Summon and Select: Rapid Interaction with Interface Controls in Mid-air

Aakar Gupta[1], **Thomas Pietrzak**[2], **Cleon Yau**[3], **Nicolas Roussel**[4], **Ravin Balakrishnan**[5]
[1]University of Toronto, Canada, [2]Univ. Lille, France, [3]Inria, France
aakar@dgp.toronto.edu, thomas.pietrzak@univ-lille1.fr, cleon.yau@mail.utoronto.ca,
nicolas.roussel@inria.fr, ravin@dgp.toronto.edu

ABSTRACT

Current freehand interactions with large displays rely on point & select as the dominant paradigm. However, constant hand movement in air for pointer navigation leads to hand fatigue quickly. We introduce summon & select, a new model for freehand interaction where, instead of navigating to the control, the user summons it into focus and then manipulates it. Summon & select solves the problems of constant pointer navigation, need for precise selection, and out-of-bounds gestures that plague point & select. We describe the design and conduct two studies to evaluate the design and compare it against point & select in a multi-button selection study. The results show that summon & select is significantly faster and has less physical and mental demand than point & select.

Author Keywords
Mid-air, Freehand, Gestures, Haptics

ACM Classification Keywords
H.5.2 User Interfaces

INTRODUCTION

Once the user decides the target control, why should there be a need to navigate to it? Contemporary freehand interactions in large displays follow the traditional point & select desktop model to manipulate controls such as buttons and sliders. Finger detection wares like Kinect and Leap Motion propagate the use of hand as a pointer proxy in their developer demos and commercial applications. Further, research in freehand interaction also centers on optimizing the pointing paradigm [22, 34, 28, 6]. However, hand as an input device in air is vastly different from a mouse. While the mouse has limited degrees of freedom (x-y movement, left-right click, scroll wheel), the hand enables a much wider range. Each of the five fingers are capable of x-y-z movement (with individuation constraints), the hand itself can move in the x-y plane, and the arm can move freely in x-y-z. The mouse is limited in that it cannot signal a variety of different input intentions in a single action, thus necessitating navigation to the targets. The hand, however, can signal a variety of intentions without navigating. Why, then, do freehand interactions in air still rely on a paradigm that catered to the limitations of traditional input devices? One possible solution is given by semaphoric gestures [19] where a unique symbolic gesture activates a command. However, these can only be performed for fixed commands and need to be learned extensively.

We introduce summon & select, a fast, low-fatigue freehand interaction model to interact with interface controls where the user summons the intended control on the large display to get it into focus and then manipulates it. In essence, it is a combination of semaphoric and manipulative gesturing, such that the user neither needs to navigate to the control, nor needs to learn an extensive gesture vocabulary. The user simply summons a control using a semaphoric gesture and then controls it using manipulative gestures. In the upcoming sections, we describe the design of summon & select interaction, its features and constraints, and report on a study that investigates its design. We then report on a second study that compares it to pointing and shows that summon and select outperforms point and select for a multi-button interface on both performance and preference. We also conduct a preliminary investigatin of how haptic feedback aids summon & select.

RELATED WORK

Multiple works have focused on mid-interactions using touch, pens, and other controllers [18, 8, 33, 24]. Our focus in this paper is on freehand interactions, whose existing literature can be divided into manipulative and semaphoric gestures.

Mid-air Manipulative & Semaphoric Gestures

The manipulative gestural work has mostly focused on point & select under two categories – a) absolute pointing: pointing at a target using raycasting and (b) relative pointing in a virtual 2D plane. Vogel *et al.* [34] found raycasting to be slower for smaller targets, but also found relative techniques to be highly time-consuming due to inconvenient clutching. Raycasting has been altered to better its performance – marker cone [28] points a cone in the ray-direction. Multiple ways for target selection after pointing have been proposed including thumb–index join [34], breach and trigger gesture [5], wrist tilt and pinch [25], double crossing the pointer [23], and using speed and distance of pointer movement to select menu items. Song *et al.* [30] propose an interaction technique to manipulate virtual objects using a handle bar metaphor. Gustafson et al's imaginary interfaces [16] investigate freehand interactions without any screen or visual feedback by

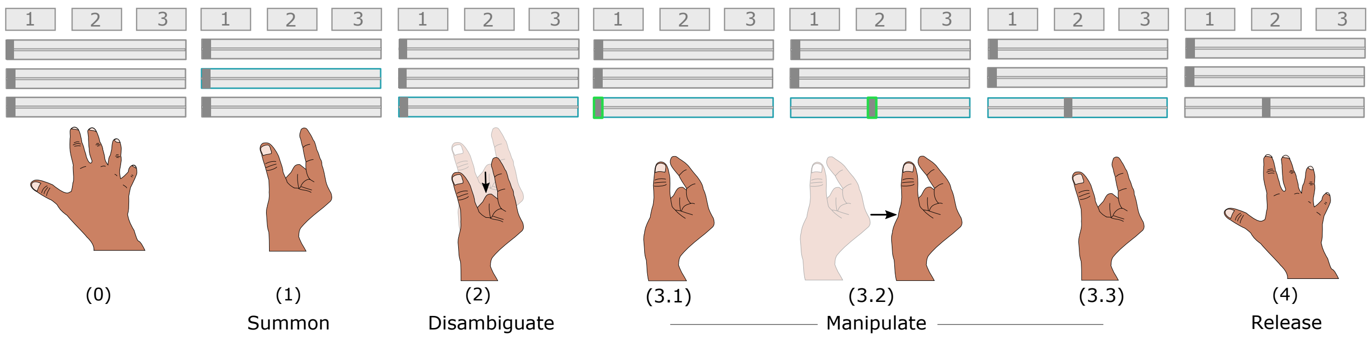

Figure 1: Steps of summon & select for the bottom slider. (0) Idle (1) Summoning gesture for slider (2) Disambiguating by zoning to the desired slider (blue focus moves to the bottom slider) (3.1-3.3) Manipulation: (3.1) *Enter Drag* gesture to enter dragging mode (green box around the bar) (3.2) Dragging the slider bar (3.3) *Exit Drag* gesture to exit dragging mode (4) Release gesture to release the control

studying how accurately a user can point in an imaginary plane fixed at the intersection of an L-shaped hand gesture. Stein et al's imaginary devices [31] is similar to summon & select in that making the pose for a physical device such as a keyboard allows the user to type on the imaginary device in air. However, summon & select is an interaction model to interact with controls in software interfaces as opposed to interacting with imaginary physical devices.

The fatigue problem in manipulative gestures is tackled in different ways. Freeman et al. [12] solve limited device tracking range by using light and tactile cues to guide where and how to gesture. Other works [27, 21] solve the problem of gesturing outside of the hand's comfort zone by enabling user creation of virtual planes. Gunslinger [22] uses bimanual relaxed arms-down gestures with one hand doing the pointing and the other performing semaphoric gestures.

Investigations on freehand semaphoric gestures [7], including proposing a library of gestures for common actions [17], eliciting gestures from users [20, 29], and studying them in comparison to navigational gestures for menu selection [3], their discoverability [35], and their use in various tasks [1, 2].

Alternative to Point & select

Balakrishnan [4] analyzes methods to beat Fitts' law for non mid-air contexts including jumps to the next target which is similar to our Zoning method. Within mid-air work, finger-count menus [3] allow menu item selection using a correspondance between the menu item ordering and the number of stretched out fingers. This method could possibly be used for buttons in interfaces as well. However, the number of targets are limited by the fingers unless higher order combinations of fingers are used. Further, the technique won't apply across other common controls. PathSync [10] and TraceMatch [11] enable object selection by mimicking the motion of the object or the motion-pattern displayed around the object.

All techniques stated above either offer improvements to pointing or enable target selection (mostly buttons) using semaphoric gestures. But none of these offer an end-to-end alternative to point & select starting with selection of a specific control amongst different types of controls, followed by a seamless transition to manipulation of the selected control. Summon & select combines semaphoric and manipulative gestures to offer a viable alternative to point & select.

SUMMON & SELECT

Summon & select takes place in four essential steps: 1) *Summon*: The user performs a specific *summoning gesture* to summon a type of control, like Button or Slider. This is denoted by a focus box on the control. 2) *Disambiguate*: If there are more than one controls of the summoned type, the user moves the focus box to the desired control by either *tabbing* through the controls or *zoning* coarsely into the region of the intended control. 3) *Manipulate*: The user can now manipulate the control by clicking or dragging it. 4) *Release*: The user releases the control by performing a release gesture.

Figure 1 illustrates how summon & select works for a slider control type. The interface consists of three buttons and three slider controls. The user intends to manipulate the bottom-most slider. *1) Summon:* The user makes the thumb+index finger summoning gesture (as if to hold the slider bar) to summon the slider control (Figure 1-(1)). This brings the middle slider into focus. At this stage, the region around the user's hand is divided into three large virtual zones vertically, each corresponding to one of the sliders. *2) Disambiguate:* The user moves the hand down to reach the volume slider's zone which brings it into focus (Figure 1-(2)). *3) Manipulate:* To enter drag state, the user performs the enter drag gesture which is to pinch the thumb and finger closer (as if to tighten the grip on the bar) (Figure 1-(3.1)). The user then moves the hand horizontally to reposition the bar (Figure 1-(3.2)). To exit drag state, the user performs the exit drag gesture which is to move the thumb and finger apart (Figure 1-(3.3)). The user can perform 3.1-3.3 repetitively to adjust the slider bar until she is satisfied. *4) Release:* The user then releases the slider by performing an open palm gesture (Figure 1-(4)).

The above example illustrates a dragging manipulation for the slider control. A button click will be simpler. The user makes an index finger-up summoning gesture (as if raised to press a button), zones into the desired button's region, performs an air tap to click it, and then releases it. Clicking a button can also open a new screen in some applications, whereupon the control will auto-release on click. Multiple controls of multiple control types can exist in a single interface. For instance, a video player application can consist of play, reverse, forward buttons and playback, volume sliders. The user can use the appropriate summoning gesture to summon a control type.

DESIGN ELEMENTS & MIDAS TOUCH

One of the biggest challenges for freehand gestures is Midas touch which is the detection of an unintended gesture because of a combination of imprecise detection and inefficient gesture language design. We design summon & select such that the possibility of accidental detection of unintended gestures is minimized. In summon & select, Midas touch can occur at each of the four steps: accidental summoning, accidental control switching, accidental manipulation, & accidental release. We now look at each of the four steps in depth and detail how we overcome the design challenges including Midas Touch.

1. Summon

For applications consisting of a multitude of control types, the user will have to learn and remember a large number of summoning gestures. To solve this, we propose that gestures be designed such that they evoke the physical manipulation of the control. For instance, the index finger-up button summoning gesture is similar to how a user would approach a physical button. Similarly, the thumb-index slider gesture is how a user would approach dragging a physical slider. Figure 2 shows summoning gestures for five different control types.

Midas Touch: Accidental Summoning

Accidental summoning can occur when a) the user's hand pose summons something when they did not intend to summon anything, b) the summon gesture summons an unintended control type, or c) when the transition to the following disambiguate or manipulation gestures are mis-recognized as part of the summon gesture or a new summon gesture. To prevent (a), we only summon a control when the summoning gesture has been held in the same pose for at least a 500ms period. To prevent (c), once a control type is summoned, the system does not register any new summoning gestures until the explicit release gesture releases the control. Solving (b) depends on consistent, non-interfering detection of easy-to-perform summoning gestures enabled on an interface.

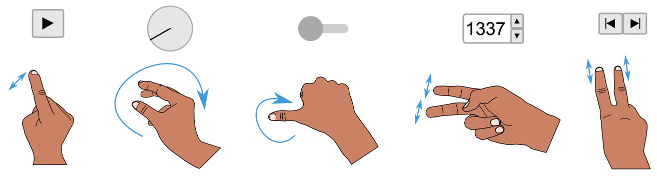

Figure 2: Example summoning gestures and manipulation gestures (in blue arrows) for different control types: Button (airtaps), dial knob (pinched rotation), switch (lateral thumb-tap), spinbox (lateral index, middle airtaps), paired buttons (vertical index, middle airtaps)

2. Disambiguate

While the interaction consists of four steps, the design focuses on making it as fluent as possible. This involves ensuring that the summoning gesture segues into manipulation seamlessly so as to complete the physical metaphor. The slider, for instance, starts with the thumb-index grip gesture that is then tightened to transition to manipulation. Similarly, Button's index finger-up gesture leads to air-tap. Thus the *Disambiguate* step should not alter the summoning gesture pose. We propose two control disambiguation techniques that satisfy this criterion: *Zoning* and *Tabbing*.

Zoning: As illustrated in Figure 1, in Zoning, after the user summons a control type, the hand tracking zone within which the user can comfortably move her hand is divided according to the number and position of controls of the summoned type. For example, in Figure 1, summoning the slider type divides the zone into 3 parts divided horizontally, each part associated with its slider. Similarly for buttons, the zone is divided into 3 vertical zones. When the user performs a summoning gesture, the control in the zone with the user's hand is summoned. The user then moves her hand into the desired control's zone which summons it. The user can now manipulate the control. Once the user starts manipulation, zoning is turned off, so the user is not required to stay within a zone while manipulating.

Accidental Zoning could occur if the user's hand accidentally stumbles into another zone while doing the manipulation gesture such as gripping the slider bar. This is the classic midas touch problem which frequently occurs in point & select interactions, especially when trying to select smaller targets. Here, since the comfortable interaction area around the user is divided into zones, each zone will be large enough to avoid accidental zoning unless there's a large number of controls of the same type. Study I and Study II shed light on this issue for a television sliders interface and a multi-button interface.

Tabbing: Even though Zoning does not ask the user to change their hand pose, since it involves using the same hand, there could be a possibility that it interferes with the enter drag gesture for certain control types. Consequently, we propose an alternate Disambiguate gesture which involves *Tabbing* through the controls by repeatedly opening and closing the second hand. Each hand closing pose results in a tab through to the next control of the summoned type in the logical ordering on the interface. This means that (a) the first hand is not required to move, and (b) the second hand only needs to change its pose without any movement, thus minimizing the first hand's wobbling even further. While Tabbing insulates the interaction from midas touch when disambiguating, it is a more tedious gesture that requires the second hand and tabbing through the controls one by one. Study I compares Tabbing and Zoning for quantitative and qualitative performance.

3. Manipulate

Control manipulation can be static such as a button click or dynamic such as dragging the slider bar. Just like point & select for the mouse has a Dragging state distinct from its Tracking state [9], summon & select for the hand has a Dragging state distinct from its Summoned state which the user needs to enter and exit explicitly. For button clicks, the enter drag-manipulate-exit drag transition happens all in one airtap. For controls that involve dragging such as sliders, an explicit enter-exit gesture pair ensures that *accidental manipulation* of a control is a remote possibility during Disambiguation. The gesture pair should be designed such that (a) the state transition is seamless as already discussed, and

(b) the gesture pair itself does not cause *accidental dragging* in the middle of performing it. As mentioned, the physical metaphor of tightening and loosening the grip on the object being dragged satisfies both the requirements. For instance, the knob control in Figure 3 involves a three-finger summoning gesture followed by tightening the grip to rotate the knob. Static operations such as button clicks are less prone to (b). The blue arrows in Figure 2 indicate how different control types' summoning gestures lead to manipulation. Once control manipulation starts, disambiguation is turned off. To select a new control, the user needs to release and summon again. This allows the user to freely move their hands and position them when not in the dragging state.

Clutching

The explicit drag state enables clutching. For example, for video playback sliders, precise dragging might be an issue if the hand tracking zone is mapped to the entire duration of a very long video. We can map it to a smaller period and the users can clutch by entering drag, dragging, and exiting drag repeatedly. Study I investigates how accurate users can be while positioning the bar to a specific value.

4. Release

The gesture that releases the summoned control should be such that (a) it does not lead to accidental manipulation, and (b) is not confused with summoning/disambiguate/manipulation gestures to prevent *accidental release*. We designate an open palm gesture with the summoning hand as the release gesture which satisfies both requirements to a very high degree. The release gesture doubles as a quick Undo when user summons the wrong control. Further, the open palm pose also acts as the perfect reset of the hand before beginning the next summoning gesture. Figure 3 shows the summon & select state machine.

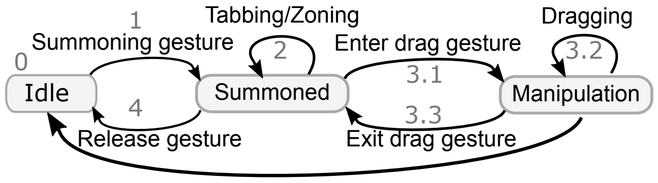

Figure 3: Summon & select state machine. Numbers match to Fig 1.

SUMMON & SELECT: ADVANTAGES AND LIMITATIONS

Advantages & Applicability

Point & select in mid-air is fatiguing due to multiple issues which summon & select overcomes by virtue of its design: 1) *Navigational motion:* Summon & select gets rid of the constant hand movement for navigating to desired controls. 2) *Precise pointing & selection:* To click a button, point & select requires the user to precisely position the hand over it and dwell/air-tap. Dwell requires the user to wait and keep the hand stationary, while air-tap requires the user to not displace from the button while performing the air-tap due to the hand wobble. This is physically & mentally fatiguing, especially for smaller targets. In summon & select, once a target has been summoned, the user only needs to remain within the zone of a button (for Zoning) which would typically be much larger than the size of the button. For Tabbing, the user is free to move the hand anywhere in the tracking zone and clicking without losing the summoned target. 3) *Out-of-bounds controls:* The user's hand can comfortably move only within a certain area relative to the body. If the user is positioned close to the center of the screen and the target is closer to the screen edges, the user needs to either stretch out the hand far out of the comfort zone or move the body itself. The problem is especially exacerbated with ultra-sized displays. 4) *Out-of-tracking range:* A related issue is that hand tracking devices have limited tracking range which might not be enough for precise control of a large displays if mapped directly for point & select. Relative pointing is a possible solution; however, there are no easy ways to clutch when using Point & select in air. Summon & select can be comfortably performed within the hand comfort zone and tracking range. Study II investigates whether these factors actually yield a better performance and user experience than point & select.

Summon & select can be useful for large screen displays for applications that involve interacting with multiple controls without restricting them to low-precision big-button interfaces; for instance, video players, menu grids (such as game menus), and television interfaces (as in Figure 4(left)). They can also be useful for other display scenarios such as car dashboards or virtual reality. Since summon & select with Tabbing requires minimal hand movement and is not affected by casual hand motion (unless it is in dragging state), it can be useful in scenarios where the user is mobile (for instance, a user walking with an augmented reality headset).

Limitations

Current large screen interfaces are designed for point & select. Summon & select is limited in its capabilities to substitute point & select across all interfaces. First, control types like checkbox groups or date-pickers can not be easily adapted for summon & select. There is no obvious summoning gesture aside from the index finger-up gesture which is already reserved for the button. Further, their manipulation is entirely dependent on point & select. While such controls are used less often in large screen applications, it is a drawback. Second, summon & select will not work in applications like maps where the user can select any random point on the interface. Given these constraints, summon & select can only replace point & select if the application is not constrained by the above two limitations. An additional factor is the simultaneous use of basic semaphoric gestures such as swiping. These gestures work at the scope of the entire screen, such as a left swipe for the next page. These can easily work together with summon & select if they are performed with poses that do not interfere with an application's summoning gestures.

PROTOTYPE IMPLEMENTATION

We built a prototype system with a ($60cm \times 33cm$) screen and Leap Motion for hand tracking. The interaction zone is fixed at ($50cm \times 27.5cm$) to account for tracking range, hand-comfort bounds, and appropriate mapping resolution. We investigate two control types: Button & Slider. For visual feedback, a blue focus frame forms around the summoned slider.

As the user zones/tabs, the focus moves accordingly. Upon entering the drag state, the slider's shading changes. Similar visual feedback is implemented for the button.

The hand gesture recognition was built on Leap's hand model data such that the gestures were recognized correctly and there was minimal conflict between various gestures and transitions. For example, the slider summon gesture was recognized when the thumb and index finger were stretched out and the other three fingers were curled in. The thresholds for streched out and curled in were defined such that they struck a balance between the exact gesture and a more relaxed gesture. After multiple iterations with the algorithm with, we conducted a pilot study with three users to investigate the gesture detection accuracy within lab conditions. After a 10 minute training and practice session where the users learned the way to perform the gestures so as to extract maximum accuracy, each user performed 32 trials: 8 trials for summoning the button, zoning into one of nine buttons laid out on the interface, clicking it, and releasing it; 8 trials for summoning the slider, zoning into one of five sliders on the interface, tightening the grip, dragging the bar to any value they wanted, loosening the grip and releasing; and 16 similar ones with tabbing. We only looked at the accuracy of the detection of each gesture. 92.7% trials were performed without any detection errors which was deemed good enough for the studies. Note that the simple gesture detection algorithm performed well for the requirements of the study within the controlled lab environment. A real world implementation would need more robust algorithms. During the studies, we discarded the trials with erroneous detection and redid them. No major Midas touch issues were observed.

STUDY I: SLIDER DISAMBIGUATION & DRAGGING

We conducted a study to investigate the following factors: 1) Tabbing vs. Zoning performance: How fast can the user tab or zone to the right slider amongst five vertically adjacent sliders? Further, how does the transition from the summoning gesture to manipulation get affected by Tabbing vs. Zoning? 2) Dragging performance: With clutching enabled, how quickly and accurately can the user drag the bar to the correct value? This might differ for Tabbing vs Zoning because the user's hand is not always centered in zoning.

Study Design

The study follows a within-subjects design with two independent variables, disambiguation technique (Tabbing, Zoning) and slider number (Figure 4(left), sliders 1-5). The sliders, modeled on television settings, go from 0-100 and have a relative mapping where a 1cm hand displacement corresponds to 1 tick. 1cm is just enough for the user to accurately position the slider to an exact value in combination with clutching. In each trial, the participant is instructed to summon a particular slider and set it to a certain target value. The trial ends when the participant releases the slider. The next trial begins 3s after this release. The slider bar is repositioned at 50 at the start of each trial. The default focus at the start of each trial in the tabbing condition is at the bottom slider *(slider 1)*.

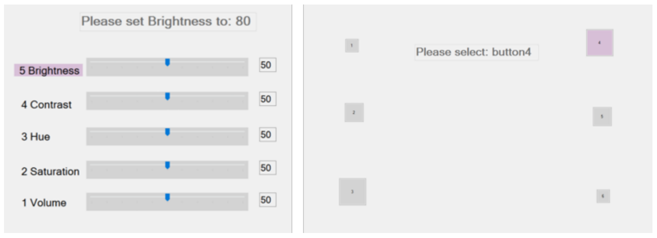

Figure 4: (left) Study I interface: Sliders are numbered 1-5 from the bottom. (right) Study II interface: Buttons are numbered 1-3 from top to bottom on left and 4-6 on right. The original screens had more free space around these snapshots which have been cropped here for space.

12 right-handed participants (mean age = 23, range: 21-26, 8 male), none of whom had experience with freehand interaction took part. The disambiguation technique was counterbalanced among participants. Participants were introduced to the gestures initially and allowed to play and practice with them. Before each technique, participants performed four practice trials. Each participant performed two trials per slider per technique with a different target value for each of the two trials. The two values remained same across sliders. The ordering of sliders and the target values was randomized. Participants were instructed to be as fast and accurate as possible. For Zoning, participants are asked to rest their elbow on the table after every trial, which brings them into the *slider 1* zone. However, participants with longer forearms would sometimes overshoot into an upper zone. The study lasted 30min. Participants were given a 2min break between the techniques. In total, we had 12 PARTICIPANTS × 2 TECHNIQUES × 5 SLIDERS × 2 REPETITIONS = 240 total trials.

Results

Midas Touch

Of 120 Zoning trials, 8 trials resulted in zoning to an incorrect SLIDER (6.67% error). This was due to accidental zoning when the participant did the *enter drag* gesture with a strong jerk which switched the hand's zone. For Tabbing, 3 of 120 trials resulted in tabbing to an incorrect SLIDER. Apart from this, no other Midas touch problems were observed.

Tabbing vs. Zoning Performance

A user can potentially reach the correct SLIDER multiple times while tabbing or zoning before starting to manipulate it. We measured the *reach time* for the intended slider starting at the time of instruction until the final instance the user reaches the correct SLIDER. So if the user passes over the desired control more than once, the time for the last such instance before they start manipulation is used. If the user selects the incorrect SLIDER, the trial is discarded. A two-way repeated measures anova with Greenhouse-Geisser correction showed a significant interaction effect of TECHNIQUE and SLIDER on the reach time ($F(1.891, 44) = 4.927$, $p < .01$, $\eta_p^2 = .309$). Figure 5 shows the mean reach times: $0.1s$, $1.8s$, $2.4s$, $3.0s$ and $3.4s$ for tabbing to sliders 1-5 and $1.1s$, $2.9s$, $1.8s$, $2.9s$ and $2.2s$ for zoning to sliders 1-5.

For SLIDER 1, Tabbing's reach time is faster. This is confirmed by posthoc tests for SLIDER 1 ($F(1, 11) = 11.262$, $p < .01$, $\eta_p^2 = .506$). In fact, Tabbing's reach time is almost zero since it is the default tab. For Zoning, most participants' hand started in SLIDER 1's zone, but a few started in different zones. Reach times are comparable for sliders 2, 3, and 4. For SLIDER 5, Zoning is faster ($F(1,11) = 14.104$, $p < .01$, $\eta_p^2 = .562$), because tabbing 5 times takes longer and because it is easy to reach the top zone quickly with less hand adjustments. The results show that after a certain number of controls of same type, zoning begins to outperform tabbing. Since the data points per slider are limited (12x2), a larger study will shed more light on slider count differences. This also depends on the arrangement of the controls as well. We explore a 2D arrangement in the next study.

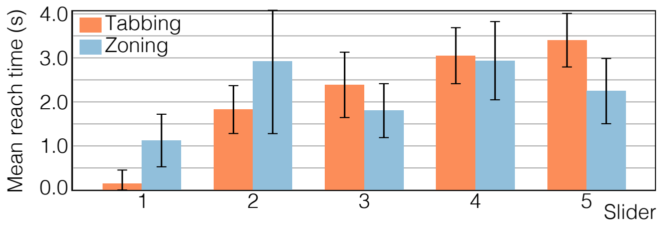

Figure 5: StudyI: Mean reach times for Tabbing and Zoning for the five sliders. Tabbing is faster for SLIDER 1, while zoning is faster for SLIDER 5. Error bars are 95% CI.

Seamless Transition

Transition time is measured starting from the last time the target slider control until the user performs the enter drag gesture. We found that Zoning had significantly faster transition (mean=0.87s, 95% CI [.77,.97]) than Tabbing (mean=1.46s, 95% CI [1.05,1.87]) ($F(1,11) = 13.917$, $p < .01$, $\eta_p^2 = .559$). This was expected since the user can zone and do the drag state gesture in one single flow with the same hand.

Dragging Performance

We measure the dragging time starting with the point at which the drag state is entered to when the control is released. No significant main or interaction effects were found. Dragging took a mean time of 6.7s (95% CI [5.1,7.8]) for Tabbing and 7.8s (95% CI [6.5,9.1]) for Zoning. Although zoning has a lower mean, the differences are not significant. However, one participant mentioned the dragging while zoning issue - *"Sliding with my hand in a high or low position was tougher."* Overall, 7 participants preferred Zoning, 5 preferred Tabbing.

Participants performed dragging highly precisely. The mean difference error across all 240 trials was 0.17. (Three outliers with a difference >25 were removed.) This shows that clutching was effective. Further, it shows that accidental manipulation due to ungrasping is not an issue. Participants performed an average of 3.1 clutches, which implies that they clutched every 8.3cm on average. Even though participants can easily move their hand within a range of 25cm, they chose to perform smaller movements with more clutching.

Summary

In summary, (a) Tabbing's summoning speed is similar to Zoning for vertically distributed controls. Tabbing is obviously faster for the default control. Zoning is faster for the control with the fifth tab. Tabbing vs. zoning is governed by the number of controls and their positioning. (b) Further, transition to drag with Zoning is significantly faster than with Tabbing. (c) Even though sliding in low or high positions was considered tougher, dragging time for Zoning and Tabbing are similar. (d) Participants can perform dragging with a very high accuracy and prefer clutching in air over moving a larger distance. (e) Dragging is sufficiently immune to accidental manipulation from enter/exit drag gestures.

STUDY II: SUMMON & SELECT VS POINT & SELECT

Study I showed that tabbing and zoning have similar performance for five vertically aligned slider controls. It also showed that the summon & select interaction works well without midas touch issues. In Study II, we see if summon & select can better point & select in terms of performance for a multi-button selection task. We investigate three specific metrics - 1) Reach Time: Time it takes to reach the desired button just before clicking. We hypothesize that the reach time will be be affected by the technique with pointing being the slowest and zoning being the fastest. 2) Selection Time: Total time it takes to select the desired button. We hypothesize that this will be affected both by technique and button size. 3) Qualitative aspects. In addition, we also hope to see how a different layout of controls which involves both horizontal & vertical movement affects Tabbing and Zoning.

Study Design

The study followed a within-subjects design with three independent variables - technique, button size, and button direction. The three techniques were: summon & select with tabbing vs. summon & select with zoning vs. point & select. The interface consists of six buttons, with three different sizes and in two directions (Figure 4 (right)). For Zoning, six zones are formed from two vertical and three horizontal divisions.

Th button sizes are $2.25cm^2$, $4.5cm^2$, and $9cm^2$. For pointing, the screen ($60cm \times 33cm$) is mapped to the tracking zone ($50cm \times 27.5cm$) centered at the screen center. The setup design was such that the user can reach the buttons comfortably which are at 20cm from the center on either side. As earlier, airtaps are used for button clicks. Earlier work [32] suggests that users overwhelmingly prefer airtaps for button clicks over 8 other alternatives (dwell, grab, etc.). For zoning, the same ($50cm \times 27.5cm$) area is mapped to six equal zones corresponding to each button. The default focus for Tabbing is at Button 1 which is small. Button 6, the other small button is last in the tabbing order.

12 right-handed participants (mean age = 24.1, range 20-28, 7 male), all different from prior studies took part. Two participants had earlier experience with Kinect. The technique is counterbalanced amongst participants using a Latin square design. Each trial starts with the user's hand in the center of the tracking zone, directly above Leap Motion with the pointer visible only in the Pointing condition. The user is instructed to select a particular button. The user summons and tabs/zones, or reaches over a button depending on the condition, then clicks it, and finally performs the release gesture to end the trial. The user is then instructed to bring the hand back to the center, after which the next trial starts. The selection time data from incorrect button selections is discarded.

In pointing, the user might airtap multiple times on screen areas without any button. These are recorded as missed clicks.

There are 4 trials for each of the 6 buttons per technique. Ordering of the 24 trials for each technique is randomized. Participants practiced the interactions for 10mins and did four practice trials before every technique. At the end, a Likert scale questionnaire was given and a brief interview was conducted. The study lasted 40 mins. In total, we had 12 PARTICIPANTS × 3 TECHNIQUES × 3 BUTTON SIZES × 2 DIRECTIONS (left/right) × 4 REPETITIONS = 864 button clicks.

Results

Reach Time

For zoning and tabbing, the reach time is measured same as in Study II, starting from the instruction until the control is in focus just before clicking it. For pointing, reach time is the first time the pointer reaches over the target. The pointer can leave and enter the button area repeatedly in an attempt to click it, but the reach time is when the user reaches it the first time. A three-way repeated measures ANOVA showed an interaction effect of TECHNIQUE and DIRECTION ($F(2,22) = 17.300$, $p < .001$, $\eta_p^2 = .611$). No effects of BUTTON SIZE were found. Figure 6 shows the reach time for the three TECHNIQUES for both DIRECTIONS: mean $2.4s/2.0s$ for pointing, $1.5s/2.6s$ for tabbing and $1.9s/1.9s$ for zoning (left/right button). Because of higher tabbing requirement, Tabbing on the right is slower than both (pairwise comparison with pointing: ($F(1,11) = 5.113$, $p < .05$, $\eta_p^2 = .317$), with zoning: ($F(1,11) = 18.792$, $p < .01$, $\eta_p^2 = .631$)). However, Tabbing outperforms Pointing in the left ($F(1,11) = 20.895$, $p < .01$, $\eta_p^2 = .655$) and is clearly benefited by the default focus and low tabbing requirement. Zoning performs similar to Tabbing in the left. Although their means are different, Pointing and Zoning did not differ significantly in their reach times in either DIRECTION. This is because the buttons were easily reachable from the center. A variable distance study can shed light on whether Zoning improves reach time.

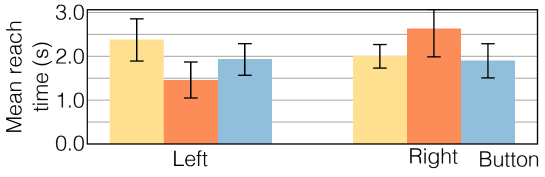

Figure 6: StudyII: Reach Time per TECHNIQUE for left & right buttons. Error bars are 95% CI

In summary, 1) Tabbing has better reach time than pointing for up to to three controls. 2) Pointing has better reach time than tabbing for >3 controls. 2) Zoning's reach time is highly consistent in both DIRECTIONS and is not outperformed in any of the cases including tabbing for <=3 controls.

Selection Time

The total selection time is logged starting from the instruction prompt. For summon conditions, selection time is measured until the completion of the release gesture. For pointing, it is measured until the button click since release is not a part of the point & select interaction and only serves to end the trial. A three-way repeated measures anova showed significant main effects of TECHNIQUE ($F(2,22) = 8.328$, $p < .01$, $\eta_p^2 = .431$) and BUTTON SIZE ($F(1.177, 12.947) = 5.548$, $p < .05$, $\eta_p^2 = .339$), and significant interaction effects of Technique × Button size ($F(1.912, 21.030) = 4.723$, $p < .01$, $\eta_p^2 = .300$), and Technique × Direction ($F(1.294, 14.233) = 17.018$, $p < .001$, $\eta_p^2 = .607$). (The last 3 are with Greenhouse-Geisser correction.) The DIRECTION effects can be attributed to the inclusion of reach time in selection time. The pairwise significant effects in reach time for DIRECTIONS are mirrored in selection time. We delve deeper into Technique and Button size effects.

Figure 7 shows the selection time for the three TECHNIQUES for each BUTTON SIZE (large, medium, small): mean $3.2s/3.5s/4.6s$ for pointing, $3.4s/3.5s/3.5s$ for tabbing and $2.6/2.7/2.8s$ for zoning. Posthoc tests with Bonferroni correction show that Zoning has a significantly lower selection time than both Pointing and Tabbing across all BUTTON SIZES ($p < 0.05$ for both). There were no significant differences between Tabbing and Pointing for any of the BUTTON SIZES. Expectedly, pointing speed for the small button was significantly slower than the other two.

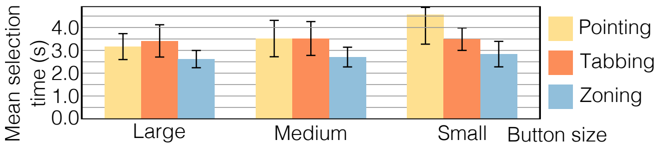

Figure 7: StudyII: Mean selection time per BUTTON SIZES per TECHNIQUE. Error bars are 95% CI

Considering that the selection time for Zoning and Tabbing included the release gesture, the across-the-board faster selection time is a very strong argument in favor of zoning. Considering that reach times for Pointing and Zoning were not significantly different, it implies that users took more time for air-tapping in pointing even for the largest button. This affirms our contention that precise selection is a problem for pointing, and is alleviated by summon & select with Zoning.

As mentioned, some buttons could auto-release by opening up a new window. Consequently, we take a look at how the three TECHNIQUES compare when the selection time for Tabbing and Zoning is also measured until the button click. Figure 8 shows the means: pointing ($3.2s/3.5s/4.6s$), tabbing ($2.7s/2.6s/2.8s$), zoning($1.9s/1.9s/2.0s$). The improvement over Pointing is clearer. In addition to earlier effects, posthoc tests with Bonferroni correction show that Tabbing's clicking time is also significantly faster than Pointing ($p < 0.01$).

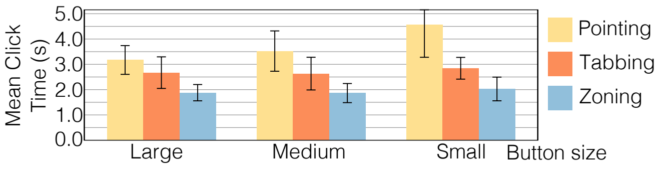

Figure 8: StudyII: Mean click time per BUTTON SIZE per TECHNIQUE. Error bars are 95% CI

Errors

We also calculated the incorrect button selection error% for every TECHNIQUE. Out of 288 total selections per TECHNIQUE, Zoning had 14 errors (4.8%), Tabbing had 18 (6.2%), and Pointing had 1 (0.3%). The errors were mostly due to users trying to do the task quickly and making mistakes as a result. For Pointing, trying to do the tasks quickly only resulted in missed clicks and not incorrect button selections.

Qualitative Results

Figure 9 shows the Likert scale response box-plot. Fridman tests show significant differences between the techniques on mental demand ($\chi^2(2) = 8.296, p < 0.05$), physical demand ($\chi^2(2) = 11.617, p < 0.01$), and frustration ($\chi^2(2) = 18.476, p < 0.001$). Posthoc Wilcoxon signed-rank tests with Bonferroni correction showed that mental demand for Zoning was significantly lower than Tabbing ($Z = -2.06, p < 0.05$), and Pointing ($Z = -2.392, p < .05$). Further, physical demand and frustration for both Zoning and Tabbing were significantly lower than Pointing ($Z = -2.946, p < .01; Z = -2.614, p < 0.01; Z = -3.082, p < .01; Z = -2.85, p < 0.01$).

Participants uniformly liked the concept of summoning. One participant with prior experience with Kinect said *"I like that I can keep my hand anywhere and still do what I want"*. 10 of 12 participants ranked Zoning as the most preferred TECHNIQUE for real-world use citing the use of two hands in Tabbing as a significant issue. Only two users preferred Tabbing, while Pointing was preferred by none. Compared to the sliders in Study II, here Zoning is heavily preferred. This could be because sliding in a high/low position was difficult or because tabbing through six buttons is more cumbersome.

Participants who preferred tabbing suggested it could be useful in gaming - *"It felt like a game to tab quickly and accurately. Tabbing is good for games where it'll be more immersive. Zoning can be used for public displays where the second hand is not free and people have less time."* One participant suggested a less physically demanding gesture for tabbing - *"I liked the second hand one for buttons 1,2,3 because it only required hand gestures, not movement of hand. But it gets too much to make a fist again and again 5-6 times. Can it just continue tabbing when the fist is made and then stop when I open the fist?"*. The Likert scale responses and qualitative feedback both indicated that participants felt less physically and mentally fatigued when using summon & select.

However, as the box plot shows, even though better than point & select, summon & select is still fatiguing to a certain extent. Summon & select does not require the user to focus while selecting small targets, but the users initially felt that remembering the gestures and their transitions was mentally demanding. This got easier once they got used to it during practicing. Consequently, the barrier of entry for point & select is much lower than summon & select.

In summary, (a) Zoning performance is consistent across BUTTON SIZES and directions and outperforms Pointing and Tabbing for all BUTTON SIZES. (b) Tabbing performance is consistent across BUTTON SIZES but dependent on the tab ordering of the buttons. This makes it faster than zoning for

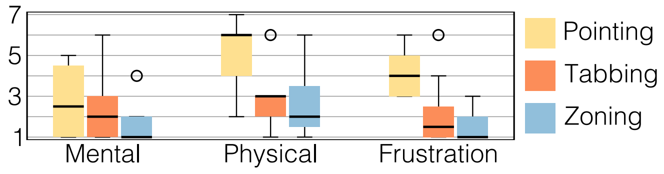

Figure 9: StudyII: Box plots for qualitative results for Button Selection

buttons 1-3, but slower than pointing for buttons 4-6. (c) Both Zoning and Tabbing had lower frustration and physical demand than Pointing. Zoning also had less mental demand than Pointing and Tabbing. (c) Pointing was slowest and least preferred, but it resulted in the least amount of errors.

HAPTIC FEEDBACK

One feedback that we received from multiple participants was that the interaction involves paying close attention to the visual feedback especially during dragging when the user repeatedly grips and releases the slider bar. In such a scenario, they suggested it would be beneficial to have other modes of feedback such as audio or haptic. Since the audio channel can be engaged elsewhere, we conducted a preliminary study to explore how haptic feedback can aid users in performing dragging in summon & select.

Multiple recent works have investigated wearable tactile feedback [15, 14, 26]. We designed haptic feedback such that in addition to confirmation of user actions, it also gives the user a hint of physical manipulation of the control. Therefore, we used miniature vibrotactile rings similar to [13] on the tips of the thumb and index finger. The rings are connected to a driving circuit placed on a wristband that communicates with the application via Bluetooth. Upon summoning, a 150ms pulse is played in both rings to indicate that the slider is summoned. When the user enters the drag state, a continuous pulse starts playing in both rings to mirror the grip of the slider bar. The pulse stops upon exit from the drag state. To reduce any perceived irritability from the vibration, the amplitude was set just above the perceivable level and the frequency was set at 350Hz. The 150ms pulse is played again upon release.

Study design

The study followed a within-subjects design to compare user performance and experience on dragging with and without haptic feedback. The interface consists of a single slider which the user is instructed to summon and then drag the bar to a target value similar to Study I. Each participant performed 10 trials, 5 each for haptic and no-haptic feedback. Five predefined target values, same for both conditions, are presented in a random order. We recorded the total time for a trial starting with the instruction until the release. Visual feedback is provided in both conditions.

12 different right-handed participants (mean age = 23.36), none of whom had experience with freehand interaction took part. The conditions were counterbalanced. Participants wore the rings in both conditions to reduce bias due to inconvenience from the wires and setup. Before each of the four conditions, the participants performed four practice trials. A small interview was conducted at the end. The study lasted

30mins. In total, we had 12 PARTICIPANTS × 2 HAPTIC × 2 SLIDERS × 5 REPETITIONS = 240 total trials.

Results

The mean total time per trial was 7.13*s* (95% CI [6.1, 8.2]) for the HAPTIC condition and 6.07*s* (95% CI [5.3, 6.9]) for No HAPTIC. No significant effect of haptics was found. However, 8 of 12 PARTICIPANTS preferred the HAPTIC condition. Participants liked the haptic confirmation - *"It just confirms that I'm doing it correctly, so I'm not focusing on the shading color changes every time I release it and grab it."*. Multiple PARTICIPANTS shared similar sentiments where they felt they could relax with the haptic feedback and not have to focus intently while doing the sliding. The relaxed focus could also mean that the PARTICIPANTS did not do the interaction as quickly as they could have. Although the difference is not significant, this might explain the HAPTIC condition's lower mean. Participants also mentioned the physical feeling - *"It's like I'm really holding something. I think I can do the task without visual feedback."*. However, some PARTICIPANTS did not like the sensations and wanted them to be more realistic - *"It's weird. Maybe I could just feel discrete notches go by. I don't like the continuous sensation."*

In summary, haptic feedback shows promise for summon & select and further studies should focus on improving the haptic actuation, and on whether haptic feedback alone without any visual feedback can enable similar performance thus allowing for a less visually engaged interaction.

DISCUSSION & FUTURE WORK

The results show that summon & select does not suffer from midas touch issues, enables pinpoint dragging accuracy, and is agnostic to button sizes. When used with zoning, it is faster on selection time, has lower physical & mental demand, and is more preferred than pointing. Consequently, it offers a realistic solution to point & select's problems of constant navigation, precise pointing, and out-of-bounds controls and lowers physical and mental fatigue for the user. Zoning is suitable for most scenarios, enables a faster seamless gestural interaction and can accommodate a large number of controls. However, Tabbing could be useful when the number of controls is less and two-handed interaction is not an issue. Plus, as mentioned earlier, it can be useful in mobile scenarios.

Integration with Point & select

While the merits of using summon & select in certain scenarios are clear, it does not apply to all large screen applications as discussed earlier. Consequently, it needs to be explored how summon & select can work alongside point & select. A crude way would be to have summon & select as the default for some apps and point & select for the others and the user can switch the defaults according to their preference using an explicit gesture. However, it would be better to integrate summon & select in the current interfaces as a shortcut alternative similar to a right click menu or keyboard shortcut. For instance, the user navigates the pointer with an open palm and if the on-screen pointer is not over a clickable area, the user can perform summon & select starting with the summoning gesture. If the pointer is over a clickable area, then the hand gestures are not recognized as summoning gestures similar to how right click does not work on controls. In such a scenario, integration with global semaphoric gestures such as swipes needs to be investigated. This is a topic for future exploration.

Generalizability and Learnability

Because summon & select is independent of target size and is less dependent on distance, we believe that the results will apply for interfaces with a larger number of buttons and sliders. However, for interfaces that consist of a larger diversity of controls with different summon gestures, the cognitive load might be higher which could affect performance when compared to point & select. This needs to be investigated in detail. Summon & select needs to be investigated for an end-to-end real-world application that consists of multiple control types and more complex interaction sequences. With more control types, it will be worth exploring how to incorporate learning-while-doing mechanisms for the gestures. These could be, for instance, iconic representation besides the controls, or animated representations which are invoked when the user mouse-overs the controls.

CONCLUSION

We introduced summon & select, a fast, low-fatigue free-hand interaction model for interface controls in mid-air. To this end, we describe its design and how it overcomes challenges including the Midas touch problem, its conceptual advantages and limitations against point & select. We conduct two studies. The first study shows that the interaction can be performed without Midas touch issues, the drag state enables highly precise dragging, and that Zoning outperforms Tabbing as the number of controls gets higher. In a second study, we compare summon & select to pointing and show that it outperforms pointing both quantitatively and qualitatively. We end with a discussion, suggestions for future work, and a preliminary investigation of haptic feedback in summon & select. Point & select was never designed to be performed in mid-air. We believe summon & select is a significant step towards offering a compelling alternative.

ACKNOWLEDGMENTS

This project was supported by the European project Happiness, H2020 645145 and NSERC Canada, #482670.

REFERENCES

1. Ackad, C., Clayphan, A., Tomitsch, M., and Kay, J. An in-the-wild study of learning mid-air gestures to browse hierarchical information at a large interactive public display. In *Proc. UbiComp '15* (sep 2015), 1227–1238.
2. Aslan, I., Uhl, A., Meschtscherjakov, A., and Tscheligi, M. Mid-air Authentication Gestures: An Exploration of Authentication Based on Palm and Finger Motions. In *Proc. ICMI '14* (2014), 311–318.
3. Bailly, G., Walter, R., Müller, J., Ning, T., and Lecolinet, E. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. In *Proc. INTERACT'11* (2011), 248–262.
4. Balakrishnan, R. "beating" fitts' law: Virtual enhancements for pointing facilitation. *Int. J. Hum.-Comput. Stud. 61*, 6 (Dec. 2004), 857–874.

5. Banerjee, A., Burstyn, J., Girouard, A., and Vertegaal, R. MultiPoint: Comparing laser and manual pointing as remote input in large display interactions. *Int. J. Hum. Comput. Stud. 70*, 10 (oct 2012), 690–702.

6. Bateman, S., Mandryk, R. L., Gutwin, C., and Xiao, R. Analysis and comparison of target assistance techniques for relative ray-cast pointing. *Int. J. Hum. Comput. Stud. 71*, 5 (may 2013), 511–532.

7. Baudel, T., and Beaudouin-Lafon, M. Charade: remote control of objects using free-hand gestures. *Commun. ACM 36*, 7 (jul 1993), 28–35.

8. Bragdon, A., and Ko, H.-S. Gesture select: acquiring remote targets on large displays without pointing. In *Proc. CHI '11* (2011), 187.

9. Buxton, W. A three-state model of graphical input. In *Proc. INTERACT '90*, North-Holland Publishing Co. (Amsterdam, The Netherlands, The Netherlands, 1990), 449–456.

10. Carter, M., Velloso, E., Downs, J., Sellen, A., O'Hara, K., and Vetere, F. Pathsync: Multi-user gestural interaction with touchless rhythmic path mimicry. In *Proc. CHI '16*, ACM (New York, NY, USA, 2016), 3415–3427.

11. Clarke, C., Bellino, A., Esteves, A., Velloso, E., and Gellersen, H. Tracematch: A computer vision technique for user input by tracing of animated controls. In *Proc. UbiComp '16*, ACM (New York, NY, USA, 2016), 298–303.

12. Freeman, E., Brewster, S., and Lantz, V. Do That, There: An Interaction Technique for Addressing In-Air Gesture Systems. In *Proc. CHI '16* (2016), 2319–2331.

13. Gupta, A., Irudayaraj, A., Chandran, V., Palaniappan, G., Truong, K. N., and Balakrishnan, R. Haptic Learning of Freehand Semaphoric Gesture Shortcuts. In *Proc. UIST 2016* (2016).

14. Gupta, A., Irudayaraj, A. A. R., and Balakrishnan, R. HapticClench: Investigating Squeeze Sensations using Memory Alloys. In *Proc. UIST '17* (October 2017).

15. Gupta, A., Pietrzak, T., Roussel, N., and Balakrishnan, R. Direct Manipulation in Tactile Displays. In *Proc. CHI '16* (2016), 3683–3693.

16. Gustafson, S., Bierwirth, D., and Baudisch, P. Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proc. UIST '10* (2010), 3–12.

17. Ismair, S., Wagner, J., Selker, T., and Butz, A. MIME: Teaching Mid-Air Pose-Command Mappings. In *Proc. MobileHCI '15* (aug 2015), 199–206.

18. Jansen, Y., Dragicevic, P., and Fekete, J.-D. Tangible remote controllers for wall-size displays. In *Proc. CHI '12* (2012), 2865.

19. Karam, M., and Schraefel, m. c. A Taxonomy of Gestures in Human Computer Interactions, aug 2005.

20. Kou, Y., Kow, Y. M., and Cheng, K. Developing Intuitive Gestures for Spatial Interaction with Large Public Displays. In *Proc. Third Int. Conf. Distrib. Ambient. Pervasive Interact. - Vol. 9189*. 2015, 174–181.

21. Lin, S.-Y., Shie, C.-K., Chen, S.-C., and Hung, Y.-P. AirTouch panel: a re-anchorable virtual touch panel. In *Proc. MM '13* (2013), 625–628.

22. Liu, M., Nancel, M., and Vogel, D. Gunslinger: Subtle Arms-down Mid-air Interaction. In *Proc. UIST '15* (2015), 63–71.

23. Nakamura, T., Takahashi, S., and Tanaka, J. Double-Crossing: A New Interaction Technique for Hand Gesture Interfaces. In *Comput. Interact.* 2008, 292–300.

24. Nancel, M., Chapuis, O., Pietriga, E., Yang, X.-D., Irani, P. P., and Beaudouin-Lafon, M. High-precision pointing on large wall displays using small handheld devices. In *Proc. CHI '13* (2013), 831.

25. Ni, T., Bowman, D. A., North, C., and McMahan, R. P. Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures. *Int. J. Hum. Comput. Stud. 69*, 9 (aug 2011), 551–562.

26. Pasquero, J., Stobbe, S. J., and Stonehouse, N. A haptic wristwatch for eyes-free interactions. In *Proc. CHI '11* (may 2011), 3257.

27. Rateau, H., Grisoni, L., and De Araujo, B. Mimetic interaction spaces: controlling distant displays in pervasive environments. In *Proc. IUI '14* (2014), 89–94.

28. Ren, G. *Designing for Effective Freehand Gestural Interaction*. PhD thesis, University of Bath, 2013.

29. Ruiz, J., and Vogel, D. Soft-Constraints to Reduce Legacy and Performance Bias to Elicit Whole-body Gestures with Low Arm Fatigue. In *Proc. CHI '15* (apr 2015), 3347–3350.

30. Song, P., Goh, W. B., Hutama, W., Fu, C.-W., and Liu, X. A handle bar metaphor for virtual object manipulation with mid-air interaction. In *Proc. CHI '12* (2012), 1297.

31. Steins, C., Gustafson, S., Holz, C., and Baudisch, P. Imaginary devices: Gesture-based interaction mimicking traditional input devices. In *Proc. MobileHCI '13* (2013), 123–126.

32. van de Camp, F., Schick, A., and Stiefelhagen, R. *How to Click in Mid-Air*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, 78–86.

33. Vatavu, R.-D., and Radu-Daniel. Point & click mediated interactions for large home entertainment displays. *Multimed. Tools Appl. 59*, 1 (jul 2012), 113–128.

34. Vogel, D., and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. In *Proc. UIST '05* (2005), 33–42.

35. Walter, R., Bailly, G., and Müller, J. StrikeAPose: revealing mid-air gestures on public displays. In *Proc. CHI '13* (apr 2013), 841.