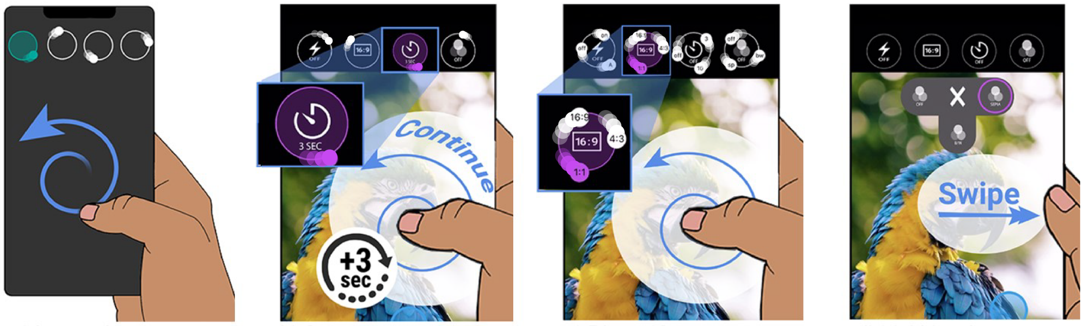
One-handed Input for Mobile Devices via Motion Matching and Orbits Controls

AUGUSTO ESTEVES, ITI / LARSyS, Instituto Superior Técnico, University of Lisbon, Portugal
ELIZABETH BOUQUET, Ludwig-Maximilians-University Munich, Germany
KEN PFEUFFER, Aarhus University, Denmark and University of the Bundeswehr Munich, Germany
FLORIAN ALT, University of the Bundeswehr Munich, Germany



a) Interaction concept b) Set camera parameters c) Direct 12-target access d) Multi-level menus

Fig. 1. One-handed operation of smartphones and other touch-based devices is often necessary or preferred in various scenarios where two-handed use is not possible or convenient. We explore touch-based motion matching to select out-of-reach targets from a comfortable device grip (a). We detail user performance in an empirical user study and demonstrate, through a camera application, how it can extend mobile UIs with precise parameter controls (b), direct access to up to 12 targets (c), and sub-level menu access and selection (d).

We introduce a novel one-handed input technique for mobile devices that is not based on pointing, but on motion matching – where users select a target by mimicking its unique animation. Our work is motivated by the findings of a survey (N=201) on current mobile use, from which we identify lingering opportunities for one-handed input techniques. We follow by expanding on current motion matching implementations – previously developed in the context of gaze or mid-air input – so these take advantage of the affordances of touch-input devices. We validate the technique by characterizing user performance via a standard selection task (N=24) where we report success rates (>95%), selection times (~1.6 s), input footprint, grip stability, usability, and subjective workload – in both phone and tablet conditions. Finally, we present a design space that illustrates six ways in which motion matching can be embedded into mobile interfaces via a camera prototype application.

Authors' addresses: Augusto Esteves, augusto.esteves@tecnico.ulisboa.pt, ITI / LARSyS, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal; Elizabeth Bouquet, e.bouquet@campus.lmu.de, Ludwig-Maximilians-University Munich, Munich, Germany; Ken Pfeuffer, ken@cs.au.dk, Aarhus University, Aarhus, Denmark and University of the Bundeswehr Munich, Munich, Germany; Florian Alt, florian.alt@unibw.de, University of the Bundeswehr Munich, Munich, Germany.

2474-9567/2022/6-ART51 $15.00
https://doi.org/10.1145/3534624

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; **Interaction techniques**; *Empirical studies in ubiquitous and mobile computing*; Mobile computing.

Additional Key Words and Phrases: motion matching, orbits, one-handed input, reachability, mobile use survey

ACM Reference Format:
Augusto Esteves, Elizabeth Bouquet, Ken Pfeuffer, and Florian Alt. 2022. One-handed Input for Mobile Devices via Motion Matching and Orbits Controls. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 2, Article 51 (June 2022), 24 pages. https://doi.org/10.1145/3534624

1 INTRODUCTION

The interaction paradigm on mobile devices is well established, but the challenge of whole-screen reachability during one-handed use persists. This is particularly true of large-screen smartphones, who have almost doubled in size since the original iPhone at 3.5", and of tablet computers. Various interaction techniques have been introduced over the years to address this challenge, ranging from screen transforms [7], proxy input areas [30], to cursor-based techniques [69].

While these are quite interesting and successful approaches, they all share the same underlying principle: they are (direct or indirect) pointing-based techniques. As such, in this paper we propose a one-handed input technique for touchscreen devices that relies not on pointing, but for the first time on a increasingly popular interaction paradigm known as *motion matching* [61]. In these types of interfaces the user selects a target by mimicking its unique animation for a short time. Selections are performed by comparing *signals* – i.e., whether target animation and user touch trajectory are similar (as illustrated in Figure 1a). The approach allows easy access to out-of-reach targets, afforded by the following unique interaction properties.

First, it eliminates finger traversal to the target as selection is not bound to the distance between finger and target (*position-independence* [17]). This works regardless of orientation or device form factor – from phones to tablets, as we assess in our experimental evaluation. Second, selection is accomplished regardless of target or hand size [37]. Targets can be as large [6] or as small as needed [18], as these do not need to match the scale of the user's input motions nor facilitate pointing (*scale-independence*). We argue these two properties can be particularly useful for one-handed operation of mobile devices.

Our objective is to investigate the first motion matching implementation for touchscreen devices in the context of one-handed use. Our proposed design expands on the state-of-the-art – aimed primarily at uncalibrated gaze input [68], mid-air controls [8], or augmented- [19] and virtual-reality [33] – through the unique affordances and challenges of touch and mobile UIs: (i) a mode switch based on a bezel swipe and the take-off selection paradigm [48], enabling our technique to coexist with direct touch; and (ii) a *matching state* that provides critical user feedback and explicit control over target selections and reduces error rates. In an empirical user study, we show how users can easily employ the technique on both phone and tablet form factors, and detail user performance through a comparison to default touch and a reachability technique from literature.

At last, we exemplify multiple ways in which our technique can be embedded in a mobile UI through a camera app prototype. For example, to quickly set a timer to capture photos (Figure 1b), directly switch photo aspect ratios (c), and integration with sub-level menus and touch gestures (d). These examples showcase the compatibility of motion matching with more familiar UI elements and gestures such as dragging or swiping, and together enable advanced operations through single thumb movements in one-handed grip.

The remainder of the document is organized according to our main contributions:

- We present the results of an online survey (N=201) that highlights how two-handed use is still prevalent due to convenience and efficiency, particularly in media-recording applications while one-handed use is much more contextual.

- We present and motivate our motion matching design and implementation for touchscreen devices, and how our mode switching and *matching state* approaches expand the state-of-the-art.
- We report on the first user study on motion matching for touchscreen devices. We demonstrate (i) the feasibility of the technique (no observable performance differences to arguably the best-performing pointing-based technique in this domain [38]); (ii) the *position*- and *scale-independence* properties above; and (iii) vast performance improvements over previous motion matching implementations (from an error rate of ∼10-30% to ∼5%).
- Finally, we present a design space that illustrates six ways in which motion matching can be embedded into existing mobile interfaces and combined with more familiar interactions for more complex operations in the context of a camera application.

2 RELATED WORK

2.1 Mobile Device Reachability

Inspired by Bergstrom-Lehtovirta et al.'s [4] model of a user's thumb reach on smart phones, researchers in HCI often look into interesting input techniques that attempt to improve user's precision, reach, grip, and overall comfort during one-handed input – a common need when interacting with these types of mobile devices [32]. These techniques cluster around three approaches [7]: screen transforms, proxy input areas, and cursor-based techniques. These are either implemented using available input (e.g., the touchscreen), embedded motion sensors, or require bespoke hardware.

2.1.1 Screen Transforms. These types of techniques are commonly available in most smart phone operating systems and work by briefly manipulating the content being displayed so it is within reach of the user's thumb. This manipulation is normally done in one of two ways: either by solely panning this content to the bottom of the screen; or by simultaneously panning and shrinking. Examples of the former include the Reachability mode in iOS[1], and a variety of others that allow the user to pan content in response to particular swipes [34, 60], implicit [14] or explicit tilts of the device [7], or by how they unlock their phones [40]. Furthermore, Le et al. [36] has explored the use of bespoke hardware on the back of the display to similarly trigger this content panning. While effective, these approaches need to omit large sections of content and, thus, hide contextual information, require mode switching often when the user needs to continuously select targets that are within and out-of-reach, and explicitly tilting the device has been found to lead to overshooting [58]. Similarly, examples of panning and shrinking approaches include Android/Samsung's One Hand Mode and TiltReduction [7], but these exacerbate the fat finger problem by making content harder to read and select [56].

2.1.2 Proxy Input Areas. This approach enables user input on a small region in reach of their fingers, which is then mapped to the entire screen. ThumbSpace [31] does this by displaying a translucent and shrunken version of the display in this accessible region, but can lead to the just described fat finger problem and to targets that are effectively smaller and, thus, harder to select. TapTap [54] does not suffer from this as it simply displays a magnified portion of the display in this accessible region. Yet, it does not scale well to larger displays, and, as with panning approaches, hides large portions of content and often needed contextual information. In response to these limitations, Löchtefeld et al. [39] explored the use of a touch surface on the back of the device, while Hasan et al. [27] described an input space that resides above and around the display. Both approaches are constrained by their need of bespoke hardware.

2.1.3 Cursor-based Input. Cursor-based approaches display an often accelerated cursor, extending the user's reach. These vary on how they are triggered, from swipes [38], double-taps [35], to tilting the device [7]; and they

---

[1] https://support.apple.com/guide/iphone/touch-iph77bcdd132

steer the cursor in the same or opposite direction of the thumb movement (to address occlusion issues) [34, 54]. As these techniques tend to cause grip instability and fatigue when the thumb is moved beyond its comfort zone, other techniques have reduced the necessary cursor travel by placing this in the corners of the display [74], or by having this cycle solely through the interactive elements on-screen [59]. The former, however, can decrease accuracy when selecting targets that are within the reach of the user's thumb. The latter can be tedious and time-consuming. Finally, several solutions have been proposed that rely on bespoke or non-ubiquitous hardware. Examples include ForceRay [11], a cursor-based approach that relies on applying pressure on the device with the thumb; HeadReach [69], a technique that facilitates the use of the cursor via head motions; Gaze'N'Touch [53], a mobile text-selection technique combining touch and eye-gaze; and Dual-Surface Input [73], a touchscreen residing on the back of the device and accessed via the index finger.

2.1.4 Summary. Existing techniques have various benefits and constraints but all share a similar characteristic: they all address the issues of reachability and one-handed input via a traditional pointing-based approach. This approach is a staple of HCI since the days of the WIMP paradigm, with very well known constraints as illustrated by various models such as Fitts' Law [21]: user performance is affected by the distance between the intended target and the thumb, the target size, and also its distance to other targets. We argue the latter two constraints are particularly limiting in the context space-constrained devices such as smart phones, and ultimately hinder how we approach mobile interface design.

2.2 Motion Matching

The ideas behind motion matching input have been introduced more than a decade ago [70] and were conceived, at first, for standard mouse-based graphical user interfaces (GUIs) [20]. In these systems, users interact with interface elements not by pointing at them, but by replicating a unique rhythmic animation that accompanies them. Over the years the advantages of these systems have been fleshed out by a variety of works.

While some exceptions exist [12], most motion matching implementations rely on normalized signals that allow for position- and scale-independent input. These implementations have enabled uncalibrated gaze interactions with out-of-reach [63, 68] and wrist-worn displays [18]. More generally, this means that the size and location of an interface target should have little impact on users' performance. That is, users can perform a matching motion in the location and range most comfortable to them, regardless of target size [19, 23]. Similarly, because no cursor travel is needed, selecting a target at the edge of the display should take no longer than any other target in the interface [17]. Second, because interfaces do not need to facilitate pointing via well-sized and well-spaced targets, these can overlap in space-constrained displays (e.g., smart watches [18]), or even be superimposed onto one another in spatial and 3D interfaces (e.g., VR/AR [19, 47, 55]). And third, when compared to traditional gestural interfaces: (1) the user does not need to learn or memorize any gestures, what Norman [43] describes as issues of *visibility*, *discoverability*, and *consistency* – i.e., interface targets continuously reinforce the matching motions needed for selection via rhythmic animations – and (2) accidental activation is less likely to occur as user input needs to match these animations in real-time, what Norman describes as an issue of *reliability* – i.e., actions need to be matched in their spatial and temporal components. Taken together, this has led motion matching to become a popular mid-air input technique for smart TVs [8, 9, 67], smart environments [22, 50, 64], or public displays [6]; seemingly not affected by what Norman describes as issues of *scalability* with gestural interfaces.

In sum, while there have been some efforts in using motion matching to support pointing-based systems by, for example, enabling implicit [46] or explicit [10] calibration of out-of-reach displays, most work in this domain looks at proof-of-concept prototypes that address the limitations of pointing-based or gestural input in emerging domains such VR/AR or smart homes and ubiquitous computing [2, 75]. That is, very little research has been done on the use of motion matching in support of more standard input for traditional GUIs since Fekete et al.'s [20] seminal motion-pointing work in 2009.

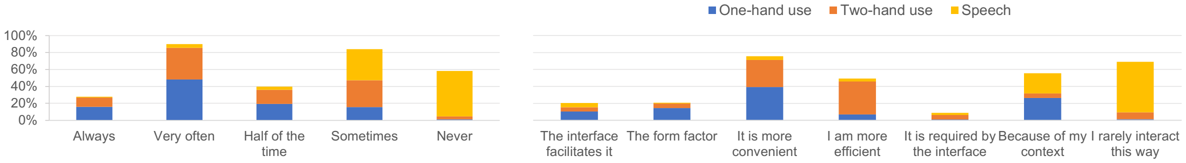


Fig. 2. Left: survey data for current smart phone use regarding one- and two-handed input and speech. Right: rationale for providing input in such a manner.

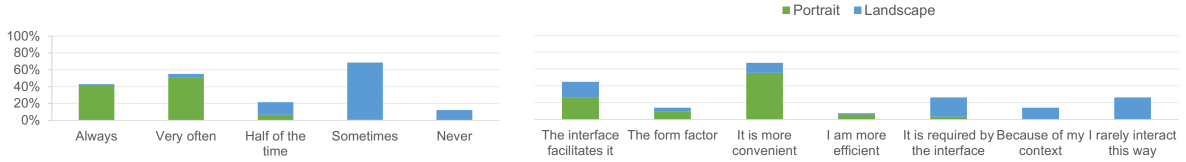


Fig. 3. Left: data for current orientation modes. Right: rationale for using the smart phone in portrait or landscape modes.

One exception in this domain is Malacria et al.'s [41] clutch-free panning an zooming controls using repeated motions on a touchscreen. While these are particularly useful in various reachability contexts and resemble motion matching input due to their repeated motions, these motions are used not to discriminate between standard interface targets but to establish the parameters of a limited set of functions (e.g., panning direction and speed). Another exception is Bennett et al.'s [3] work on Resonant Bits, a motion matching interface for smart phones that has users tilt the device at a rhythm that matches the oscillation of one of the targets in the interfaces. While this is an inspiring look at these types of interfaces for standard mobile devices, we argue the constant tilting of the device can make contextual information harder to read (particularly while walking), and the design of the oscillation animations can be challenging to embed into already existing mobile interfaces. We address the latter when describing our motion matching design and implementation, but first, we motivate this work and future work via a mobile use survey that provides timely insights into current input modalities and device orientations in the context of various mobile applications.

3 MOBILE USE SURVEY

At the outset of our work we conducted an online survey to explore which applications or scenarios still require two-handed use and why, as well as the prevalence of novel forms of hands-free input such as speech. Most cited work in this area is already dated, such as Karlson et al.'s survey on one-handed use of mobile devices where more than half of participants still operated flip phones [32]; or Quinn et al.'s survey from 2013 on device orientation [49]. To facilitate comparisons between our data and these earlier works, our survey remained as close as possible to the original questions. It can be accessed at https://forms.gle/houU8kTevWJrwMJh8.

Our survey was conducted via an online crowd-sourcing platform specifically aimed at academic studies[2]. It took approximately nine minutes to complete, and participants were awarded ~\$1 for their time. The survey was conducted in English and required being at least 18 year of age. 201 participants took part (75 females), aged between 18 and 69 years (M = 28.32, SD = 10.21). 52.24% of these were employed, 35.82% were students, 7.96% were unemployed, and 3.98% worked either in higher education, research, or were retired. 47.76% of participants were from America (USA, Canada, Mexico, Chile), 40.8% from Europe (including the UK), 7.46% were from Oceania,

---

[2] https://www.prolific.co/

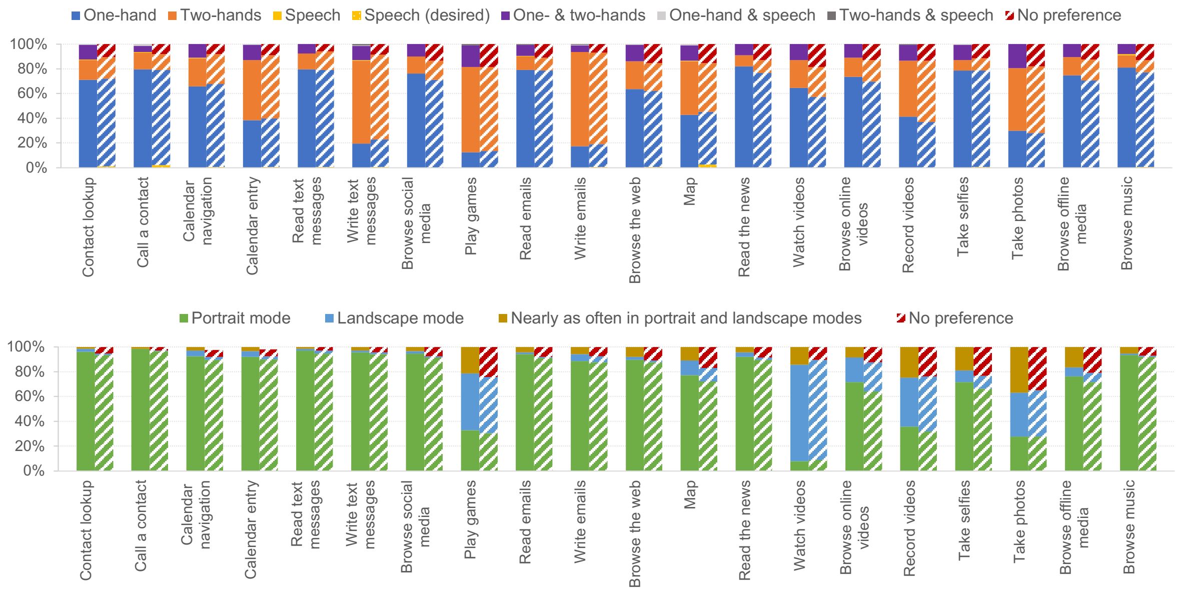


Fig. 4. Top: survey data on how respondents interact (solid fill) and how do they prefer to interact (dashed fill) with a variety of applications. Bottom: in which orientations do respondents interact (solid fill) and which orientations do they prefer to interact (dashed fill) with the same applications.

and 3.99% were from Africa (Zimbabwe), Asia (Malaysia and Japan), or the Middle East (Jordan). Finally, 87.06% of these were right-handed, 4.98% were ambidextrous, and 99% reported their main smart phone to support touchscreen-only input (one participant relied on a 12-key candy bar-type phone; another owned a phone with both a touchscreen and a physical keyboard).

3.1 Handedness

Using multiple-choice questions, we asked participants how often and why they use one- and two-handed input, or speech (see Figure 2). 48.26% of participants reporting two-handed use *Always* or *Very Often*, and 90.55% relying on speech *Sometimes* or *Never*. Furthermore, 70.65% of participants cited convenience or efficiency when opting for two-handed input, and 23.88% cited context for the use of speech. At the same time, participants reported using one-handed input because it is more convenient (39.30%) or because of the context they are in (26.37%); but only 14.43% and 10.45% because of the device's form factor or interface, respectively.

3.2 Orientation

Similarly, we asked participants how often and why they use their smart phones in landscape and portrait modes (see Figure 3). 93.03% participants reported using their devices in portrait mode *Always* or *Very Often*, citing convenience (55.22%) and interface design (25.87%). On the other hand, 22.89% of participants cited using the landscape mode because it is required by the interface, and 13.93% because of their context.

3.3 Application Usage

Using multiple-choice questions, we asked participants about *how often they use* and how they *prefer to use* their devices across 20 activities or applications (see Figure 4). These questions included common activities such as contact lookup, reading emails, or taking photos and selfies.

Regarding *input modality*, there were very little variations found between use and preference: only 1.79% of participants reported wanting to use one-handed input less often, and only 0.97% reported the need to use two-handed input more often. Noteworthy exceptions include browsing social media (4.98% would rather use one-handed input less often) or music (4.48%), reading the news (5.47%), and watching (7.46%) or recording videos (4.48%). Furthermore, two-handed input was the most common way to engage with video-games (69.15%) and text entry applications such as messaging (67.16%) or email (76.12%), but also other activities such as calendar entries (48.76%), map navigation (43.28%), or recording videos (45.27%) and taking photos (50.75%). Speech was reported to be rarely used as the sole input modality for these 20 activities (0.2%), but several participants highlighted an interest in using it to call a contact (1.49% increase from use to preference) or engage with the map (1.99%).

Similarly, in terms of *device orientation*, little variations were found between use and preference, with only 2.71% of participants reporting to want to use the smart phone in portrait mode less often than they already do. Exceptions included browsing online videos (7.46%), taking selfies (5.47%), using the map application (4.98%), or browsing offline media (4.48%). Furthermore, the landscape mode was the most common way to interact with video-games (45.77%), taking photos (35.32%), and watching (77.61%) and recording videos (39.3%). 4.98% of participants want to use their smart phones in landscape mode more often in the latter activity.

3.4 Discussion

Despite the known limitations of recall surveys, we argue it is important to renew the works of Karlson et al. [32] and Quinn et al. [49] as they are often still used to ground novel one-handed input techniques. Further, this contribution should stand on its own as it offers researchers updated findings on general smart phone use. In the context of our work – i.e., reachability – we highlight a high prevalence of two-handed use due to it being more convenient and efficient than other alternatives, namely one-handed input and speech; while one-handed use was often cited alongside context. Speech was reported to be rarely used by most respondents. The use of smart phones in landscape mode was still imposed on users from time to time: either because of the interface design or the user's context, which in turn tends to require two-handed use for most tasks. For example, our data shows that photo and video recording applications are still quite popular in two-handed and landscape modes. This might be because the user has to delegate various tasks to their hands: framing the picture or video, focusing on the area of interest, adjusting potential parameters (e.g., zooming), and finally snapping the photo or pressing record. Conversely, users rather take selfies in one-handed and portrait modes, arguably because it is a simpler operation with little framing or focusing needed.

In sum, these updated findings continue to illustrate several design opportunities for one-handed input on smart phones. In scenarios where two-handed use was described as being more convenient – for example while snapping a photo – novel one-handed input techniques can attempt to close this usability gap. In scenarios where one-handed use is already prevalent or needed due to a particular context of use, novel one-handed input techniques can attempt to improve the overall user experience by offering more effective interactions. Finally, and while this survey does not cover tablet use, we argue this is another type of mobile device that can benefit from novel one-handed input techniques – particularly techniques such as motion matching that, in principle, show not be affected by target selections even further away from a user's thumb (i.e., *position-independence*). While it is not clear how often users rely on one-handed use while interacting with tablets, several examples illustrate a need for techniques that reduce cursor travel on tablet devices. These include Pfeuffer et al.'s Gaze-Shifting technique [45] and Smith and Schraefel's Radial Scroll Tool [57]: they facilitate access to different points of the

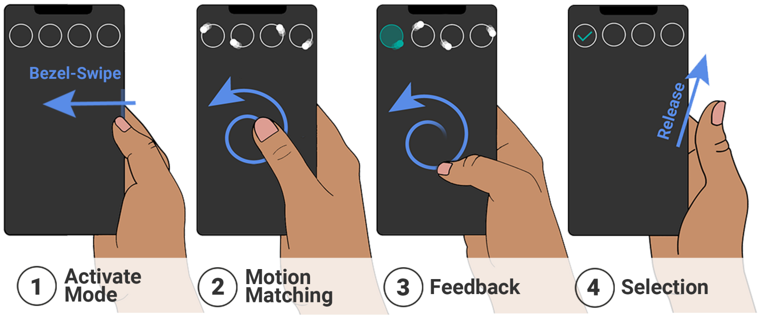


Fig. 5. The four stages of interacting with an interface element using touch-based motion matching: (1) the user performs a bezel swipe to enable the drivers (i.e., the animation that needs to be matched); (2) the user seamlessly transitions to a matching motion using its thumb; (3) if the correlation coefficient between one of the drivers and the user input exceeds a specific threshold, the corresponding target provides feedback to the user; and (4) the user lifts its finger to confirm the selection and disable the mode.

UI or to particular functionality such as scrolling with minimal movement of the main input modality (i.e., the user's finger or pen). As such, the design of our motion matching technique and user study below contemplate both phone and tablet scenarios.

4 DESIGNING MOTION MATCHING FOR TOUCHSCREEN DEVICES

In this section we describe how we expanded the design of motion matching interfaces for embedding into any touchscreen device. We directly build on Orbits controls [18] – arguably one of the most popular motion matching designs – in which interface targets display a distinct animated component moving around their contours (described by Fekete et al. as the *driver* [20]). We extended this design with two new features, illustrated in Figure 5: quickly enabling (or disabling) motion matching drivers on a touchscreen, and displaying a type of *hover state* that enables users to *explicitly* confirm selections. Furthermore, we describe our initial implementation for both phone and tablet devices.

4.1 Enabling (And Disabling) Motion Matching on Touchscreen Devices

Most work in this domain has looked at emerging interactive systems where motion matching is the sole input modality. We are instead interested in exploring how Orbits controls can be embedded into existing mobile interfaces side-by-side with direct touch – so that UI targets in reach of the user's thumb can still be accessed via a simple tap. Visually this is quite straightforward, as minimal changes to the UI are needed to support motion matching (i.e., a driver that navigates the contours of the target it belongs to). The challenge is knowing when to enable these drivers – i.e. when is the user trying to interact with an out-of-reach target – as otherwise these can be visually overwhelming if always present. Previous mode switches have been explored in the context of smart homes, using gaze/attention [19, 63] or an explicit gesture (a flick of the wrist [50, 67]) to enable or disable drivers in a variety of smart devices. On the other hand, several pointing-based one-handed input techniques have looked into implicit indicators that a target is out-of-reach as triggers to the techniques themselves (e.g., the tilting of the device towards the user's thumb [7]).

Design: our concept uses a low-effort switch built around the take-off selection paradigm [48]. We opted for a bezel swipe as it allows the user to immediately transition into matching motion input. As soon as the drivers are enabled and the user identifies the intended one, the swiping motion started at the bezel can quickly turn into a matching attempt. This seamlessly enables the drivers, allows for target selection, and disables them afterwards with little overhead. At the same time, if after the drivers are enabled user change their mind, they can simply swipe backwards to disable them.

4.2 Matching State and Selection Confirmation

Orbits controls have little feedback mechanisms for discrete selections – a selection is triggered immediately after the user has matched the motion of a driver for a short time. This not only avoids further delays, but also stems from earlier systems using gaze or mid-air pointing where supporting explicit selection confirmations is not trivial [29]. While previous work has looked at ideal matching times and correlation thresholds to minimize false positives [68], this is still an issue when multiple drivers exist (with smaller phases between them). Reported error rates are between approx. 10∼20% when as little as eight drivers are present [18, 66]. With the exception of Carter et al. [6], where selections are delayed by one second to provide some user feedback, in most situations the user has no way of knowing which target the system is about to select (and thus correct their input if needed).

Design: we propose a *matching state* feedback mechanism that mimics the *hover state* of mouse-based interfaces. That is, as the user continuously attempts to match the motion of a driver, at particular short intervals the system will change the color of the driver that elicits the minimum correlation coefficient defined. If this matches the user's intent, a simple finger lift triggers a selection and disables the drivers. If it does not, the user can simply adjust its input by slowing it down or picking up speed. This seamlessly feeds into the take-off selection paradigm described in our previous design factor as it closes the loop that started with the bezel swipe that enabled the drivers in the first place. It also addresses Norman's [43] critique of a lack of *feedback* in gestural interfaces.

4.3 Implementation Details

Our motion matching implementation displays drivers with a diameter between 8 (phone) and 12 pts (tablet). Their movement is defined by their initial position on the contour of a UI target (0, 120, or 240 degrees), their speed (2 or 5 deg./sec.), and one of two directions. To implement the *matching state* described above, we continuously collect the coordinates of all drivers and touch input after the user has enabled these via a bezel swipe (identified via a 5pt invisible border on the right and left sides of the phone or tablet). We perform a Pearson's correlation at 400 ms intervals and highlight the target with the highest correlation coefficient over 0.8, if any (green fill, border, and driver). We then empty all data arrays before starting a new matching interval. 400 ms is shorter than the state-of-art – normally between 500 [68] and 2000 ms [6] – because we are not triggering selections solely based on matching time, and, thus, are not concerned about false positives. The correlation coefficient of 0.8 follows similar implementations and results from our own informal pilot sessions. Finally, a finger lift selects the highlighted target in the latest interval (if any), and disables the mode. If not in this highlighted/matching state, the driver has the same color as the target border.

5 USER STUDY

Motion matching studies tend to follow one of two approaches: either the selection task is designed to explore the characteristics of these types of interfaces (e.g., how many targets can be superimposed before their animations become too overwhelming or indiscernible for the user [18]); or they follow the standard selection task of traditional pointing-based studies so that we learn about motion matching in the context of traditional interfaces (at the expense of not fully eliciting the benefits of this novel approach [17]). Because we are interested in looking at motion matching and Orbits controls in the context of mobile devices, we have opted for the latter.

5.1 Participants

We recruited 24 participants (8 female), aged between 18 and 56 (M = 28.83, SD = 7.33). The majority of participants were employed (14) and lived in *country anonymized* (17); 21 were right-handed, and one was ambidextrous. Using a multiple-choice grid, 62.5% of participants reported using one-handed input *Always* or *Very Often*, 41.67% using two-handed input *Very Often* or *Half of the time*, and 95.83% using speech *Sometimes* or *Never*. Similarly, 91.67% of participants reported using their phones in portrait mode *Always* or *Very Often*, and 66.67% in landscape mode *Sometimes* or *Never*. Finally, using a 5-point Likert scale, participants reported having little experience with one-handed input techniques such the iOS Reachability mode (M = 1.75, SD = 0.99).

5.2 Experimental Design

The study included two mobile devices: a smartphone and a tablet. We decided to invite participants to perform a similar task on a tablet to exacerbate the distance between the intended target and participants' thumbs, and thus explore the *position-independent* property of motion matching interfaces in the context of touch input. This also allows us to explore motion matching in touch-based scenarios with larger displays where minimal cursor travel might be desired [41, 45, 57]. Half of participants started by interacting on the phone and then used the tablet (and vice-versa).

The study followed a within-subjects design with two independent variables: *input technique* and *target location*. The order in which they experienced the input techniques was entirely counterbalanced. Target locations were presented in a randomized order. Varying driver speeds were used to facilitate discrimination between targets and not as a study factor (i.e., participants always interacted with a driver moving at 2 deg. / sec).

5.3 Input Techniques

In addition to a *direct touch* baseline and the *motion matching* implementation described earlier, we implemented the BezelCursor [38], a state-of-the-art input technique for one-handed input and reachability. This resembles our technique in how it is enabled (bezel swipe) and how targets are selected (finger lift), and still outperforms more recent techniques in this domain across a wide range of metrics (e.g., Corsten et al. [11]). In sum, after the user performs a bezel swipe (in either side of the device, on a 20 pt invisible border), a red line (3 pt width) grows linearly by a factor of three (phone) or four (tablet) in the direction of the thumb movement. When the end point of this line is within the boundaries of a target this turns green (fill and border). A finger lift selects the highlighted target (if any) and disables the mode.

5.4 Target Locations

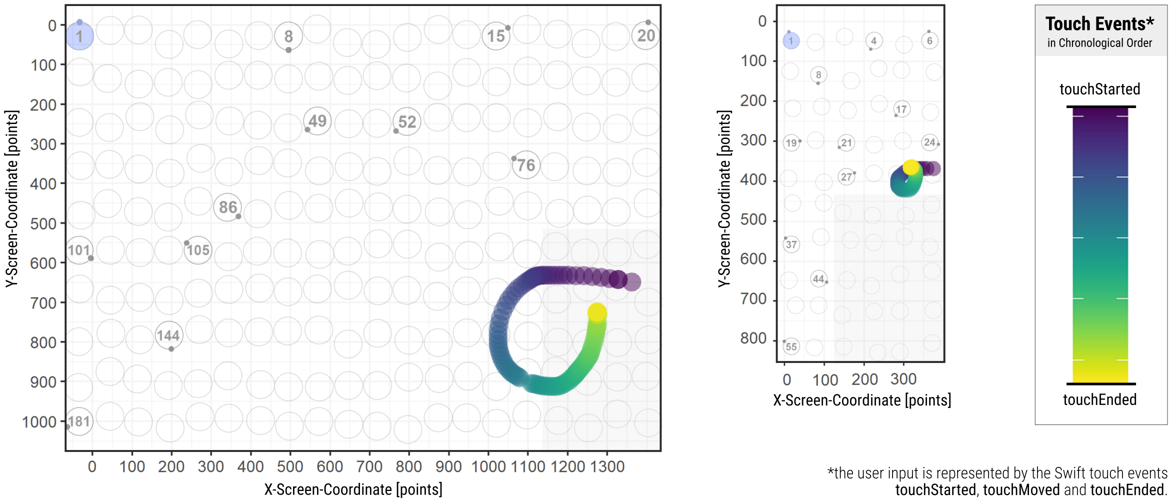
Each study block comprised of 12 subsequent target selections, with six displayed at the *edge* and six closer to the *center* of the display, in a randomized order (Figure 6). The target to be selected next was displayed with a blue background and blue border. Correct selections produced sound *1002* from the iOSSystemSoundsLibrary[3]; incorrect selections resulted in sound *1006*. An incorrect selection would move the target to the end of the block until all 12 targets had been correctly selected.

5.5 Experimental Setup and Task

Our study was implemented in Xcode using Swift for both mobile devices: a 5.8-inch iPhone XS (375 × 812 pt), and a 4th generation 12.9-inch iPad Pro (1366 × 1024 pt). Each displayed a target-dense selection task representative of browser interactions (which 63.68% of our survey participants perform on their phones with one-hand). Our prototypes ran at approximately 60 Hz.

---

[3] https://github.com/TUNER88/iOSSystemSoundsLibrary



*the user input is represented by the Swift touch events touchStarted, touchMoved and touchEnded.

Fig. 6. Target layout during tablet (left) and phone (right) conditions. *Edge* targets are represented by IDs 1, 8, 15, 20, 101, and 181 in the tablet condition; and IDs 1, 4, 6, 19, 37, and 55 in the phone condition. The remaining IDs are considered *center* targets. In these two trials participants had to select the target represented in blue (ID 1). Target IDs and grid were not shown during the trials, and the figure was purposely made slightly transparent after being screenshot to highlight P6 and P7's touch heat maps during a motion matching trial. These are color coded from purple (touch started) to yellow (touch ended), and highlight the *scale-independent* property of motion matching input where, regardless of target size, participants' matching attempts can be as large (left) or as small (right) as its more comfortable to them.

5.5.1 Phone. Interactions with the phone were done in portrait mode (93.03% of survey participants reported using their phones in this mode *Always* or *Very Often*). 60 circular targets with a diameter of 40 pt were displayed in an invisible grid of 6 × 10 tiles, each measuring 62.5 × 76.8 pt. 40 pt is a representative target size (approximately 6.4 mm), with the latest iOS guidelines describing targets ranging from 30 to 60 pt[4]. The 12 targets to be selected (black border) were displayed at the center of the tiles, and the remaining distractors (grey border) were positioned in random locations within the tile so as to avoid a regular looking arrangement [30]. Finally, because we are only interested in interactions with out-of-reach targets, the distractors in the 4 × 5 grid at the bottom right of the display (i.e., in reach of the thumb) were not selectable during the trials. Participants completed this task across 216 trials: three input techniques (touch, BezelCursor [38], motion matching) × 12 targets × six blocks (5184 trials in total across 24 participants).

5.5.2 Tablet. Participants performed a similar task on a tablet device in landscape mode. This implementation was very similar to the above, with the following exceptions. We now had 200 targets with a diameter of 60 pt (approximately 9.6 mm), across a grid of 10 × 20 tiles (68.3 × 98 pt each). Because one-handed direct touch using the thumb is not feasible in this scenario, participants completed this task across 144 trials: two input techniques (BezelCursor, motion matching) × 12 targets × six blocks (3456 trials in total).

5.6 Procedure

The study sessions took part in a quiet environment. Participants completed the study while sitting, and started by disinfecting their hands and filling in the consent and demographics forms. When starting a new input

---

[4] https://developer.apple.com/design/human-interface-guidelines/ios/icons-and-images/app-icon/

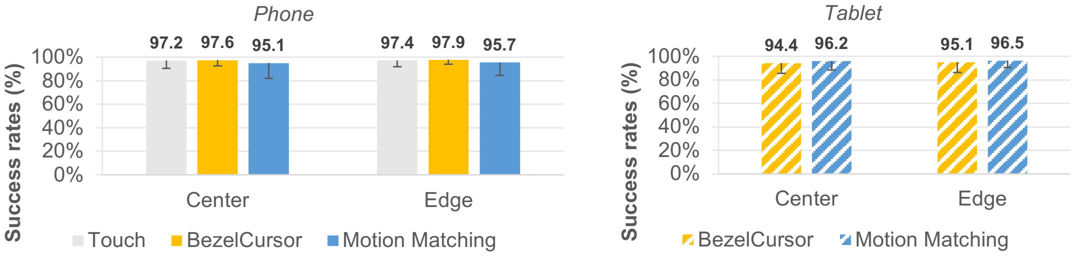


Fig. 7. Success rates for targets closer to the center or edge of the display, in both the phone (left) and tablet (right) conditions. Standard deviation in error bars.

technique, the researcher would demonstrate it for a block (12 trials), and then ask participants to try them for two practice blocks. When repeating the same technique on a new device (phone or tablet), participants were asked to complete one practice block. Furthermore, at the end of each block, participants were free to take a break and re-start when ready. Participants were asked to hold the phone without supporting their arms on the table. The tablet laid on the table at a 60 degree angle (using an iPad cover) – this enabled participants to operate the device using one hand, but not feel its weight. Finally, participants completed several subjective metrics at the end of each input technique and at the end of the study. At the end of the session the researcher disinfected both the phone and tablet devices.

5.7 Metrics

We characterize our motion matching implementation via standard performance metrics (success rates, selection times), in addition to its input footprint (i.e., touch heatmaps) and grip stability. The latter was inspired by Eardley et al. [13, 15], where the device's rotation is captured in degrees for the x-, and y-, and z-axis (the sum of each per trial). Finally, we asked participants to complete a subjective workload assessment using the NASA-TLX [26]. At the end of the study participants reported and reasoned on their favorite and least favourite techniques.

5.8 Results

Before starting the analysis we discarded a third of the blocks as practice (the first two out of every six). As performance data was not normally distributed (verified via Shapiro-Wilk tests), we analyze it with Aligned Ranks Transformation ANOVAs [71]. Participants' subjective workload data was analyzed via related-samples Friedman tests and post hoc Wilcoxon signed-rank analysis (phone), and Mann-Whitney U tests (tablet).

5.8.1 Success rates. These results are summarized in Figure 7. Success rates on the phone varied between 95.14% (*SD* = 13.10, motion matching – center) and 97.92% (*SD* = 3.69, BezelCursor – edge), with no significant differences found for input technique (F(2) = 0.04, p = .964), target location (F(1) = 0.70, p = .406), or for the interaction between the two (F(2) = 0.06, p = .945). Similarly for the tablet, success rates varied between 94.44% (*SD* = 8.92, BezelCursor – center) and 96.53% (*SD* = 5.85, motion matching – edge), with no significant differences found for input technique (F(1) = 1.22, p = .273), target location (F(1) = 0.03, p = .863), or for their interaction (F(1) = 0.74, p = .393). Success rates for phone and tablet were not normally distributed for all variables ($p < .001$).

5.8.2 Selection times. These results are summarized in Figure 8. Selection times on the phone varied between 1.38 s (*SD* = 0.79, touch – center) and 1.88 s (*SD* = 1.55, motion matching – edge), with no significant differences found for input technique (F(2) = 0.36, p = .697) or for the interaction between input and location (F(2) = 0.20, p = .821). A significant effect of target location was found (F(1) = 10.70, p = .001). Selection times on the tablet varied

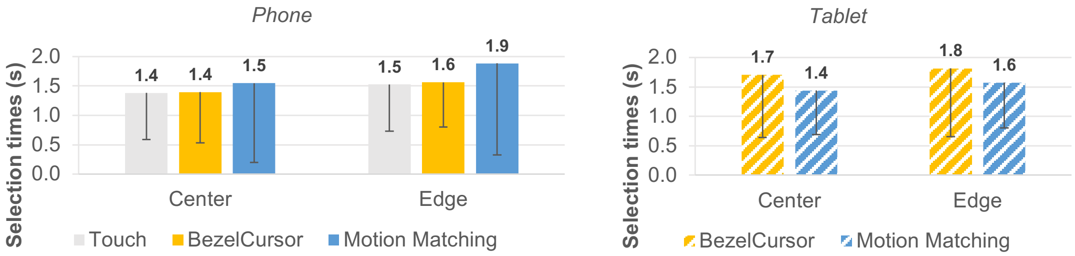


Fig. 8. Selection times for targets closer to the center or edge of the display, in both the phone (left) and tablet (right) conditions. Standard deviation in error bars.

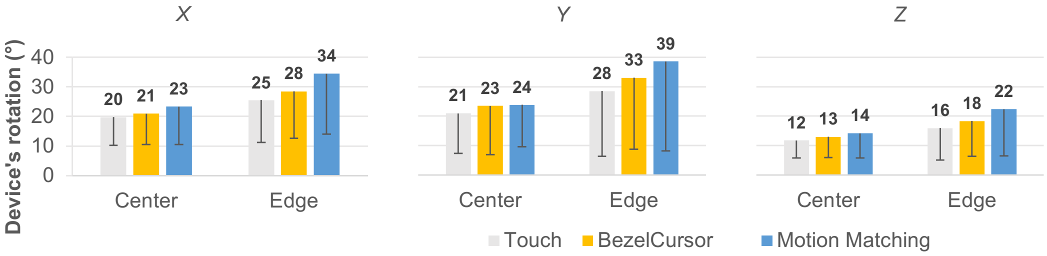


Fig. 9. Grip stability via the device's rotation, captured in degrees for the x-, and y-, and z-axis. This aspect was only assessed for phone interactions, as it was held mid-air. Standard deviation in error bars.

between 1.44 s (motion matching – center) and 1.82 s (BezelCursor – edge), with no significant differences found for input technique (F(1) = 0.69, *p* = .410), target location (F(1) = 1.09, *p* = .300), or for the interaction between the two (F(1) = 0.06, *p* = .815). All but one of the selection times for the phone and tablet were not normally distributed (*p* < .012).

5.8.3 Grip stability. Grip stability was only assessed during phone interactions, as it was held in mid-air – the results can be found in Figure 9. The phone's rotation varied between 19.78 (*SD* = 9.60, touch – center) and 34.40 degrees (*SD* = 20.32, motion matching – edge) on the x-axis, with no significant differences found for input technique (F(2) = 2.18, *p* = .118) or for the interaction between input and location (F(2) = 0.620, *p* = .540). A significant effect of location was found (F(1) = 13.44, *p* < .001). Similar results can be observed for the y-axis, with phone's rotation varying between 20.90 (*SD* = 13.51, touch – center) and 38.61 degrees (*SD* = 30.48, motion matching – edge). No significant differences were found for input technique (F(2) = 1.43, *p* = .243) or for the interaction between input and location (F(2) = 0.32, *p* = .730), but a significant effect of target location was found (F(1) = 8.12, *p* = .005). Finally, the phone's rotation on the z-axis varied between 11.74 (*SD* = 5.94, touch – center) and 22.37 (*SD* = 15.93.32, motion matching – edge) degrees, with no significant differences found for input technique (F(2) = 2.43, *p* = .093) or for the interaction between input and location (F(2) = 0.66, *p* = .520). A significant effect of target location was found (F(1) = 11.26, *p* = .001).

Rotation data was not normally distributed for half of the variables on the x-, y-, and z-axis (*p* < .046).

5.8.4 Subjective Workload. The results for the subjective workload (NASA-TLX), in a scale of 0 to 20, can be found in Figure 10. Significant differences were found between touch and motion matching for mental demand

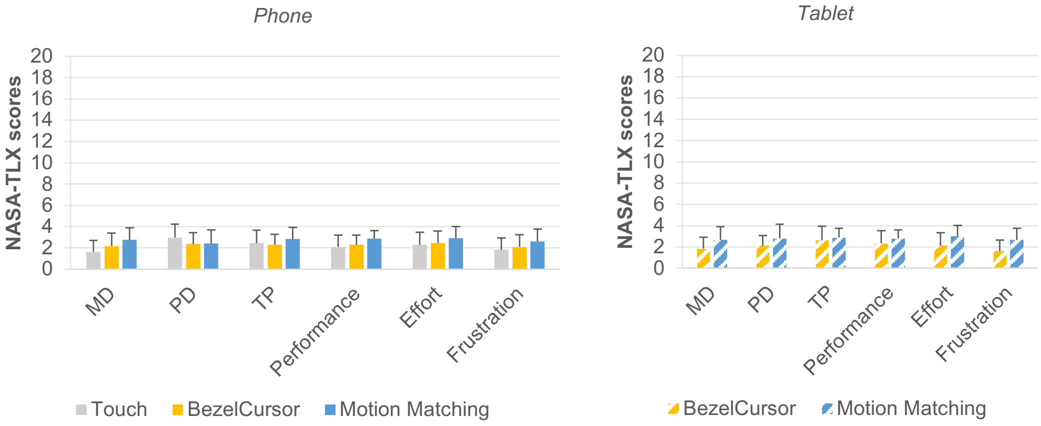


Fig. 10. Subjective workload via NASA-TLX scores (lower is better), for both phone (left) and tablet (right) conditions. MD, PD, and TP represent mental, physical, and temporal demand – respectively. Standard deviation in error bars.

(means of 1.63, *SD* = 1.10, and 2.75, *SD* = 1.15, respectively; $\chi^2(2) = 17.45$, $p < .001$; Z = -3.07, $p = .009$), performance (means of 2.08, *SD* = 1.14, and 2.88, *SD* = 0.74, respectively; $\chi^2(2) = 15.44$, $p < .001$; Z = -3.34, $p = .003$), and frustration (means of 1.83, *SD* = 1.09, and 2.63, *SD* = 1.13, respectively; $\chi^2(2) = 8.27$, $p = .016$; Z = -2.57, $p = .030$). Significant differences were also found between the BezelCursor and motion matching for mental (means of 2.17, *SD* = 1.24, and 2.75, *SD* = 1.15, respectively; $\chi^2(2) = 17.45$, $p < .001$; Z = -2.40, $p = .048$) and temporal demand (means of 2.29, *SD* = 1.00, and 2.83, *SD* = 1.09, respectively; $\chi^2(2) = 9.16$, $p = .010$; Z = -2.59, $p = .030$). Finally, the input techniques had significant effects on effort ($\chi^2(2) = 11.03$, $p = .004$) – means between 2.29 (*SD* = 1.20, touch) and 2.92 (*SD* = 1.10, motion matching) – but not on physical demand ($\chi^2(2) = 2.62$, $p = .270$) – means between 2.42 (*SD* = 1.28, motion matching) and 2.96 (*SD* = 1.27, touch).

Regarding the tablet, no significant effect was found for the two input techniques in terms of physical (U = 208, z = -1.71, $p = .088$) or temporal demand (U = 249.5, z = -0.82, $p = .411$). Conversely, a significant effect was found for ease of use (U = 107, z = -3.88, $p < .001$), effort (U = 151, z = -2.93, $p = .003$), frustration (U = 128.5, z = -3.44, $p = .001$), learnability (U = 111, z = -3.96, $p < .001$), mental demand (U = 174.5, z = -2.43, $p = .015$), performance (U = 197, z = -2.02, $p = .043$), precision (U = 90, z = -4.23, $p < .001$), and speed (U = 140, z = -3.18, $p = .001$).

5.8.5 Preference. Participants' preferences can be seen in Figure 11. During phone interactions, motion matching was the preferred input technique of five participants, describing it not only interesting (1), fast (1), easy to use (1), and precise (1), but also as less fatiguing (1) and requiring a smaller input area (1). Touch and BezelCursor were the preferred input method by 11 and 8 participants, respectively. These described touch as fast (7), easy to use (5), and precise (3). Likewise, the BezelCursor was described as fast (4), precise (3), easy (2), and comfortable to use during one-handed input (2). In contrast, 15 participants described motion matching as their least favorite input method, followed by the BezelCursor (6). The former described various hurdles such as needing more practice (8), how it can be slow (3), frustrating (3) and uncomfortable (2), and being just "awkward" (P17). The latter described the BezelCursor as being slow (2) and uncomfortable (2), requiring some concentration (1), and inconvenient for larger displays (1). Finally, touch input was also described as inconvenient (1), slow (1), and the cause of occlusion on the bottom half of the display (1).

Regarding tablet interactions, motion matching was the favorite input technique of 7 participants, describing it as easy (1), precise (1), comfortable (1), "incredibly fast" (P7), and "like a (...) game" (P2). P1 stated that "after

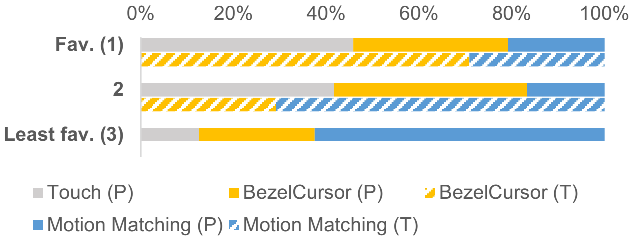


Fig. 11. Preference rankings in both the phone (P) and tablet (T) conditions.

getting practice (...) one could hit the targets quite fast". The BezelCursor was the favorite input method of the other 17 participants, describing it as easy to use (9), fast (8), precise (3), and less physically demanding (1). These participants described motion matching as "hard to master" (P10), slow (5), tiring (2), frustrating (2), and overall requiring more practice (8). The BezelCursor was described by seven participants as slower (3), uncomfortable (3), and frustrating (2).

5.9 Discussion and Limitations

The goal of our work was to design and develop the first motion matching implementation using touch, and to characterize its use in the context of one-handed input. To better contextualize our results (i.e., user performance and feedback), we compared it to two traditional pointing-based approaches – direct touch and the BezelCursor – in a standard selection task. The results achieved are quite positive: despite the novelty of our selection mechanic and the minimal amount of practice afforded to participants (24 trials), motion matching did not *seem* to affect participants' success rates, selection times, or grip stability when compared to either our baseline (direct touch) or state-of-art implementation (BezelCursor). While this was also true for some subjective measures such as physical demand, and despite the overall low workload scores across conditions – 2.96 on the phone (touch, physical demand) and 3.00 on the tablet (motion matching, effort) – a significant effect of the motion matching approach was still found for effort and mental demand when compared to its pointing-based peers.

While this was the case, when discussing their preference for the techniques participants reported that they simply needed more practice – this was the reason provided by half of participants who described motion matching as their least favorite technique. This might also explain the high variance observed for motion matching on, for example, selection times or grip stability. This kind of issue is not new when introducing a novel interaction technique, and we look forward to revisiting these results once longer or longitudinal studies are conducted – for example, by following Corsten et al.'s [11] methodology where participants complete four study sessions per day across multiple consecutive days. These sessions could also be carried out with considerably larger phones (e.g., the iPhone 13 Pro or the Galaxy Z Fold 3) in order a have broader understanding of the differences between motion matching and direct touch on, for example, grip stability or physical demand.

It is also worth framing our results in the context of other motion matching implementations. The *matching state* and *explicit* selection confirmation steps introduced are promising: participants were able to select 12 targets with an error rate of ~5%, while other work using Orbits controls and a similar implementation (Pearson's correlation, correlation threshold, and target speeds) describe error rates between 10 and 30% when eight targets or more are presented. Selection times did not seem to be negatively affected by the two additional selection steps: we report a mean selection time across device and target location conditions of ~1.6 s (which includes the time it took participants to switch on the mode). Previous work relying on head- and hand-based input report on selection times between ~1.6 and ~2.2 s for interfaces involving eight targets or less [6, 17, 66]. While these are encouraging findings, further work is needed to assess the effects of our low-noise input channel (i.e., touch) on these results. Further work should also be carried out to explore other potential mode switching mechanisms (i.e., other than the bezel swipe), other types of *matching state* feedback (e.g., haptic feedback), and other target speeds, correlation parameters, or even input matching algorithms (e.g., [12, 62]) that might even be better suited for the particular type of visual-motor coordination taking place during touch-based input. We opted to purposely focus on a study factor that is particular to mobile devices – device type (or orientation) – instead of iterating once again over, for instance, target speeds [9, 66] or driver trajectories [6, 22] in the context of manual input. That being said, we are also intrigued by how many more drivers could have been added to our study before we saw a significant effect on user performance, and the overall performance of the technique while walking.

We also want to revisit the known affordances of motion matching that we argued could be interesting in the context of mobile interface design. That is, how targets in a motion matching interface, unlike pointing-based approaches, should be *position-* and *scale-independent*. The first was assessed via the target location condition (center and edge), but we were not able to observe any positive effects of using motion matching to select targets that were displayed further away – selection times and grip stability continued to be negatively affected by such targets on the phone. At the same time, and while no significant differences were found between the BezelCursor and motion matching on the tablet condition, a trend is visible in Figures 7 and 8, showing fewer errors, shorter selection times, and less data variance in the motion matching condition. Furthermore, while participants' selection times seem to have worsen during the BezelCursor condition between phone (M = 1.48 s) and tablet (M = 1.76 s), the same did not happen for motion matching (phone: M = 1.71 s, tablet: M = 1.51 s). In any case, the successful one-handed use of motion matching with the tablet, where touch is not viable, positions this as an interesting solution for future devices with larger touch input areas (e.g., smart surfaces). Future work should continue to explore this *position-independence* property, first observed for head-pointing systems [17] where the visual-motor coordination between eye and head and eye and hand are inherently different.

Finally, and because our study was designed around a standard selection task that ultimately represents an abstraction of current mobile interfaces, we were not able to systematically assess the *scale-independent* affordances of motion matching in the context of touch input. The target sizes used followed standard iOS UI guidelines that prioritized comparison to our baselines. As such, further work is required to fully assess these affordances. While that is the case, we were able to observe this affordance in action as illustrated in Figure 6, where the range of participants' input motions varied broadly independent of target size – from motions taking place in an area as small as 1.7 × 1.7 cm, to an area as large as 6.7 × 6.7 cm.

6 INTEGRATING MOTION MATCHING WITH A CAMERA APPLICATION

In this section we propose different ways in which Orbits controls can be leveraged in mobile applications for more than atomic selection tasks. That is, how can touch-based motion matching support sequential selections, menu-based interactions, or provide continuous controls over UI parameters that might be hard to reach. We provide six examples that often combine Orbits controls with more standard input techniques (e.g., swiping) or UI elements (e.g., carousel), and invite the community to expand on this design space in the future. Our examples aim to improve the user experience of a classical camera application, often used in one-handed mode while recording videos or taking selfies – as reported by 41.29% and 78.61% of our survey participants, respectively. A summary of these examples, together with excepted benefits and limitations, can be seen in Table 1.

Our camera application was developed with Xcode and Swift, and supports the following functions and options displayed at the top of the screen: flash (on, off, auto), image ratio (1:1, 4:3, 16:9), shutter timer (from off to 10 s), and filter (off, sepia, b&w, etc). Distinct haptic feedback is played when a new matching state is found (in addition to a visual confirmation as before), when cycling through options, and when a selection is confirmed.

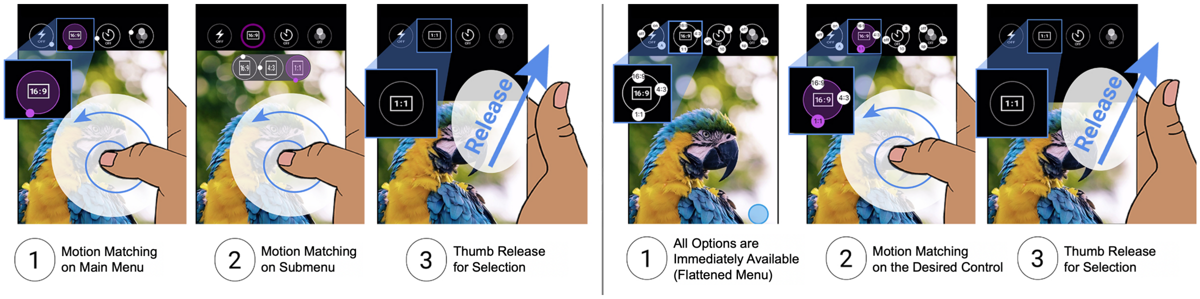


Fig. 12. Two examples of how sequential selections could be made using Orbits controls. On the left is what we describe as an *Orbits Menu*: instead of asking the user to enable the mode and confirm selection twice, after the user successfully matches the motion of a driver, and without lifting its finger, a new set of Orbits controls are displayed. At this point the user can simply adjust its thumb movement so it matches one of the new drivers. On the right is what we describe as *Always-on Orbits*, which leverages the *space-independent* property of motion matching interfaces. In this example, each function is represented by an Orbits control with as many drivers as options available. These offer a non-abstract representation of their function and allow the user to immediately select the intended option (e.g., an image ratio of 1:1) without having to select the desired function first (i.e., image ratio).

6.1 Orbits Menu vs Always-on Orbits

Our first example illustrates how two sequential selections could be made using Orbits controls. Instead of asking the user to enable the mode and confirm selection twice, after the user successfully matches the motion of a driver for 1700 ms (i.e., the mean selection time for edge target across phone and table conditions), and without lifting its finger, a new set of Orbits controls are displayed. At this point the user can simply adjust its thumb movement so it matches one of the new drivers (inspired by the take-off selection paradigm [48]), with a final selection being issued by a finger lift. In Figure 12 (left) we illustrate this by having a driver in each function. For example, after matching the image ratio function, three new Orbits controls are displayed just below: one for each of the function's options (i.e., 1:1, 4:3, 16:9).

A more immediate approach leverages another property of motion matching interfaces we did not study (again to prioritize comparisons to our baselines): as long as the animations are perceivable and distinct, targets can be displayed in very close proximity or even superimposed onto one another (*space-independent* [55]). In this example, each function is represented by an Orbits control with as many drivers as options available – with the exception of the filter function, where we represent three potentially popular options (so that the maximum number of drivers on-screen is 12). Finally, each driver has a non-abstract representation of its function (e.g., "on"). In Figure 12 (right) we illustrate this by showing how a user could immediately select the intended image ratio of 1:1 without having to select the ratio function first.

6.2 Orbits Controls with Swiping and Dragging Input

Our next two examples continue the exploration above on sequential or menu-based selections – this time combining Orbits controls with more standard input techniques. The first example has the user match the motion of a driver for 1700 ms, after which up to four buttons or targets are displayed around that particular Orbits control. The user performs a selection by simply swiping and releasing the finger in the direction (up, down, left, or right) of the intended option – no selection is issued if the user releases the finger without performing a swipe. In the example in Figure 13 (left), the user has matched the motion of the driver for the image filter function and immediately swiped right to select the sepia filter.

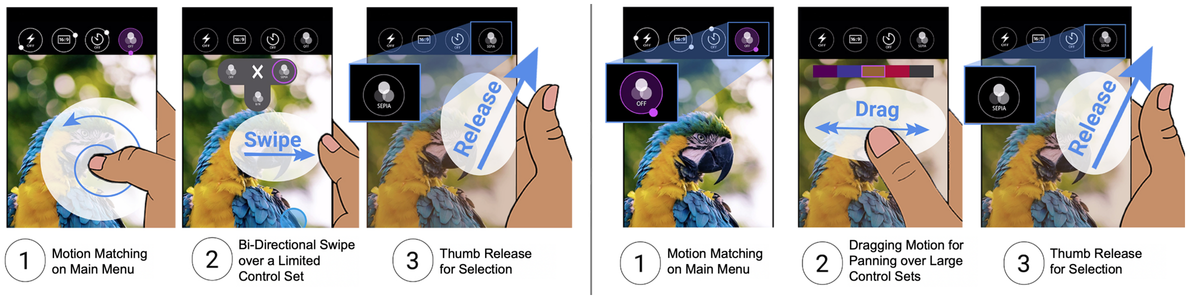


Fig. 13. Two examples of how Orbits controls can be combined with more standard input techniques. On the left we illustrate a pairing with *Swiping Input*, where after the user matches the motion of a driver up to four new targets are displayed around that particular Orbits control (e.g., the image filter). The user performs a selection by simply swiping and releasing the finger in the direction of the intended option. On the right we illustrate a pairing with *Dragging Input*, enabling the selection of a broader set of options. In this example we display a carousel UI next to an Orbits control after the user has matched its driver. The first option of the carousel is the current setting, and users pan around by dragging their fingers to the left or right.

The second example looks at how to enable the selection of more options (> 4). Inspired by Clarke and Gellersen [10] where a pointer is displayed after a motion matching selection via mid-air pointing, in our example we display a carousel UI next to an Orbits control after the user has matched its driver for 1700 ms. The first option of the carousel is the current setting, and users pan around by dragging their fingers to the left or right. Users control the panning speed by dragging their fingers further or closer to their starting positions. In the example in Figure 13 (right), the user has selected the filter function via motion matching, after which a carousel with several filters is displayed as colored rectangles. As the user pans through these options, the resulting image and filter is displayed for preview – lifting the finger sets the filter.

6.3 Continuous Orbits vs Rotary Input

In our final two examples we explore how Orbits controls can support continuous input of an option such as the shutter timer. This approach was first introduced by Esteves et al. [18] for gaze-based motion matching, and it simply increments (or decrements) a value for as long as the user is matching the motion of the appropriate driver. In Figure 14 (left) we illustrate how a user can adjust the shutter timer in this manner: the first 1700 ms of motion matching selects the shutter function; every other 1000 ms increments this value up to 10 s, if and while the user maintains the correct matching motion. After reaching 10 s the shutter option cycles back to zero (i.e. off).

Another approach – one that does not require the user to wait for the right input to cycle by – displays a standard knob UI after the initial motion matching selection of 1700 ms. After this, the user's input matches that of a rotary dial in which she/he is free to move their finger to the desired shutter option (e.g., 3 s) – moving the finger clockwise increases this value; moving it counterclockwise decreases it (see Figure 14 – right). We selected a knob as this requires a circular motion, being in many ways similar to that of the motion matching input that precedes it. This method was inspired again by the take-off selection paradigm.

In both examples haptic feedback enables eyes-free input after the initial motion matching selection.

7 FUTURE WORK

In addition to future work described in previous sections, we highlight two more ways in which this work can be expanded upon. The first is by complementing the growing body of work on the use of motion matching to control IoT devices. The approach we propose via touchscreen devices, such as the phone or even the watch (using the index finger), would make for a more widely available solution compared the eye-trackers proposed by

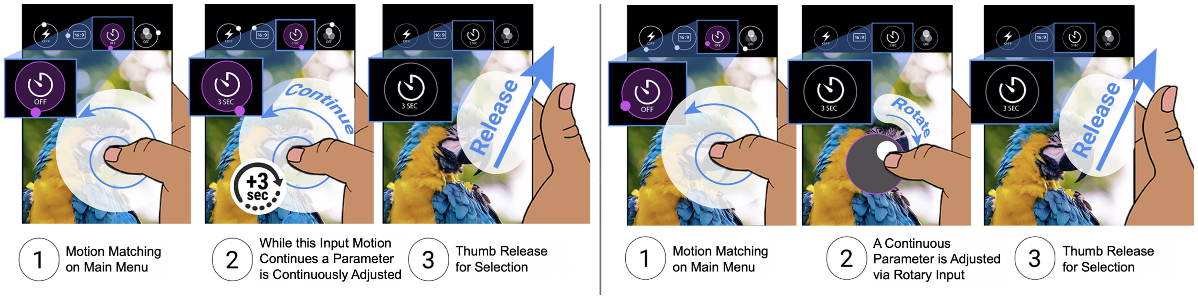


Fig. 14. Two examples of how Orbits controls can support *continuous input*. On the left we illustrate what we call *Continuous Orbits*, where a user increments (or decrements) a value for as long as she/he is matching the motion of the appropriate driver (at 1 s intervals). On the right we illustrate a *Rotary Input* that does not require the user to wait for the right input to cycle by. It displays a standard knob UI after the initial motion matching selection; after this, the user's input matches that of a rotary dial in which she/he is free to move its finger to the desired option (e.g., a 3 s shutter timer).

Velloso et al. [63]. Furthermore, it arguably raises less privacy concerns and field-of-view issues than the optical sensors used by Clarke et al. [8–10]. Finally, because it should not be affected by the gorilla arm [5], the approach would arguably be less tiring than the mid-air motion gestures proposed in previous works [16, 50, 64–67] that are sensed via the inertial measurement units found on most smart watches. To this effect, we are naturally quite interested in validating our approach using the touchscreen on smart watches, particularly as participants were able to perform input motions in an area as small as 1.7 × 1.7 cm (Figure 6 – right). We should also compare it to other motion matching approaches in this domain such as SeeSaw [72] or SynchroWatch [51].

This thinking leads us to the second way in which this work can be expanded upon: the use of touch-based motion matching on smart watches for augmented-reality (AR) interfaces. AR headsets in particular are an emerging device platform without a standard input modality. Speech [42], on-device input, or mid-air gestures [28] still present social acceptability issues when used in public [52]. The latter can also present privacy concerns when gestures are optically tracked [24]. On the other hand, subtle gaze-based approaches using a clicker [25] or dwell time [44] for selection confirmation are not without their own limitations: the former is another device users need to carry and maintain. The latter is still often affected by what is known as the 'Midas touch' – unintentional target activations during exploration or natural pauses in gaze motion [17].

In sum, our approach would rely on a readily available device (i.e., a smart phone or smart watch), highly inconspicuous input that could resemble nothing more than idle fidgeting [1], and could be used to interact with overlapping AR UI elements, as demonstrated in a similar gaze-based motion matching approach in virtual-reality [55]. Furthermore, AR would be an ideal scenario in which to continue to explore the *position-independent* property of motion matching interfaces – a trend we observed even within the constrained space of our tablet condition (see Figure 8).

8 CONCLUSION

In this paper we have introduced the first touch-based interaction technique based on motion matching, a relatively new interaction paradigm that relies not on pointing but on mimicking a UI targets' unique animations. We have expanded on the design of Orbits controls so these take advantage of the affordances of touch, particularly a user feedback mechanism and selection confirmation. Furthermore, we have characterized the performance of this technique via a standard reachability study using a phone and tablet form factors. Despite a clear need for more practice, participants were able to perform in a manner comparative to direct touch or a state-of-the-art

Table 1. Summary of the design space enabled by motion matching and Orbits controls.

	Combines Orbits controls with...	**Selection type**	**Strengths**	**Potential drawbacks**
Orbits Menu	Orbits controls	Discrete	Consistent multi-level selections	Slow selections (at least 2 × 1700 ms)
Always-on Orbits		Discrete	Immediate access to all options (flattened menus)	Visually and mentally demanding
Swiping Input	Swiping gestures	Discrete	Fast selection of up to four options	Hard to scale
Dragging Input	Dragging; Carousel UI	Discrete	Quick preview of a wide range of options	Frustrating as it requires both circular and horizontal input motions
Continuous Orbits		Continuous	Continuous and consistent input motions	Frustrating if cycling times are set too low or too high
Rotary Input	Knob UI	Continuous	Direct control of continuous options	Not eyes-free if non-numerical options are used

implementation (i.e., the BezelCursor). Unlike the latter, motion matching performance was not affected when participants moved from the phone to the tablet – a scenario where one-handed direct touch is no longer feasible for most operations.

Furthermore, our study also suggests that the user feedback mechanism and selection confirmation introduced in this work enabled our motion matching implementation to outperform similar state-of-the-art implementations using, e.g., gaze or mid-air input. As such, we conclude our work with a brief design exploration, describing (a) how motion matching can be combined with other established input techniques and UI elements and (b) how other areas can benefit from our findings (e.g., augmented-reality). In addition we contribute a contemporary survey on mobile use directly inspired by Karlson et al. and Quinn et al. We hope that these findings can motivate further work in this area.

ACKNOWLEDGMENTS

This project was funded by the Fundação para a Ciência e a Tecnologia and LARSyS (grant UIDB/50009/2020).

REFERENCES

[1] Fraser Anderson, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2015. *Supporting Subtlety with Deceptive Devices and Illusory Interactions.* Association for Computing Machinery, New York, NY, USA, 1489–1498. https://doi.org/10.1145/2702123.2702336

[2] Alessio Bellino. 2018. SEQUENCE: A Remote Control Technique to Select Objects by Matching Their Rhythm. *Personal Ubiquitous Comput.* 22, 4 (Aug. 2018), 751–770. https://doi.org/10.1007/s00779-018-1129-2

[3] Peter Bennett, Stuart Nolan, Ved Uttamchandani, Michael Pages, Kirsten Cater, and Mike Fraser. 2015. Resonant Bits: Harmonic Interaction with Virtual Pendulums. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '15).* ACM, New York, NY, USA, 49–52. https://doi.org/10.1145/2677199.2680569

[4] Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. 2014. Modeling the functional area of the thumb on mobile touchscreen surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* Association for Computing Machinery, New York, NY, USA, 1991–2000. https://doi.org/10.1145/2556288.2557354

[5] Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (OZCHI '09).* ACM, New York, NY, USA, 161–168. https://doi.org/10.1145/1738826.1738853

[6] Marcus Carter, Eduardo Velloso, John Downs, Abigail Sellen, Kenton O'Hara, and Frank Vetere. 2016. PathSync: Multi-User Gestural Interaction with Touchless Rhythmic Path Mimicry. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3415–3427. https://doi.org/10.1145/2858036.2858284

[7] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1499–1508. https://doi.org/10.1145/2702123.2702425 event-place: Seoul, Republic of Korea.

[8] Christopher Clarke, Alessio Bellino, Augusto Esteves, and Hans Gellersen. 2017. Remote Control by Body Movement in Synchrony with Orbiting Widgets: An Evaluation of TraceMatch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3 (Sept. 2017), 45:1–45:22. https://doi.org/10.1145/3130910

[9] Christopher Clarke, Alessio Bellino, Augusto Esteves, Eduardo Velloso, and Hans Gellersen. 2016. TraceMatch: a computer vision technique for user input by tracing of animated controls. In *UbiComp '16: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM Press, 298–303. https://doi.org/10.1145/2971648.2971714

[10] Christopher Clarke and Hans Gellersen. 2017. MatchPoint: Spontaneous Spatial Coupling of Body Movement for Touchless Pointing. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 179–192. https://doi.org/10.1145/3126594.3126626

[11] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. 2019. ForceRay: Extending Thumb Reach via Force Input Stabilizes Device Grip for Mobile Touch Input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 212:1–212:12. https://doi.org/10.1145/3290605.3300442 event-place: Glasgow, Scotland Uk.

[12] Heiko Drewes, Mohamed Khamis, and Florian Alt. 2019. DialPlates: enabling pursuits-based user interfaces with large target numbers. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia (MUM '19)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3365610.3365626

[13] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. 2017. Understanding Grip Shifts: How Form Factors Impact Hand Movements on Mobile Phones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4680–4691. https://doi.org/10.1145/3025453.3025835 event-place: Denver, Colorado, USA.

[14] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. 2018. Designing for Multiple Hand Grips and Body Postures within the UX of a moving Smartphone. In *Proceedings of the 2018 Designing Interactive Systems Conference (DIS '18)*. Association for Computing Machinery, New York, NY, USA, 611–621. https://doi.org/10.1145/3196709.3196711

[15] Rachel Eardley, Anne Roudaut, Steve Gill, and Stephen J. Thompson. 2018. Investigating How Smartphone Movement is Affected by Body Posture. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 202:1–202:8. https://doi.org/10.1145/3173574.3173776 event-place: Montreal QC, Canada.

[16] Augusto Esteves, Filipe Quintal, Fábio Caires, Vítor Baptista, and Pedro Mendes. 2019. Wattom: Ambient Eco-feedback with Mid-air Input. In *2019 5th Experiment International Conference (exp.at'19)*. 12–15. https://doi.org/10.1109/EXPAT.2019.8876565

[17] Augusto Esteves, Yonghwan Shin, and Ian Oakley. 2020. Comparing selection mechanisms for gaze input techniques in head-mounted displays. *International Journal of Human-Computer Studies* 139 (July 2020), 102414. https://doi.org/10.1016/j.ijhcs.2020.102414

[18] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM Press, 457–466. https://doi.org/10.1145/2807442.2807499

[19] Augusto Esteves, David Verweij, Liza Suraiya, Rasel Islam, Youryang Lee, and Ian Oakley. 2017. SmoothMoves: Smooth Pursuits Head Movements for Augmented Reality. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (UIST '17)*. ACM, New York, NY, USA, 167–178. https://doi.org/10.1145/3126594.3126616

[20] Jean-Daniel Fekete, Niklas Elmqvist, and Yves Guiard. 2009. Motion-pointing: Target Selection Using Elliptical Motions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 289–298. https://doi.org/10.1145/1518701.1518748

[21] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (1954), 381–391. https://doi.org/10.1037/h0055392

[22] Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2016. Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2319–2331. https://doi.org/10.1145/2858036.2858308 event-place: San Jose, California, USA.

[23] Euan Freeman, Gareth Griffiths, and Stephen A. Brewster. 2017. Rhythmic Micro-gestures: Discreet Interaction On-the-go. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI 2017)*. ACM, New York, NY, USA, 115–119. https://doi.org/10.1145/3136755.3136815

[24] Taejin Ha, Steven Feiner, and Woontack Woo. 2014. WeARHand: Head-worn, RGB-D camera-based, bare-hand user interface with visually enhanced depth perception. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 219–228. https://doi.org/10.1109/ISMAR.2014.6948431

[25] John Paulin Hansen, Vijay Rajanna, I. Scott MacKenzie, and Per Bækgaard. 2018. A Fitts' Law Study of Click and Dwell Interaction by Gaze, Head and Mouse with a Head-mounted Display. In *Proceedings of the Workshop on Communication by Gaze Interaction (COGAIN '18)*. ACM, New York, NY, USA, 7:1–7:5. https://doi.org/10.1145/3206343.3206344

[26] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage Publications Sage CA: Los Angeles, CA, 904–908. Issue: 9.

[27] Khalad Hasan, Junhyeok Kim, David Ahlström, and Pourang Irani. 2016. Thumbs-Up: 3D Spatial Thumb-Reachable Space for One-Handed Thumb Interaction on Smartphones. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. Association for Computing Machinery, New York, NY, USA, 103–106. https://doi.org/10.1145/2983310.2985755

[28] Yi-Ta Hsieh, Antti Jylhä, Valeria Orso, Luciano Gamberini, and Giulio Jacucci. 2016. 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 '16)*. ACM, New York, NY, USA, 4203–4215. https://doi.org/10.1145/2858036.2858436

[29] Robert J. K. Jacob. 1990. What you look at is what you get: eye movement-based interaction techniques. ACM Press, 11–18. https://doi.org/10.1145/97243.97246

[30] Amy K. Karlson and Benjamin B. Bederson. 2007. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. In *Human-Computer Interaction – INTERACT 2007 (Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, 324–338. https://doi.org/10.1007/978-3-540-74796-3_30

[31] Amy K. Karlson and Benjamin B. Bederson. 2008. One-handed Touchscreen Input for Legacy Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1399–1408. https://doi.org/10.1145/1357054.1357274

[32] Amy K Karlson, Benjamin B Bederson, and Jose L Contreras-Vidal. 2006. Studies in one-handed mobile design: Habit, desire and agility. In *Proc. 4th ERCIM Workshop User Interfaces All (UI4ALL)*. Citeseer, 1–10.

[33] Mohamed Khamis, Carl Oechsner, Florian Alt, and Andreas Bulling. 2018. VRpursuits: Interaction in Virtual Reality Using Smooth Pursuit Eye Movements. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces (AVI '18)*. ACM, New York, NY, USA, 18:1–18:8. https://doi.org/10.1145/3206505.3206522

[34] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. 2012. Interaction Techniques for Unreachable Objects on the Touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference (OzCHI '12)*. ACM, New York, NY, USA, 295–298. https://doi.org/10.1145/2414536.2414585

[35] Jianwei Lai and Dongsong Zhang. 2015. ExtendedThumb: A Target Acquisition Approach for One-Handed Interaction With Touch-Screen Mobile Phones. *IEEE Transactions on Human-Machine Systems* 45, 3 (June 2015), 362–370. https://doi.org/10.1109/THMS.2014.2377205 Conference Name: IEEE Transactions on Human-Machine Systems.

[36] Huy Viet Le, Patrick Bader, Thomas Kosch, and Niels Henze. 2016. Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, 27:1–27:10. https://doi.org/10.1145/2971485.2971562

[37] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2018. *Fingers' Range and Comfortable Area for One-Handed Smartphone Interaction Beyond the Touchscreen.* Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173605

[38] Wing Ho Andy Li, Hongbo Fu, and Kening Zhu. 2016. BezelCursor: Bezel-Initiated Cursor for One-Handed Target Acquisition on Mobile Touch Screens. https://doi.org/10.4018/IJMHCI.2016010101

[39] Markus Löchtefeld, Christoph Hirtz, and Sven Gehring. 2013. Evaluation of hybrid front- and back-of-device interaction on mobile devices. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM '13)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/2541831.2541865

[40] Markus Löchtefeld, Phillip Schardt, Antonio Krüger, and Sebastian Boring. 2015. Detecting users handedness for ergonomic adaptation of mobile user interfaces. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia (MUM '15)*. Association for Computing Machinery, New York, NY, USA, 245–249. https://doi.org/10.1145/2836041.2836066

[41] Sylvain Malacria, Eric Lecolinet, and Yves Guiard. 2010. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the cyclostar approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 2615–2624. https://doi.org/10.1145/1753326.1753724

[42] Evan Miller, Silvano Galliani, and Seth Paniagua. 2020. Use your voice to operate HoloLens. https://docs.microsoft.com/en-us/hololens/hololens-cortana

[43] Donald A. Norman and Jakob Nielsen. 2010. Gestural Interfaces: A Step Backward in Usability. *interactions* 17, 5 (Sept. 2010), 46–49. https://doi.org/10.1145/1836216.1836228

[44] Hyung Min Park, Seok Han Lee, and Jong Soo Choi. 2008. Wearable augmented reality system using gaze interaction. IEEE, 175–176. https://doi.org/10.1109/ISMAR.2008.4637353

[45] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. 2015. Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 373–383. https://doi.org/10.1145/2807442.2807460

[46] Ken Pfeuffer, Melodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 261–270. https://doi.org/10.1145/2501988.2501998

[47] Thammathip Piumsomboon, Gun Lee, Robert W. Lindeman, and Mark Billinghurst. 2017. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. 36–39. https://doi.org/10.1109/3DUI.2017.7893315

[48] R. L. Potter, L. J. Weldon, and B. Shneiderman. 1988. Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88)*. ACM, New York, NY, USA, 27–32. https://doi.org/10.1145/57167.57171

[49] Jeffrey M Quinn, Justin Eddings, and Tony Stewart. 2013. Portrait Versus Landscape: Task and Hardware Determine User's Mobile Phone Orientation Behavior. In *Education Symposium (IDSA '13)*. Industrial Designers of America, 1–5. https://www.idsa.org/content/portrait-versus-landscape

[50] Filipe Quintal, Augusto Esteves, Fábio Caires, Vítor Baptista, and Pedro Mendes. 2019. Wattom: A Consumption and Grid Aware Smart Plug with Mid-air Controls. ACM, 307–313. https://doi.org/10.1145/3294109.3295642

[51] Gabriel Reyes, Jason Wu, Nikita Juneja, Maxim Goldshtein, W. Keith Edwards, Gregory D. Abowd, and Thad Starner. 2018. SynchroWatch: One-Handed Synchronous Smartwatch Gestures Using Correlation and Magnetic Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4 (Jan. 2018), 158:1–158:26. https://doi.org/10.1145/3161162

[52] Julie Rico and Stephen Brewster. 2010. Gesture and Voice Prototyping for Early Evaluations of Social Acceptability in Multimodal Interfaces. In *International Conference on Multimodal Interfaces and the Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI '10)*. ACM, New York, NY, USA, 16:1–16:9. https://doi.org/10.1145/1891903.1891925

[53] Radiah Rivu, Yasmeen Abdrabou, Ken Pfeuffer, Mariam Hassib, and Florian Alt. 2020. Gaze'N'Touch: Enhancing Text Selection on Mobile Devices Using Gaze. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382802

[54] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. 2008. TapTap and MagStick: Improving One-handed Target Acquisition on Small Touch-screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '08)*. ACM, New York, NY, USA, 146–153. https://doi.org/10.1145/1385569.1385594

[55] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. 2020. Outline Pursuits: Gaze-assisted Selection of Occluded Objects in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376438

[56] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Human-Computer Interaction - INTERACT 2005*. Springer, Berlin, Heidelberg, 267–280. https://doi.org/10.1007/11555261_24

[57] G. M. Smith and m. c. schraefel. 2004. The Radial Scroll Tool: Scrolling Support for Stylus- or Touch-Based Document Navigation. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (Santa Fe, NM, USA) *(UIST '04)*. Association for Computing Machinery, New York, NY, USA, 53–56. https://doi.org/10.1145/1029632.1029641

[58] Daniel Spelmezan, Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. 2013. Side pressure for bidirectional navigation on small devices. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services (MobileHCI '13)*. Association for Computing Machinery, New York, NY, USA, 11–20. https://doi.org/10.1145/2493190.2493199

[59] Qingkun Su, Oscar Kin-Chung Au, Pengfei Xu, Hongbo Fu, and Chiew-Lan Tai. 2016. 2D-Dragger: unified touch-based target acquisition with constant effective width. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 170–179. https://doi.org/10.1145/2935334.2935339

[60] Hsin-Ruey Tsai, Da-Yuan Huang, Chen-Hsin Hsieh, Lee-Ting Huang, and Yi-Ping Hung. 2016. MovingScreen: selecting hard-to-reach targets with automatic comfort zone calibration on mobile devices. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 651–658. https://doi.org/10.1145/2957265.2961835

[61] Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. 2017. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Trans. Comput.-Hum. Interact.* 24, 3 (April 2017), 22:1–22:35. https://doi.org/10.1145/3064937

[62] Eduardo Velloso, Flavio Luiz Coutinho, Andrew Kurauchi, and Carlos H Morimoto. 2018. Circular Orbits Detection for Gaze Interaction Using 2D Correlation and Profile Matching Algorithms. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 25, 9 pages. https://doi.org/10.1145/3204493.3204524

[63] Eduardo Velloso, Markus Wirth, Christian Weichel, Augusto Esteves, and Hans Gellersen. 2016. AmbiGaze: Direct Control of Ambient Devices by Gaze. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*. ACM Press, 812–817. https://doi.org/10.1145/2901790.2901867

[64] David Verweij, Augusto Esteves, Saskia Bakker, and Vassilis-Javed Khan. 2019. Designing Motion Matching for Real-World Applications: Lessons from Realistic Deployments. ACM, 645–656. https://doi.org/10.1145/3294109.3295628

[65] David Verweij, Augusto Esteves, Vassilis-Javed Khan, and Saskia Bakker. 2017. Smart Home Control Using Motion Matching and Smart Watches. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17)*. ACM, New York, NY, USA, 466–468. https://doi.org/10.1145/3132272.3132283 event-place: Brighton, United Kingdom.

[66] David Verweij, Augusto Esteves, Vassilis-Javed Khan, and Saskia Bakker. 2017. WaveTrace: Motion Matching Input Using Wrist-Worn Motion Sensors. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 2180–2186. https://doi.org/10.1145/3027063.3053161

[67] David Verweij, Vassilis-Javed Khan, Augusto Esteves, and Saskia Bakker. 2017. Multi-User Motion Matching Interaction for Interactive Television Using Smartwatches. In *Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video (TVX '17 Adjunct)*. ACM, New York, NY, USA, 67–68. https://doi.org/10.1145/3084289.3089906

[68] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 439–448. https://doi.org/10.1145/2493432.2493477

[69] Simon Voelker, Sebastian Hueber, Christian Corsten, and Christian Remy. 2020. HeadReach: Using Head Tracking to Increase Reachability on Mobile Touch Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376868

[70] John Williamson and Roderick Murray-Smith. 2004. Pointing Without a Pointer. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04)*. ACM, New York, NY, USA, 1407–1410. https://doi.org/10.1145/985921.986076

[71] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[72] Jason Wu, Cooper Colglazier, Adhithya Ravishankar, Yuyan Duan, Yuanbo Wang, Thomas Ploetz, and Thad Starner. 2018. Seesaw: Rapid One-Handed Synchronous Gesture Interface for Smartwatches. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers* (Singapore, Singapore) *(ISWC '18)*. Association for Computing Machinery, New York, NY, USA, 17–20. https://doi.org/10.1145/3267242.3267251

[73] Xing-Dong Yang, Edward Mak, Pourang Irani, and Walter F. Bischof. 2009. Dual-Surface Input: Augmenting One-handed Interaction with Coordinated Front and Behind-the-screen Input. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, 5:1–5:10. https://doi.org/10.1145/1613858.1613865

[74] Neng-Hao Yu, Da-Yuan Huang, Jia-Jyun Hsu, and Yi-Ping Hung. 2013. Rapid Selection of Hard-to-access Targets by Thumb on Mobile Touch-screens. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 400–403. https://doi.org/10.1145/2493190.2493202

[75] Tengxiang Zhang, Xin Yi, Ruolin Wang, Jiayuan Gao, Yuntao Wang, Chun Yu, Simin Li, and Yuanchun Shi. 2019. Facilitating Temporal Synchronous Target Selection through User Behavior Modeling. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article 159 (dec 2019), 24 pages. https://doi.org/10.1145/3369839