctant to use BoldMove when the target appears after the first three items. However, BoldMove's scalability has great room for improvement. Fisrt, the user can combine semantic filtering with device filtering of functions to further reduce items for selection. For example, the user can use BoldMove on the touch interface of the target device so that only the device's functions are included for later selection. The remaining function can then be fewer than three; Second, a recommendation algorithm can be implemented based on the context. For example, the system can sort the appearance order based on selection possibilities. In this way, it becomes possible that the target function is within the Top-3 recommendations. At last, previous research [20] has shown that it is possible to design a more efficient touch interface to improve the input efficiency of the wait-confirm selection strategy. Development and evaluation of these methods to improve BoldMove's scalability are beyond the scope of this paper, which we plan to explore in the future.

The underlying concept of BoldMove can be readily applied to input modalities other than touch, as long as it can express the three basic types of semantics with a wait-confirm selection strategy. It is especially useful for input modalities that can easily express input semantics but cannot easily indicate devices and functions, such as interaction using teeth [14], feet [12], and hair [4].

One of the limitations of this work is that the experiments were conducted in a controlled lab environment. BoldMove is also suitible for public smart spaces where ultra-low-power ubiquitous control is a better fit. So we plan to conduct in-the-wild outdoor user studies to understand user needs and better evaluate the concept in real usage settings. We also plan to validate BoldMove's performance on a self-powered touch interface with a local E-ink display [6]. For touch

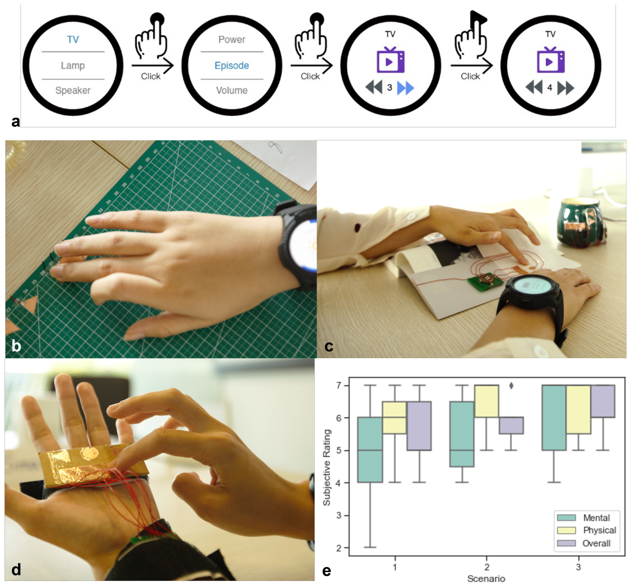

Figure 4: (a) IoT control procedure using our ubiquitous interface with device-based filtering and menu navigation. (b) Scenario 1: Users issue commands using a touch interface on the table; (c) Scenario 2: Users issue commands using two touch interfaces, one on a cup and the other on a book; (d) Scenario 3: Users issue commands using an on-body touch interface; (e) Mental, physical, and overall user rating of BoldMove

interfaces with moderate input lag, the 'multiple-click' selection strategy (long click on buttons for semantic selection, short click on left/right arrow buttons to navigate through candidate functions, and on the toggle button to confirm) can be more efficient. We plan to conduct studies to understand the user's preference for the passive wait-confirm 'one-click' strategy and the active navigate-confirm 'multiple-click' strategy regarding input delay. Another interesting research topic is the theoretical analysis of the semantic-based filtering method based on information entropy.

7 CONCLUSION

BoldMove is a novel IoT device control interaction design for resource-constrained touch interfaces. It leverages *in-situ* semantic-based function filtering to improve selection efficiency. It uses a 'one-click' wait-confirm sequential selection strategy to avoid lags due to sensing, communication, and feedback for each click operation. The results of User Study 1 show that users prefer the target function to appear within the first three items with a display duration of 2 seconds. We then implemented this configuration and evaluated our system in User Study 2. BoldMove outperforms the conventional control method with device-based function filtering and menu-navigated selection on such a resource-constrained interface. We argue that BoldMove can be readily applied to other input modalities, especially those with limited input ability.

ACKNOWLEDGMENTS

The work is supported by National Key Research and Development Plan under Grant No. 2020YFC200400, No.2019YFB140070, National Natural Science Foundation of China under Grant No. 6210240, and Beijing Key Laboratory of Mobile Computing and Pervasive Device.

REFERENCES

[1] Nivedita Arora, Ali Mirzazadeh, Injoo Moon, Charles Ramey, Yuhui Zhao, Daniela C. Rodriguez, Gregory D. Abowd, and Thad Starner. 2021. MARS: Nano-Power Battery-free Wireless Interfaces for Touch, Swipe and Speech Input. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, New York, NY, USA, 1305–1325. https://doi.org/10.1145/3472749.3474823

[2] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. 2013. DopLink: Using the Doppler Effect for Multi-device Interaction. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 583–586. https://doi.org/10/gfsvhm

[3] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5909–5920. https://doi.org/10/gg4nkk

[4] Christine Dierk, Sarah Sterman, Molly Jane Pearce Nicholas, and Eric Paulos. 2018. HairIO: Human Hair as Interactive Material. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '18)*. Association for Computing Machinery, New York, NY, USA, 148–157. https://doi.org/10/gmg92j

[5] Alix Goguey, Julie Wagner, and Géry Casiez. 2015. Quantifying Object- and Command-Oriented Interaction. In *Human-Computer Interaction – INTERACT 2015*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.), Vol. 9299. Springer International Publishing, Cham, 231–239. https://doi.org/10.1007/978-3-319-22723-8_18 Series Title: Lecture Notes in Computer Science.

[6] Tobias Grosse-Puppendahl, Steve Hodges, Nicholas Chen, John Helmes, Stuart Taylor, James Scott, Josh Fromm, and David Sweeney. 2016. Exploring the Design Space for Energy-Harvesting Situated Displays. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10/br3q

[7] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press, Santa Barbara, California, USA, 441. https://doi.org/10/b8sdk6

[8] Hsin-Liu (Cindy) Kao, Artem Dementyev, Joseph A. Paradiso, and Chris Schmandt. 2015. NailO: Fingernails as an Input Surface. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3015–3018. https://doi.org/10/cpb9

[9] Keiko Katsuragawa, Ju Wang, Ziyang Shan, Ningshan Ouyang, Omid Abari, and Daniel Vogel. 2019. Tip-Tap: Battery-free Discrete 2D Fingertip Input. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1045–1057. https://doi.org/10/gg7kqr

[10] Simon Mayer, Andreas Tschofen, Anind K. Dey, and Friedemann Mattern. 2014. User interfaces for smart things – A generative approach with semantic interaction descriptions. *ACM Transactions on Computer-Human Interaction* 21, 2 (Feb. 2014), 1–25. https://doi.org/10/gfs78x

[11] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. 2012. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. ACM Press, Austin, Texas, USA, 483. https://doi.org/10/gg5tzd

[12] Eduardo Velloso, Dominik Schmidt, Jason Alexander, Hans Gellersen, and Andreas Bulling. 2015. The Feet in Human–Computer Interaction: A Survey of Foot-Based Interaction. *Comput. Surveys* 48, 2 (Sept. 2015), 21:1–21:35. https://doi.org/10/gmg92d

[13] Anandghan Waghmare, Qiuyue Xue, Dingtian Zhang, Yuhui Zhao, Shivan Mittal, Nivedita Arora, Ceara Byrne, Thad Starner, and Gregory D Abowd. 2020. UbiquiTouch: Self Sustaining Ubiquitous Touch Interfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (March 2020), 27:1–27:22. https://doi.org/10/ggpb3k 00000.

[14] Xuhai Xu, Chun Yu, Anind K. Dey, and Jennifer Mankoff. 2019. Clench Interface: Novel Biting Input Techniques. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300505

[15] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Ruichen Meng, Sumeet Jain, Yizeng Han, Xinyu Li, Kenneth Cunefare, Thomas Ploetz, Thad Starner, Omer Inan, and Gregory D. Abowd. 2018. FingerPing: Recognizing Fine-grained Hand Poses Using Active Acoustic On-body Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 437:1–437:10. https://doi.org/10/gg7kqn

[16] T. Zhang, N. Becker, Y. Wang, Y. Zhou, and Y. Shi. 2017. BitID: Easily Add Battery-Free Wireless Sensors to Everyday Objects. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. 1–8. https://doi.org/10/gg4nkm

[17] Tengxiang Zhang, Xin Zeng, Yinshuai Zhang, Ke Sun, Yuntao Wang, and Yiqiang Chen. 2020. ThermalRing: Gesture and Tag Inputs Enabled by a Thermal Imaging Smart Ring. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, Honolulu, HI, USA, 1–13. https://doi.org/10/gg7k4j 00000.

[18] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. Electrick: Low-Cost Touch Sensing Using Electric Field Tomography. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 1–14. https://doi.org/10/gg4nkg

[19] Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. SkinTrack: Using the Body As an Electrical Waveguide for Continuous Finger Tracking on the Skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1491–1503. https://doi.org/10/gg7kqj

[20] Shengdong Zhao, Pierre Dragicevic, Mark Chignell, Ravin Balakrishnan, and Patrick Baudisch. 2007. Earpod: eyes-free menu selection using touch input and reactive audio feedback. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. Association for Computing Machinery, New York, NY, USA, 1395–1404. https://doi.org/10.1145/1240624.1240836

[21] Y. Zhao, J. R. Smith, and A. Sample. 2015. NFC-WISP: A sensing and computationally enhanced near-field RFID platform. In *2015 IEEE International Conference on RFID (RFID)*. 174–181. https://doi.org/10/gn24rq