CountMarks: Multi-finger Marking Menus for Mobile Interaction with Head-Mounted Displays

Jordan Pollock[1] and Robert J. Teather[2(✉)]

[1] School of Computer Science, Carleton University, Ottawa, ON, Canada
[2] School of Information Technology, Carleton University, Ottawa, ON, Canada
rob.teather@carleton.ca

Abstract. We designed, implemented, and evaluated CountMarks, a novel menu that extends marking menus with multi-touch input on a secondary touchscreen, for use with smart glasses and head-mounted displays. We experimentally compared CountMarks to marking menus. Results indicate that CountMarks offers faster selection and better search accuracy, but slightly worse selection accuracy. A second experiment compared standing vs. walking while using CountMarks, and handheld vs. on-leg interaction. Results indicate that CountMarks can be used effectively with a handheld device while both standing and walking. We present an example application employing CountMarks in a mock Netflix UI, to demonstrate how the technique can be applied in existing applications.

Keywords: Mobile VR · Smart glasses · Marking menus · Multi-touch

1 Introduction

Low-cost self-contained head-mounted displays (HMDs) are now a reality and are advancing to more closely resemble everyday eyewear. Smart glasses give the user supplementary information while performing a task; the user need not move their head to look away from their main task. Because such workflows typically engage the hands, the user generally does not have access to direct touch interaction with virtual elements and data. Current methods for interacting with HMDs include handheld devices ("wands") like those with the Oculus Rift or HTC Vive, or combinations of speech and gestures, like with the HoloLens. These can be impractical: speech recognition is inappropriate for public use [29] and in air gestures (including wands) are prone to arm fatigue [12] and gesture recognition problems [8]. Handheld devices introduce mapping problems and cumbersome interaction [28].

Interaction techniques should be designed in consideration of the environment in which they will be used. Smart glasses may exacerbate the problems associated with distracted device usage[1] as they provide ever-present, information-rich displays. Consequently, interaction with such devices should be designed considering everyday usage

[1] For examples, see, e.g., https://www.nhtsa.gov/risky-driving/distracted-driving or https://news.osu.edu/distracted-walking-injuries-soar-for-pedestrians-on-phones/.

C. Stephanidis et al. (Eds.): HCII 2020, LNCS 12424, pp. 241–260, 2020.
https://doi.org/10.1007/978-3-030-60117-1_18

scenarios, such as usage while sitting, standing, or walking [22]. Ideally, the technology should be unobtrusive [34], be socially acceptable, and performant. To satisfy these requirements we first developed CountMarks, a multi-touch marking menu for selection with mobile HMDs (see Fig. 1).

Fig. 1. From left to right showing a) a user places a finger on the screen to open the first menu. b) Placing two fingers on the screen to open the 2nd menu. c) Swiping right with two fingers to select the target.

CountMarks extends marking menus [18] (see Fig. 2), with multi-touch swipe gestures on touch devices (e.g., smartphones), providing more menu items in fewer submenu layers than marking menus. Our approach combines marking menus with Count Menus [1] (Fig. 2b), which count the fingers used on one hand to select a specified linear menu. The finger count on the other hand then determines which item is selected from the open menu. CountMarks combines these two strategies, employing marking menu gestures with finger count. We envision using CountMarks with HMDs, smart glasses, or external displays (e.g., TVs). CountMarks uses the number of fingers touching the screen during swipe to determine which of up to four options is selected. We use a mode changing double tap to provide additional menu options.

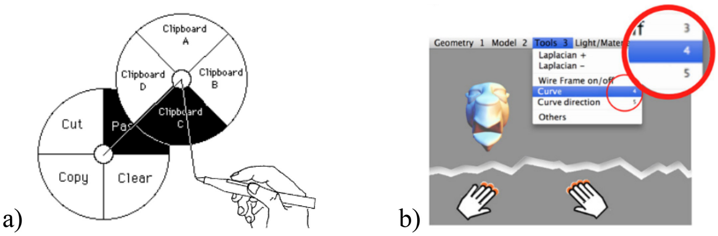

Fig. 2. a) Marking menu (via Kurtenbach, 1993 [18]); b) Count menu (Bailly et al. 2012 [1])

We discuss the design of CountMarks, and then present two user studies. The first study compares two CountMarks variants to classic marking menus and demonstrates the effectiveness of our approach. The second study evaluates one CountMarks variant in standing and walking use cases. We then present an example CountMarks implementation on an existing interface. Finally, we discuss strengths and weaknesses of using CountMarks and suggest future work to improve its design.

To our knowledge, our experiments are the first evaluations of marking menu interaction in mobile use cases and with input at the leg, despite previous research on marking menus designed for smartphone use [40].

2 Related Work

2.1 Marking Menus and Count Menus

Marking menus [18] (see Fig. 2a) are swipe-gesture based menus that are quicker and more accurate than typical linear menu selection [2]. Modern marking menus show that multiple individual strokes improve selection speed and accuracy [37]. The primary limitation of marking menus is that they only support up to 8 items per menu and two levels of depth to retain 90% selection accuracy [17]. In general, broad menus (i.e., menus with more items in a single menu level) perform better than deep menus (i.e., menus requiring drilling down into a hierarchy, increasing the number of selections required to select an item) because users tend to get lost in deep menu hierarchies [14]. As a result, variations of marking menus have been created to increase breadth: the flower menu [3] which uses curved instead of linear strokes, and the zone and polygon menus [38] which let users make different selections based on where their swipe gesture begins. These variations demonstrate that greater breadth can increase selection speed and/or accuracy.

Marking menus help transition users from novice to expert [18]. Novices search the menu hierarchy to find the desired target. In contrast, experienced users may already know how to find or access a target item and may use gesture shortcuts to select it. Much like other shortcuts, these gestures must first be learned by the user. Due to the effort required to learn them, Expert shortcuts are rarely used [19], even by experienced users who know about shortcuts [7].

Lepinski et al. [20] first explored multi-touch input for marking menus as a way to increase menu breadth. By holding different combinations of fingers (chords) to a touch surface different menus popup which the user can swipe on to make selections on a large touch surface. While most chords were difficult to perform, a simple subset were shown to offer faster selections than traditional marking menus.

Finger Count menus [1] (Fig. 2b) let the user select menus and items according to the number of fingers used in a gesture, regardless of which fingers are used. This technique uses two hands: one hand to select one of 5 menus and the other hand to select one of 5 items from that menu. Count Menus were found to be simple to use and easy to learn [4] and twice as fast as 3D, in-air marking menus [24] while maintaining similar accuracy [16].

2.2 Interaction Techniques for HMDs and Smart Glasses

In-air gestures are an obvious choice for interacting with HMDs because they offer the freedom to interact with all of the space in front of the user at any time. However, gestures can be obtrusive, attention-grabbing and often unnatural or dissimilar to normal gestures which can make them potentially embarrassing to use in public settings [15, 23, 26].

Additionally, in-air gestures have the well-known problem of "gorilla arm syndrome" where gesture can yield excessive arm fatigue [12].

Researchers have sought to leverage the "anywhere, anytime" nature of gestural interaction while avoiding these limitations. One approach involves mounting depth trackers to the shoulders [10] or hips [21] to turn normal surfaces into touch surfaces, or to support subtle in-air gestures without raising the arm. These touch recognition techniques provide tactile feedback, improving gesture recognition accuracy and precision [6]. Other researchers have considered using touch on the body itself as an input source, for example, using the user's skin [11, 30, 31]. Skin supports a large number of gestures such as grab, pull, press, scratch shear, squeeze, and twist [31].

The palm is a logical choice for on-skin touch interaction [31]. Previous work has shown that multi-stroke palm gestures can be detected with 90% accuracy [30] but requires bulky accessories. While other research explored interactions on other body parts such as the ear [32, 39], there is little research on the leg as an input surface, despite previous work showing that 9% of gestures produced in gesture elicitation [31] were performed on the leg (vs. 51% on the hand, 12% on a ring, and 10% between fingers). Some researchers have looked at areas such as belts [9] and pockets [23, 25]. Smart textiles (e.g., a finger sleeve providing bend and pressure input) offer accuracy greater than 80% while walking, running and driving [35]. All these areas provide possible interaction surfaces for the implementation of CountMarks.

3 Design and Implementation of CountMarks

CountMarks extends marking menus [18] by detecting how many fingers are placed on a secondary touchscreen. This allows up to 4 times the number of selections possible in each swipe direction. This leverages the benefits of both marking menus (speed, accuracy, and scale-invariant) and Count Menus (simple, intuitive, and the ability to "short-cut" parts of the menu) [4]. CountMarks supports the well-known "recognition over recall" principle better than marking menus: previous work [14] found that marking menu navigation can be difficult as users can get lost in the hierarchy. In contrast, broader menus, like CountMarks, offer better recognition [27].

Marking menus typically hold up to 8 menu items in a single menu level [18], or 64 items over two levels. In contrast, CountMarks can fit 32 items in a single menu level (four fingers, eight directions) and 1024 items over two menu levels. For a larger number of items, we added the ability to change modes via a double tap on the screen with one finger, to access another set of menus or items.

We designed CountMarks with smart glasses in mind, due to the challenges faced by their public and mobile nature. We describe these design considerations next.

3.1 Social Acceptability

Because of the public nature of smart glasses, we designed CountMarks to comply with Hsieh et al.'s [13] recommendations for social acceptably:

1. Isolate sensing technology from the glasses,
2. Use relative pointing for adapting to various postures,
3. Design small movements for subtle interaction,
4. Aim for intuitive gestures,
5. Enhance tangibility.

CountMarks is more subtle than in-air gestures as it uses small touch gestures on a peripheral surface. Noting the poor precision of in-air interaction due to the absence of tactile feedback [8], we employ CountMarks on a smartphone touch screen. This leverages hardware users already own, rather than requiring an additional device. Future implementations could instead use smart textiles, or skin-based input [11] providing users precise haptic feedback, while allowing them to perform gestures wherever is most comfortable. We predict this may be on the hand or the upper leg.

Like marking menus, CountMarks allows swipe gestures anywhere on the touch surface. This facilitates eyes-free interaction, as the user does not need to see the surface; the user simple swipes in the desired direction with the appropriate number of fingers. We now discuss two CountMarks variants, *MenuCount* and *ItemCount*.

3.2 CountMarks Variant #1: MenuCount

MenuCount presents different menus based on how many fingers touch the screen. Holding fingers on the screen opens and preview each menu. Changing the number of fingers changes which menu is displayed on the primary display (e.g., HMD). MenuCount is seen in Fig. 3 (left), where the user has activated the "Animals" menu by touching two fingers to the screen. The user swipes in the direction of the desired item with the same number of fingers used to open the menu. Finally, the arrow to the right of the menus indicates that a double tap mode change is available, and toggle 4 more menus.

Fig. 3. (Left) *MenuCount Variant*: The user presses two fingers to the touch surface; this opens the 2^{nd} menu. A two-finger swipe up would select the "pig" icon. (Right) *ItemCount Variant*: The user presses two fingers to the touch surface; this highlights all the 2^{nd} options in each sub-menu. A two-finger swipe up would select "fish" from the top menu.

3.3 CountMarks Variant #2: ItemCount

ItemCount presents a single large radial menu containing up to eight linear sub-menus in each of (up to) eight directions. See Fig. 3 (right). Each linear menu can display four items at once. Items are selected by touching the screen with the required number of fingers and swiping in the direction of that item's menu. When touching the screen, all selectable menu items corresponding to the finger count are highlight. For example, a single finger touching the screen highlights all top menu items, two fingers highlights all the second items, etc. Similar to MenuCount, a double-tap toggles another four menu items in each menu. With ItemCount, this is presented as ellipses at the bottom of each linear sub-menu, as seen in Fig. 3 (right).

3.4 Interaction Location

We designed both CountMarks variants to work on the human body because it is comfortable and easy for people to interact with [28]. Body-based input provides tactile feedback and leverages proprioception, allowing precise and well-coordinated interactions. Since reliable detection of on-body input is an on-going engineering challenge [11, 33, 36], we instead implemented our prototypes using a smartphone as a proof of concept. We considered two main interaction locations: the palm and leg.

Palm interaction is likely the most intuitive on-body input location since people naturally perform interactions on their palm [31]. Conveniently for the purpose of our study, it is also the most natural place to interact with a smartphone. We note potential limits of palm-based input: it necessitates the use of both hands for interaction, and is potentially fatiguing as users must hold out both hands. Thus, we also consider the upper leg as a less obvious, but potentially ergonomic and efficient location for interaction. With interaction occurring on the leg the user can rest their arms at their side, as they naturally do while standing, and perform interactions where their hand meets their leg. As an added benefit, this can be done one-handed.

We evaluated CountMarks in two experiments. The first experiment compared both CountMarks variants against traditional marking menus. The second experiment compared interaction locations (hand vs. leg), and mobility (standing vs. walking).

4 Experiment 1: Comparing CountMarks Variants

This experiment compared both CountMarks variants, ItemCount and MenuCount, to multi-stroke marking menus [37].

4.1 Participants

We recruited 18 participants (mean age of 25.95 years, SD = 6.2, all right-handed, 10 female). Participants were recruited via posters on campus and social media.

4.2 Apparatus

We used a Samsung Galaxy S8 smartphone (running Android 8.0) as the touch input device. The smartphone was secured to the desk 10 in. from the participant. A 23.5-in. BenQ 1920 × 1080p computer monitor was positioned 19 in. away from the participant (Fig. 4), connected to a PC with an Intel Core i7-7700K 4.20 GHz CPU and 32 GB of RAM running 64-bit Windows 10.

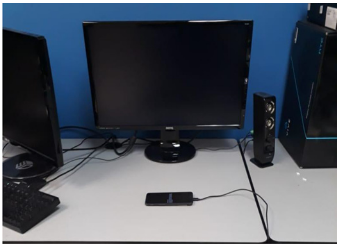

Fig. 4. Apparatus showing the position of the display and smartphone.

We developed a Unity Android application (Fig. 5) that controlled a desktop Unity app. Touch input detected by the smartphone was sent to the desktop app via Unity Remoting. The desktop monitor displayed the menus, rather than the smartphone screen which was used exclusively for input.

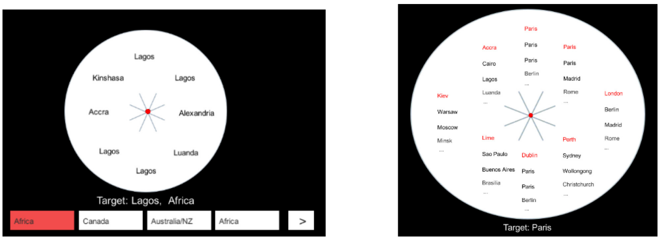

Fig. 5. Both CountMarks variants in their level 3, 8 × 8 configurations (left) MenuCount and (right) ItemCount.

The software presented a series of menus and menu items, with a target item listed at the bottom of the screen. Participants selected items from the menus via swiping gestures on the phone, as described earlier. The visual layout of each menu changed according to the menu style (see Fig. 5). The number of root menus and the number of items per menu changed according to difficulty level.

The software recorded time, and both correct and erroneous selections per trial. Execution time and number of fingers were recorded per selection.

4.3 Procedure

Participants were briefed on arrival, then provided informed consent. We then explained the task, which was based on Bailly et al. [5]. The task required selecting each instance of a specified target city name from all menus, organized by titles of countries and continents (Fig. 5). The menus were randomized so half of them included the target. Similarly, the menu items (cities) were randomized and in each target menu, the target item appeared as half of the menu items plus or minus one. This unpredictability ensured that participants could not count targets in each menu but would have to search for the targets to confirm none were left.

After a target item was correctly selected it disappeared from the menu leaving a blank space (or dashed lines for ItemCount). Trials started and ended when the participant double tapped the smartphone screen with two or more fingers. The first two trials of each block were practice trials. When participants believed they had found all target instances, they double tapped with two fingers to end the trial. Participants were instructed to complete each trial as quickly but accurately as possible.

Participants performed the task on 3 difficulty levels using each menu style: MenuCount, ItemCount and Marking Menu. They received a 3-min practice period each time they started a new menu style. Difficulty level was based on the number of menus, items per menu, and whether a mode change double-tap was required.

4.4 Design

Our experiment employed a 3 × 3 within-subjects design. The independent variables were menu style (ItemCount, MenuCount and Marking Menu) and difficulty level (levels 1, 2 and 3). Menu style was counterbalanced according to a Latin square. For each menu style, participants progressed through 3 levels of difficulty. See Table 1 for a summary of the conditions.

Table 1. Experiment conditions. Levels 1 and 2 of CountMarks variants are ordered by whether or not a mode change was required.

Difficulty Level	Menu style								
	Marking Menu			MenuCount			ItemCount		
	1	2	3	1	2	3	1	2	3
Menu items	4 × 8	8 × 4	8 × 8	4 × 8	8 × 4	8 × 8	8 × 4	4 × 8	8 × 8
Mode change	No	Yes	Yes	No	Yes	Yes	No	Yes	Yes

There were 7 trials with each difficulty level. Each trial consisted of 7 (for difficulty levels 1 and 2) or 15 (for level 3) individual target selections. In total, each participant performed 49 target selections for difficulty levels 1 and 2, and 105 target selections for difficulty level 3, for a total of 203 individual target selections for each menu style, or 609 selections total. Over all 18 participants, our analysis is based on 10962 target menu selections over 1134 trials. The experiment took about 1 h for each participant.

Dependent variables included *total selection time, selection accuracy*, and *search accuracy*. Total selection time is the average time (in seconds) per selection. We further sub-divided total selection time into execution time and reaction time. Execution time is measured as the time from touching the screen to releasing the finger from the screen, i.e., how long it takes to perform a selection gesture. Reaction time is total selection time minus execution time, i.e., the time required for participants to understand the stimulus and touch the screen. Selection accuracy (%) is the number of correct selections divided by the total number of selections. Finally, since participants could end trials without selecting all targets, search accuracy is the number of correct selections divided by the total number of targets presented.

4.5 Results

Total Selection Time. Total selection time is seen in Fig. 6 separated by reaction and execution time. We found significant main effects for both menu style ($F_{2,34} = 25.03$, $p < .001$) and difficulty level ($F_{2,34} = 62.49$, $p < .001$) on selection time.

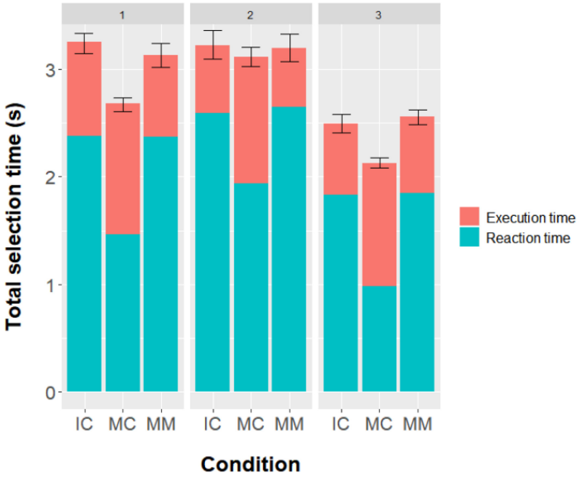

Fig. 6. Total selection time for each menu style (IC = ItemCount, MC = MenuCount, MM = Marking Menu) by difficulty level. Error bars show $\pm 1SD$.

Post-hoc testing with the Tukey-Kramer test revealed that MenuCount offers faster selections (2.64 s) than both ItemCount (3.00 s) and Marking Menu (2.97 s). This supports our hypothesis that the increased breadth of CountMarks offers faster selection time over menus with two depths.

Post-hoc testing with the Tukey-Kramer revealed that difficulty level 3 offered faster selection (2.39 s) than levels 2 (3.18 s) and 1 (3.02 s). We believe this is likely due to always presenting difficulty levels in the same order; it corresponds to participants improving with practice. The added difficulty of introducing the mode change did not seem to offset gains from learning effects.

Selection Accuracy. Selection accuracy by menu style is seen in Fig. 7 (left). ANOVA revealed a significant main effect of menu style on selection accuracy ($F_{2,34} = 14.26, p <$

.001). A Tukey Kramer posthoc revealed that Marking Menu was more accurate (97.91%) than MenuCount (95.00%) which was more accurate than ItemCount (88.44%). We believe this is due to the added complexity of extra fingers needed to perform selections with the two CountMarks variants; more fingers increases the opportunity for errors.

Search Accuracy. Search accuracy by menu style is seen in Fig. 7 (right). There was a significant main effect for menu style on search accuracy ($F_{2,34} = 100.98$, $p < .001$). A post-hoc Tukey-Kramer test showed that MenuCount (97.43%) and ItemCount (96.45%) had significantly better search accuracy than Marking Menu (92.16%).

This supports our hypothesis that CountMarks can improve search accuracy by increasing menu breadth. We also found a significant main effect for difficulty level on search accuracy ($F_{2,34} = 3.73$, $p < .05$). A Tukey-Kramer posthoc shows level 2 was significantly less accurate than levels 1 or 3. This was likely due to participants having difficulty finding items when first introduced to the mode-change double-tap, then becoming used to it by difficulty level 3. There was no interaction effect found.

Fig. 7. Mean selection and search accuracy by condition. Error bars show ±1*SD*.

4.6 Discussion

Overall, the MenuCount variant outperformed Marking Menus in selection speed and search accuracy. Marking Menus offered better selection accuracy. These results align with Lepinski et al.'s study [20] comparing multitouch marking menus to multi-stroke marking menus. They report that by increasing menu breadth with multi-finger chording gestures, their multi-touch marking menu could decrease selection speed.

Of the two CountMarks variants, MenuCount was faster and offered better selection accuracy than ItemCount. ItemCount's low selection accuracy may be due to two factors: item location did not always perfectly match its selection direction and participants may have been confused how many fingers to select with due to blank spaces left by previously selected items. Alternatively, participants may have been confused by how many fingers were required for selection as evidenced by ItemCount showing over twice as many errors from selecting items that had already been selected compared to MenuCount. This suggests an alternative design, where remaining items (not yet selected) could move up in the linear menu to take the place of selected ones.

The two CountMarks variants also had quite different execution and reaction times. ItemCount required more search (i.e., reaction) time for targets, while MenuCount had a longer execution time, but the overall shortest total selection time. This suggests that breaking the search component up into smaller, more easily accessible and reversible components (menus can be quickly previewed by placing a number of fingers down and releasing them) selection time improved. However, ItemCount offered better search accuracy than MenuCount. We believe this reflects the ability for participants to see more items at once and determine whether any targets remain.

The differences between difficulty levels 1 and 2 were minimal. These two levels showed either 4 menus with 8 items each or 8 menus with 4 items each. Level 2 also required a double tap gesture to access the other set of items or menus. The double tap is likely why MenuCount took longer in level 2. ItemCount was not significantly different across the two difficulty levels, suggesting its difficulty level 2 layout (4 menus $\times$ 8 items $+$ double tap) was quicker to search than having all 32 items displayed at once across 8 menus in level 1. This suggests that breadth offers better efficiency to a point, at which point semantic groupings offer better efficiency, despite additional menu depth. Difficulty level 3 outperformed the rest likely due to practice effects and the larger number of targets and selections without breaks in between.

5 Experiment 2: CountMarks Use Scenarios

We evaluated MenuCount in more realistic use cases (standing vs. walking) using a head-mounted display.

5.1 Participants

We recruited 20 participants (mean age of 24.6 years, $SD = 4.3$, all right handed, 12 female). Nine participants reported often texting while walking. Six participated in Experiment 1. Participants were recruited via posters and social media.

5.2 Apparatus

We used the same Samsung Galaxy S8 smartphone as the touch input device. We used a Microsoft HoloLens HMD, to more realistically simulate smart-glasses usage scenarios. The software was built in Unity and used Unity's built-in networking tools, UNet, to take input from the Android app to the same desktop computer as Experiment 1. The computer display was wirelessly mirrored to the HoloLens via the HoloLens Remoting app.

Depending on the condition, the smartphone was either held in the participant's non-dominant hand or mounted on the upper part of the leg in a comfortably reachable position for the dominant hand. Participants walked on a Tempo Fitness 610T treadmill. Figure 8 depicts the hardware setup. The software recorded reaction time, execution time, and selection and search accuracies, as in Experiment 1.

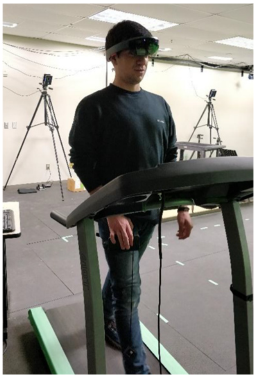

Fig. 8. A participant walking while using CountMarks on a leg-mounted smartphone.

5.3 Procedure

Upon arrival, participants were briefed and provided informed consent prior to starting. They were then given a demonstration of how to properly wear the HoloLens to see the full display. We began the experiment by identifying which way participants thought was "up" when swiping on their leg. This was non-obvious because "up" depends on the participant's frame of reference, and could mean up relative to the participant, the forward direction they were facing, or the direction their hand is pointing at the time of interaction. Participants were shown an image of an arrow pointing to the top of the monitor and asked to perform a swipe gesture on the leg-mounted smartphone to determine the "up" direction. While we recorded it, it was not used to determine the swipe direction during the study, which instead used a consistent swipe direction for all participants (towards the knee, based on pilot testing).

Participants then completed a demographic questionnaire before receiving instructions on how to use CountMarks to complete the Novice Task. They were given 3 min to practice CountMarks with an unrecorded trial. After practice, we found a comfortable treadmill walking speed for each participant. We asked participants to walk on the treadmill at a speed of 2.0 mph while wearing the HoloLens. Participants could increase or decrease the speed by up to 0.5 mph (the smallest increment on the treadmill). No participant chose to go higher than 2.0 mph, but 12 participants chose to go slower. In walking conditions, the treadmill was activated at the chosen speed. In standing conditions, participants stood on the deactivated treadmill.

The experiment task was identical to that used in Experiment 1. Participants performed 4 trials per condition.

5.4 Design

The experiment employed a 2 × 2 within-subjects design, with the following independent variables and levels:

- Device position (hand, leg)
- User stance (standing, walking)

The order of device position and user stance conditions were counterbalanced via a Latin square.

The dependent variables included *selection time* (further sub-divided into *reaction time*, *execution time*), *selection accuracy,* and *search accuracy*. All were calculated as described in Experiment 1.

In total, 20 participants were tested across 2 user stances with 2 device positions. In each condition, participants completed 4 trials of 15 selections each, for a total of 20 participants × 2 user stances × 2 device positions × 15 selections per trial * 4 trials = 4800 selections across 320 trials. The experiment took about 30 min to complete.

5.5 Results

Due to a software error the final trial of some of the first 5 participants was not recorded. Of the remaining 4800 recorded selections, we filtered selections faster than 0.001 s reaction or execution times, and those with longer than 10 s reaction or execution, leaving 4397 selections. These times were either too fast or slow to be intentional and were thus software or hardware errors.

Selection Time. Analysis of variance revealed a significant main effect of device position on selection time (F1,19 = 26.82, $p < .001$). Holding the device in the hand offered faster selection time (2.00 s) than interacting with the device on the leg (2.41 s). This may suggest that the leg is an unfamiliar place for interaction. Analysis of variance revealed a significant main effect of user stance on selection time (F1,19 = 6.94, $p < .01$). Participants had faster selection times while standing (2.09 s) than when walking (2.30 s). The interaction effect between device position and user stance was also significant (F3,57 = 6.15, $p < .05$). When walking, selection time was much higher on the leg than the hand. In contrast, device position had little impact when the user was standing. Total selection times for each condition are seen in Fig. 9.

Selection and Search Accuracy. Analysis of variance revealed a significant main effect of device position on accuracy ($F_{1,19} = 76.89$, $p < .001$). Selections with the device in the hand were more accurate (95.56%) than selections with the leg-mounted device (84.50%). As before, this is likely because the leg is an unfamiliar interaction location. See Fig. 10 (left). The main effect of user stance on selection accuracy was also significant ($F_{1,19} = 51.44, p < 0.001$). Participants were more accurate while standing (94.53%) than while walking (85.71%). The device position/user stance interaction effect was significant ($F_{3,57} = 33.69$, $p < .001$) This is seen Fig. 10; walking was substantially worse with leg-mounted conditions, than hand-held. This is likely due to the added impact of the device moving on the leg during walking.

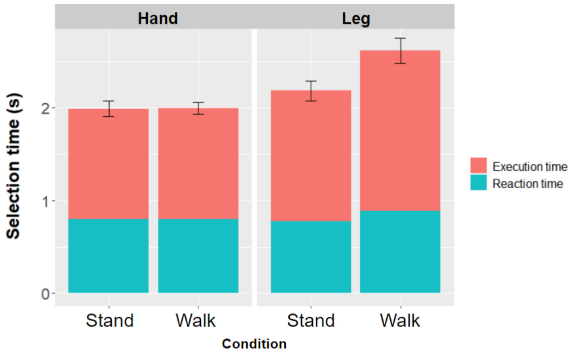

Fig. 9. Selection times broken down by condition. Error bars show ±1*SD*.

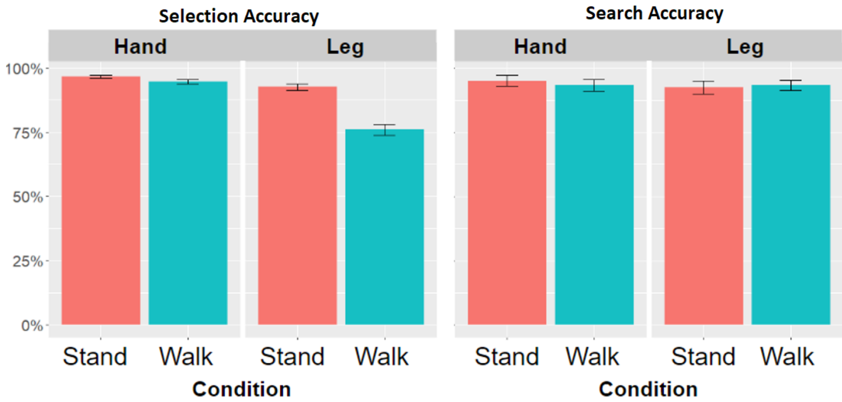

Fig. 10. (Left) Selection accuracy broken down by condition. (Right) Search accuracy by condition. Error bars show ±1*SD* for mean trial accuracy.

Search Accuracy. There were no significant effects for user stance or device positioning on search accuracy. See Fig. 10 (right). Search accuracy was slightly lower in this experiment (93.55%) than in level 3 of the first experiment (97.13%).

5.6 Discussion

The goal of this experiment was to assess realistic use cases of CountMarks by testing different device positions (the hands vs the leg) with different user stances (standing and walking). For both selection time and accuracy, the combination of leg-mounted device and walking was significantly worse than other conditions. Search accuracy was consistent across all conditions. This makes sense; unlike Experiment 1, which compared different visual menu layouts, we only used a single variant of CountMarks (MenuCount) in this experiment. Clearly, the menu layout impacts search accuracy, while the style of interaction with the menu (and other external factors like grip and mounting of the device) influence selection speed and accuracy.

As described earlier, we asked participants to define the “up” swipe direction on their leg at the onset of the experiment. As expected, results were mixed; 40% of participants said they would swipe from back to front, 30% would swipe towards their feet and 30% would swipe towards their head. Future work could further explore how this preconceived orientation of swipe gesture affects performance. The inconsistency of participants’ default “up” swipe direction when perform on the leg suggests that a calibration setting may be beneficial to CountMarks. This confusion may contribute to the lower accuracy of leg interactions. Future implementations could allow users to choose their desired swipe orientation to improve accuracy further.

6 Overall Discussion

While Experiment 1 validated the efficiency of CountMarks, Experiment 2 explored interactions using CountMarks in mobile settings at different device positions. Selection speed and accuracy was comparable for corresponding conditions in both experiments (e.g., the Experiment 2 hand-held position vs. the difficulty level 3 in Experiment 1). Search accuracy was the only measure notably different between the two experiments. This is likely due to the higher variance seen in Experiment 2; the standing hand-held condition offered a comparable 95.10% search accuracy vs. 97.17% in Experiment 1. Such gestures would also work in practical contexts, unlike the desk-mounted touchscreen used in our first experiment.

6.1 CountMarks Compared to Other Techniques

Table 1 shows selection times for CountMarks compared to several other reported studies on marking menus.

Table 2 A comparison of performance for different marking menu styles using 8 × 8 menus. Note that while the methodologies are not identical for each evaluation, they correspond to the same or similar tasks. *Polygon uses 12 × 12. **CountMarks in the standing, handheld condition. *** MenuCount variant at difficulty level 3.

Name of technique	Selection type	Selection time	Selection accuracy
Kurtenbach marking menu [15]	Expert	~2.3 s	80%
Multi-stroke marking menu [37]	Expert	~2.3 s	93%
Polygon menu* [38]	Expert	~2.0 s	95%
Multi-touch marking menu [20]	Novice	2.37 s	93.60%
Multi-touch marking menu [20]	Expert	0.81 s	85.10%
CountMarks (Exp. 2) **	Novice	1.99 s	96.50%
CountMarks (Exp. 1) ***	Novice	2.64 s	97.40%

Selecting targets required two search components: the first to choose a menu, and the second to find the target in the menu. While selecting a menu relies on visual search to find the correct menu, subsequent menu selections may not require the initial menu search when selecting multiple targets in the same menu. Such selections may be performed like *expert* selections that do not require search.

We include comparable conditions from both Experiment 1 and 2 data. These are the MenuCount + difficulty level 3 condition (Experiment 1) and standing with the phone in the hand condition (Experiment 2). Notably, despite requiring search, the Experiment 2 condition was faster than, or at least comparable to expert selection (i.e., without search) with most other marking menus while offering better accuracy. On average (2.315 s), CountMarks is comparable to other techniques, but the higher performance seen in Experiment 2 suggests great potential of the technique.

According to Kurtenbach [15] acceptable accuracy depends on the consequence of errors, and the difficulty in reversing them. Compared to other marking menus (Table 2), ideal accuracy should be around 95%, but these studies only tested selection while sitting at a desktop computer with a stationary input device. It is unclear what an acceptable error rate is for mobile interactions. As seen in Table 2, CountMarks outperformed other marking menu variants in terms of selection accuracy.

6.2 User Stance and Device Position

We evaluated marking menu selections in two previously unexamined domains: standing vs. walking, and with the input device handheld vs. leg-mounted. We were surprised that walking did not affect selection speed during the hand-held condition and increased selection speed by only 0.41 s during the unfamiliar leg condition. This suggests that users can use CountMarks almost as quickly while moving as when standing still. Since we intended CountMarks to be used with a mobile HMD, this is an important finding.

Compared to standing, walking slightly reduced selection accuracy by around 2% when the device was handheld. While standing, leg-mounted selection had accuracy of 92.5% which is still comparable to other marking menu techniques (Table 2). Leg-mounted accuracy dropped considerably to 75.9% when walking. Since past marking menus have not been tested in walking/standing scenarios, it is unclear if our results are typical for such conditions.

Compared to Experiment 1, we anticipated lower accuracy for the Experiment 2 Leg/Walking conditions due to device location being on the body part used for motion. This motion causes the plane of the interaction surface to be constantly changing. As the leg moves while walking, the touch surface shifts its orientation with the leg such that the horizontal axis of the phone is not the same at all times. Therefore, swipe directions may need to be relative to the orientation of the device in motion. Eight participants specifically mentioned the difficulty of coordinating hand and leg movements when making selections. Despite this difficulty, we were surprised with how accurate user were able to perform Leg/Walking gestures. We propose that future work investigate adjusting the user's swipe direction based on the input device's change in orientation while walking. We hypothesize that by fixing issues of orientation during leg movement and calibrating for users' preconceived notion of "up" that the leg may become a viable option for mobile marking menu interactions.

7 Example Application

Despite their reported efficiency [5], marking menus are rarely used in commercial products. We argue that CountMarks opens up the design space for designers to implement marking menus into modern applications. To demonstrate that CountMarks is flexible enough for use in existing applications, we developed a Netflix mockup using CountMarks (Fig. 11). Netflix was chosen to demonstrate selection from a large number of menu items, it is not intended as a mobile use scenario.

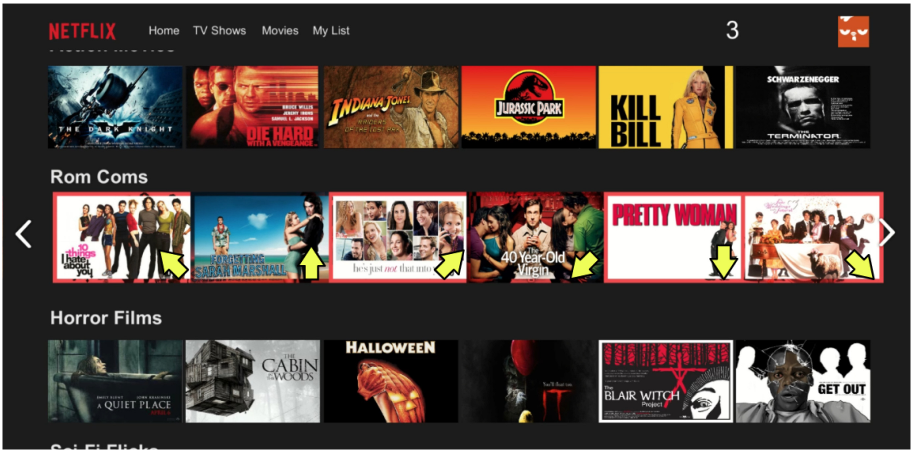

Fig. 11. A demo Netflix application where the user has 3 fingers held down to select the middle row of movies. Swipe directions to specify a desired movie are indicated by yellow arrows on the currently indicated row.

The user can scroll vertically by sliding one finger on the screen. Touching two or more fingers on the screen activates CountMarks to select from different movie genres. Touching two fingers activates the top row, three fingers activate the middle, and four fingers activate the bottom row. Upon activating a row in this fashion, the row becomes highlighted, and arrows appear in the corner of the movie images, corresponding to the swipe direction required to select that movie. Swiping left or right with the desired number of fingers will horizontally scroll the specified row. Finally, a double tap toggles the selection mode to focus on the top bar of items where the user can select settings and other options.

While the Netflix app has content neatly organized into rows and columns, future work is needed to investigate how CountMarks can generalize to other less structured user interfaces to make selections. Websites and many other applications have undergone major redesigns over the last decade to accommodate small screen mobile devices. It is unclear how interfaces will change for smart glasses.

8 Conclusions

In this paper, we demonstrate how we designed, implemented and evaluated a novel multi-touch marking menu technique designed for mobile interactions with HMDs. This work was inspired by previous research on Count Menus, marking menus, and on-body interaction, to create an interaction technique that is designed to be fast, accurate, ergonomic and appropriate for use in a public setting. We conducted two experiments to evaluate CountMarks. The first experiment compared two variations of CountMarks to existing marking menus. CountMarks overcomes the limited breadth of traditional marking menus by allowing 32 items in a single depth and up to 64 items with a mode shifting double tap. Our findings show that the MenuCount variation of CountMarks offers quicker selection and improved search accuracy than multi-stroke marking menus at the cost of a minor decrease in selection accuracy.

In the second experiment, participants used the MenuCount variant with a Microsoft HoloLens while standing and walking and with the phone held in the hand and attached to the leg. Walking slightly reduced selection time but did cause a significant decrease in accuracy when the input device was at the leg. Leg interactions in general were slower and more prone to errors than handheld interactions. Overall, we show that Novice selection using CountMarks outperforms even Expert selection from most other marking menu variations. Future work should explore expert selection with CountMarks as well as the ergonomic and social acceptability of CountMarks when performed at different device positions.

Altogether we contribute an improved marking menu style of selection using multiple fingers. We provide empirical evidence into the strengths and weakness of performing these interactions while in mobile settings and with different device input locations. Expanding upon well-established marking menus, we believe CountMarks is a viable alternative for mobile interaction with HMDs that would otherwise use in-air gestures, voice or joystick-based controls.

References

1. Bailly, G.: Finger-count & radial-stroke shortcuts: two techniques for augmenting linear menus on multi-touch surfaces. In: Hand, pp. 1–4 (2010)
2. Bailly, G., Lecolinet, E., Nigay, L.: Visual menu techniques. ACM Comput. Surv. **49**(4), 1–41 (2016)
3. Bailly, G., Lecolinet, E., Nigay, L.: Flower menus. In: Proceedings of the Working Conference on Advanced Visual Interfaces – AVI 2008, p. 15 (2008)
4. Bailly, G., Müller, J., Lecolinet, E.: Design and evaluation of finger-count interaction: combining multitouch gestures and menus. Int. J. Hum. Comput. Stud. **70**(10), 673–689 (2012)
5. Bailly, G., Lecolinet, E., Nigay, L.: Wave menus: improving the novice mode of hierarchical marking menus. In: Baranauskas, C., Palanque, P., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4662, pp. 475–488. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74796-3_45
6. Cao, C.G.L., Zhou, M., Jones, D.B., Schwaitzberg, S.D.: Can surgeons think and operate with haptics at the same time? J. Gastrointest. Surg. **11**(11), 1564–1569 (2007). https://doi.org/10.1007/s11605-007-0279-8

7. Carroll, J.M., Rosson, M.B.: Paradox of the active user. In: Carroll, J.M. (ed.) Interfacing Thought: Cognitive Aspects of Human-Computer Interaction, pp. 80–111. MIT Press, Cambridge (1987)
8. Cockburn, A., Quinn, P., Gutwin, C., Ramos, G., Looser, J.: Air pointing: design and evaluation of spatial target acquisition with and without visual feedback. Int. J. Hum. Comput. Stud. **69**(6), 401–414 (2011)
9. Dobbelstein, D., Hock, P., Rukzio, E.: Belt: an unobtrusive touch input device for head-worn displays. In: Proceedings of the ACM CHI 2015 Conference on Human Factors in Computing Systems, vol. 1, pp. 2135–2138 (2015)
10. Harrison, C., Benko, H., Wilson, A.D.: OmniTouch. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology – UIST 2011, p. 441 (2011)
11. Harrison, C., Tan, D., Morris, D.: Skinput. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems – CHI 2010, p. 453 (2010)
12. Hincapié-Ramos, J.D., Guo, X., Moghadasian, P., Irani, P.: Consumed endurance. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems – CHI 2014, February, pp. 1063–1072 (2014)
13. Hsieh, Y.-T., Jylhä, A., Orso, V., Gamberini, L., Jacucci, G.: Designing a willing-to-use-in-public hand gestural interaction technique for smart glasses. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems – CHI 2016, pp. 4203–4215 (2016)
14. Kiger, J.I.: The depth/breadth trade-off in the design of menu-driven user interfaces. Int. J. Man Mach. Stud. **20**(2), 201–213 (1984)
15. Koelle, M., Kranz, M., Andreas, M.: Don't look at me that way! – understanding user attitudes towards data glasses usage. In: MobileHCI 2015 Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Service, pp. 362–372 (2015)
16. Kulshreshth, A., LaViola, J.J.: Exploring the usefulness of finger-based 3D gesture menu selection. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems – CHI 2014, pp. 1093–1102 (2014)
17. Kurtenbach, G., Buxton, W.: The limits of expert performance using hierarchic marking menus. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI 1993, pp. 482–487 (1993)
18. Kurtenbach, G.: The design and evaluation of marking menus. University of Toronto, Toronto (1993)
19. Lane, D.M., Napier, H.A., Peres, S.C., Sandor, A.: Hidden costs of graphical user interfaces: failure to make the transition from menus and icon toolbars to keyboard shortcuts. Int. J. Hum. Comput. Interact. **18**(2), 133–144 (2005)
20. Lepinski, G.J., Grossman, T., Fitzmaurice, G.: The design and evaluation of multitouch marking menus. In: Proceedings of the 28th International Conference on Human Factors in Computing Systems – CHI 2010, p. 2233 (2010)
21. Liu, M., Nancel, M., Vogel, D.: Gunslinger: Subtle Arms-down Mid-air Interaction, pp. 63–71 (2015)
22. Lucero, A., Vetek, A.: NotifEye. In: Proceedings of the 11th Conference on Advances in Computer Entertainment Technology – ACE 2014, pp. 1–10 (2014)
23. Pielot, M., Hesselmann, T., Heuten, W., Kazakova, A., Boll, S.: PocketMenu: non-visual menus for touch screen devices. In: Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI 2013), pp. 327–330 (2012)
24. Ren, G., O'Neill, E.: 3D Marking menu selection with freehand gestures. In: IEEE Symposium on 3D User Interfaces 2012, 3DUI 2012 - Proceedings, pp. 61–68 (2012)

25. Saponas, T.S., Harrison, C., Benko, H.: PocketTouch. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology – UIST 2011, Figure 1, p. 303 (2011)
26. Serrano, M., Ens, B.M., Irani, P.P.: Exploring the use of hand-to-face input for interacting with head-worn displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 3181–3190 (2014)
27. Shneiderman, B.: Direct manipulation: a step beyond programming languages (excerpt). In: Readings in Human-Computer Interaction: A Multidisciplinary Approach, pp. 461–467 (1987)
28. Teather, R.J., Stuerzlinger, W.: Pointing at 3D target projections with one-eyed and stereo cursors. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 159–168 (2013)
29. Tung, Y.-C., et al.: User-defined game input for smart glasses in public space. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems – CHI 2015, pp. 3327–3336 (2015)
30. Wang, C.-Y., et al.: PalmGesture: using palms as gesture interfaces for eyes-free input. In: Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 217–226 (2015)
31. Weigel, M., Mehta, V., Steimle, J.: More than touch: understanding how people use skin as an input surface for mobile computing. In: Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems – CHI 2014, pp. 179–188 (2014)
32. Weigel, M., Lu, T., Bailly, G., Oulasvirta, A., Majidi, C., Steimle, J.: iSkin: flexible, stretchable and visually customizable on-body touch sensors for mobile computing. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems – CHI 2015, pp. 2991–3000 (2015)
33. Weigel, M., Nittala, A.S., Olwal, A., Steimle, J.: SkinMarks: enabling interactions on body landmarks using conformal skin electronics. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems – CHI 2017, pp. 3095–3105 (2017)
34. Weiser, M.: The computer for 21st century. Sci. Am. Ubicomp Pap. **265**(3), 8 (1991)
35. Yoon, S.H., Huo, K., Ramani, K.: Wearable textile input device with multimodal sensing for eyes-free mobile interaction during daily activities. Pervasive Mob. Comput. **33**, 17–31 (2016)
36. Zhang, Y., Zhou, J., Laput, G., Harrison, C.: 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 2016, pp. 1491–1503 (2016)
37. Zhao, S., Balakrishnan, R.: Simple vs. compound mark hierarchical marking menus. In: Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology – UIST 2004, vol. 6, no. 2, p. 33 (2004)
38. Zhao, S., Agrawala, M., Hinckley, K.: Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1077–1086 (2006)
39. Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., Baudisch, P.: earPod: eyes-free menu selection using touch input and reactive audio feedback. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, p. 1395 (2007)
40. Zheng, J., Bi, X., Li, K., Li, Y., Zhai, S.: M3 gesture menu. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing System – CHI 2018, pp. 1–14 (2018)