Gaze+Gesture: Expressive, Precise and Targeted Free-Space Interactions

Ishan Chatterjee[1] Robert Xiao[2] Chris Harrison[2]

[1]Harvard University
Electrical Engineering
29 Oxford Street, Cambridge, MA 02138
ishanchatterjee@college.harvard.edu

[2]Carnegie Mellon University
Human-Computer Interaction Institute
5000 Forbes Avenue, Pittsburgh, PA 15213
{brx, chris.harrison}@cs.cmu.edu

ABSTRACT

Humans rely on eye gaze and hand manipulations extensively in their everyday activities. Most often, users gaze at an object to perceive it and then use their hands to manipulate it. We propose applying a multimodal, gaze plus free-space gesture approach to enable rapid, precise and expressive touch-free interactions. We show the input methods are highly complementary, mitigating issues of imprecision and limited expressivity in gaze-alone systems, and issues of targeting speed in gesture-alone systems. We extend an existing interaction taxonomy that naturally divides the gaze+gesture interaction space, which we then populate with a series of example interaction techniques to illustrate the character and utility of each method. We contextualize these interaction techniques in three example scenarios. In our user study, we pit our approach against five contemporary approaches; results show that gaze+gesture can outperform systems using gaze or gesture alone, and in general, approach the performance of "gold standard" input systems, such as the mouse and trackpad.

Categories and Subject Descriptors

H.5.2 [**Information interfaces and presentation (e.g., HCI)**]: User Interfaces – *input devices and strategies*.

Keywords

Eye tracking; touch-free interaction, free-space gestures; input technologies; interaction techniques; sensors; pointing; cursor.

1. INTRODUCTION

Eye tracking and free space gesturing have individually been explored as methods for control in interactive systems. Like all input approaches, they have strengths and weaknesses. For example, eye movement is extremely quick, precise, and generally requires low effort (see e.g., [39] for an extended discussion). Further, gaze naturally corresponds to a user's focus of attention, which can be used for directed control of user interfaces (e.g., selection), or monitored passively to track a user's attention over time.

Unfortunately, eye tracking continues to be relatively inaccurate – on the order of ±1 degree, which translates to centimeter inaccuracies for monitors roughly 0.5 meters away. This is due to both natural jitter of the eyes and sensing limitations [27]. This precludes common fine-grained operations, such as clicking buttons, selecting text and opening menus. Although superior accuracies can be achieved with head-worn setups, these can be intrusive and generally cumbersome [26]. Furthermore, as the eyes are primarily used for sensory input, it is fatiguing to use them for manipulation tasks such as continuous gestures [44]. This significantly limits the modal expressivity of gaze, with systems often relying on winks, blinks and dwells to trigger interactive functions [10, 44].

Conversely, our hands excel at continuous manipulation and can occupy a wide variety of poses (i.e., gestures), which can be used to trigger interactive functions. This capability is not surprising given that our hands are our chief means for manipulating the world around us [7, 21, 42]. However, by operating in free-space (i.e., in the air), we lose many of the tactile affordances and visual references that make our hands dexterous at small scales. This makes targeting, for example, considerably more time consuming and fatiguing as compared to operating on a touchscreen. Previous free-space gesturing systems often mitigate this by having a low control-device (CD) gain [6], but this in turn means the hands must traverse a larger volume, compounding issues of fatigue.

To summarize, gaze is particularly well suited to rapid, coarse, absolute pointing, but lacks natural and expressive mechanisms to support modal actions. Conversely, free space gesturing is slow and imprecise for pointing, but has unparalleled strength in gesturing, which can be used to trigger a wide variety of interactive functions. Thus, these two modalities are highly complementary. By fusing gaze and gesture into a unified and fluid interaction modality, we can enable rapid, precise and expressive free-space interactions that mirror natural use. Moreover, although both ap-

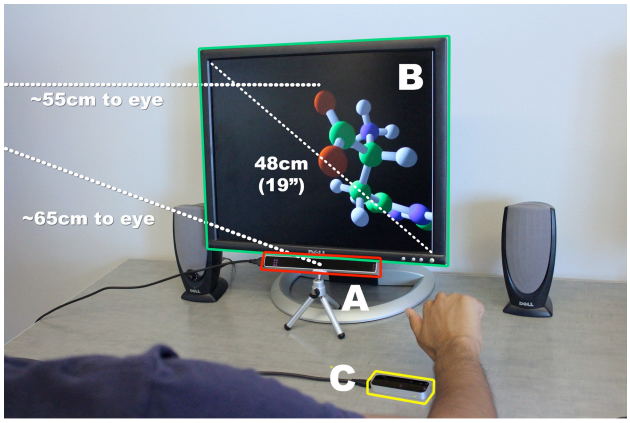

Figure 1. The apparatus used for our example interaction techniques and evaluation. The eye tracker (A) is located beneath the monitor (B), while the hand tracker (C) lies under the user's hand.

ICMI '15, November 09 - 13, 2015, Seattle, WA, USA

ACM 978-1-4503-3912-4/15/11…$15.00
DOI: http://dx.doi.org/10.1145/2818346.2820752

proaches are independently poor for pointing tasks, combining them can achieve pointing performance superior to either method alone. This opens new interaction opportunities for gaze and gesture systems alike.

There are two main contributions of this work. Foremost, we present a series of gaze+gesture interaction techniques, contextualized in three example application scenarios. To help explore the interaction design space of gaze and free-space gesture, our development efforts were guided by an interaction taxonomy we extended from Stellmach and Dachselt's [38] work on gaze and direct touch. Secondly, we present a user study that rigorously compares gaze+gesture against five contemporary approaches. Such an *apples-to-apples* comparison is vital, as the present literature employs a variety of study designs that generally preclude direct comparison. The results of this study suggest our approach has a similar index of performance to "gold standard" input methods, such as mouse, and can target small elements that are generally inaccessible to gaze-only or gesture-only systems.

2. RELATED WORK

Related work to our gaze+gesture system can be divided into three primary categories: gesture-centric input approaches, multimodal gaze-based interactions and gaze-and-gesture input systems.

2.1 Multimodal Gesture Input

Compared with touch, keyboard or mouse input, hand gesture input can better utilize the expressive power of the human hand to enable rich interactions. For instance, hands excel at 3D manipulation tasks, leading to a profusion of hand-gesture based 3D input techniques (see e.g., [16, 18]). Thus, automatic tracking of hands in free-space has long been sought after. Early systems used mechanical gloves [47] or wearable sensors [36], but it is now possible to avoid instrumenting the user using e.g., computer vision [8].

Perhaps due to the expressive nature of gesture input, multimodal gesture-centric systems are much less common. By far, the most common multimodal pairing for gesture input is speech. As shown in e.g., Put-That-There [2], the flexible, free-form nature of speech input can complement the more directed nature of gesture input. Researchers have also paired free-space gesturing with touch [41], pen input [17] and EMG sensing [45].

2.2 Multimodal Gaze Input

Computerized eye trackers were first used interactively in seminal work such as the 1989 Erica system [13], which was designed to enable paralyzed users to select menu items using only eye gaze.

Fundamentally, pure-gaze input approaches tend to suffer from "Midas Touch" issues [15] due the lack of a natural input delimiter. To alleviate this, gaze is often combined with another input modality, e.g., mice [44], keyboards [19], voice [34], touch input [28, 38], head movements [35] and even facial EMG [23]. Gaze can be used to rapidly target items, which are then selected by using the second input modality. As Zhai et al. [44] showed, such a hybrid approach can significantly enhance the performance of an input system, as compared to either approach alone. This approach can also compensate for gaze inaccuracies by relying on the more accurate modality (e.g., mouse) for the final selection of targets.

More recently, with the proliferation of touchscreen devices, gaze and touch input systems have become an active area of research. Gaze-Touch [28] combines gaze interaction with touch screen input, focusing primarily on selection and spatial manipulation of objects, such as dragging, scaling, rotating and grouping. Modal gestures are partially explored, but are limited to finger chording. Finally, there is no mechanism presented for targeting objects smaller than the gaze tracker's accuracy limits. Stellmach and Dachselt further explored gaze and touch in [37, 38], with the use of an eye tracker and a handheld touchscreen device, presenting interaction techniques for manipulating both small and large objects. Of note, although both papers evaluate their individual technique, neither compares against other techniques, an issue we attempt to rectify in our study design.

Importantly, our work differs from the prior gaze+touch work in that it explores the interaction design space of two non-tactile input methods. Free-space gesture is well established as being different than direct touch [40], with unique challenges and benefits, and so the interactions and methods also naturally differ. Touch has the advantage of absolute positioning with respect to the touchscreen borders, while mid-air gestures only offer relative positioning. Additionally, touch has superior physical affordances and an innate ability to clutch [40]. However, mid-air gestures can be more expressive as the free-space gesture set is larger than that of conventional, planar touch interfaces.

2.3 Gaze and Free-Space Gesture Input Systems

Despite the appearance of inexpensive eye trackers (such as the Eye Tribe Tracker [9]) and consumer-level gesture trackers (such as the Microsoft Kinect [24] and Leap Motion Controller [20]), comparatively little work has been done on combining gaze with free-space gesture input. Chen [4] examines the role of gaze and gesture in human communication. Pouke et al. [29] demonstrate a system that uses gaze input to select large targets on a screen, and a hand-worn accelerometer to sense discrete gestures. This system did not compensate for gaze-tracking inaccuracies, leading to poor results. Yoo et al. [43] describe an approach using a head tracker and depth camera to enable targeted interactions with large targets on a wall-sized display. Hales et al. [11] present a system in which users select physical objects by gazing at markers printed on them, and use hand gestures to interact or control them.

Our present system advances upon this prior gaze and gesture work in four significant ways. First, we provide interaction techniques for selecting objects of any scale, even those that are below the minimum threshold of gaze accuracy. Second, we enable continuous free-space manipulations with our gestures, whereas past systems focused on discrete hand gestures. Third, we use a structured taxonomy to guide the creation of interaction techniques, in which we situate familiar digital tasks. Finally, we conducted an evaluation benchmarking our technique against several of the aforementioned techniques, and against common input devices.

3. SYSTEM

As a vehicle for exploration, we developed a proof of concept system using off the shelf components. Importantly, we chose to use minimally invasive, low-cost, consumer-grade devices. Although offering lower accuracy compared to professional systems (e.g., head mounted cameras), they are the types of sensors that are most likely to be integrated into consumer electronics – and thus user experiences – in the near future.

Specifically, we use an off-the-shelf, $99 Eye Tribe Tracker [9]. This device consists of an infrared emitter (to illuminate the eyes) and a pair of cameras inside a bar-shaped case, mounted on a miniature tripod. This is positioned below a conventional 19" LCD monitor, set back ~55cm from the user's chest (Figure 1). Following a per-session calibration, the Eye Tribe software produces a stream of screen-space X/Y gaze coordinates at 30 FPS.

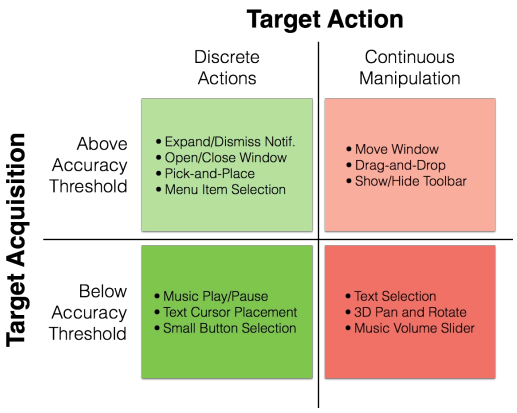

Figure 2. A taxonomy for gaze+gesture interaction techniques, showing sample interactions in each division.

To track users' free-space hand movements and gestures, we use an off-the-shelf, $80 Leap Motion Controller [20]. This hand tracker also uses infrared light (to illuminate the skin) and a pair of cameras. We use the Leap Motion software to obtain continuous 3D hand pose estimation, and employ their API to detect predefined gestures, such as pinching. Our interactions only used relative motions of the hand, and thus we did not need to calibrate the Leap Motion against the eye tracker or the display. We situate this device to the front and right of the computer monitor in order to capture the movement of the user's right hand, and to avoid occluding the eye tracker with the hands.

Finally, we used a desktop computer running Windows 8.1 to run our software, example applications and user study, which were developed in C++ using the respective device SDKs and the OpenFrameworks graphics library. It should be noted, however, this desktop setup was chiefly for prototyping and experimentation purposes, and that our techniques and exploration are applicable to other gaze+gesture contexts, such as large, vertical displays.

4. INTERACTION TECHNIQUES

Our approach divides interactive tasks between the visual and gestural channel in a natural fashion. The eyes, possibly in conjunction with the hands, specify the user's focus or target area (*target acquisition phase*), followed by the hands performing gestural actions or spatial manipulations on the virtual objects (*target action phase*). Emphasizing user comfort, we avoid interaction techniques requiring continuous gaze, as it is fatiguing and unnatural [15]. Thus, gaze is never used in the *target action phase*. To guide our work, we started with a taxonomy of the interaction design space from prior work on gaze+touch systems and extended it to include the capabilities of our free-space gesture tracking.

4.1 Target Acquisition Phase

Inaccuracies and noise inherent in gaze tracking systems results in a target size threshold, below which targets cannot be accurately determined by gaze alone. As noted in Stellmach and Dachselt [38] and Zhai et al. [44] there exists an area around the detected gaze position in which one can be reasonably sure the intended target resides. This natural demarcation forms a size threshold for targets: targets larger than e.g. the 95% confidence interval around the gaze position can be selected by gaze alone, while targets smaller than this size threshold require additional information from the other input modality to be accurately selected. Therefore, we split targets into two categories: those *above* the accuracy threshold (only coarse targeting required) and *below* the accuracy threshold (finer targeting possibly required), similar to the method described in Stellmach and Dachselt [38]. Although eye tracker accuracy will improve with better technology, there will always be an upper limit on accuracy due to intrinsic eye behavior such as tremors and microsaccades, and so this distinction will continue to be relevant.

4.1.1 Above Accuracy Threshold

When the gaze target is larger than the eye tracker's accuracy threshold, the target can be immediately determined using gaze information alone. Typical above-threshold targets include large icons and buttons, sliders, windows, text paragraphs, and screen edges. The subsequent hand gesture therefore acts directly on the target. In some cases, the intended target can be ambiguous, not because of size, but because of z-ordering or hierarchy – for example a file icon and the window containing that file. To disambiguate, we use different gestures (e.g. using a pinch for icons and a grab for windows), though other strategies are possible.

4.1.2 Below Accuracy Threshold

When the desired target is smaller than the eye tracker's accuracy threshold, the application must disambiguate between proximate targets in some way. Typical below-threshold targets include individual text characters, map positions and small buttons.

We devised three *target disambiguation* strategies, some of which utilize gesture input in addition to the gaze input. 1) *Loose targeting*: an application can choose to accept "loose" targeting, which is appropriate if a precise target is not critical, e.g. when choosing a focal point for zooming a map. 2) *Contextualization*: different gestures can be used to disambiguate between multiple possible targets. For example, the controls on a music player might lie below the accuracy threshold. In this case, we can disambiguate between "previous", "play", and "next" controls on a music player by using left swipe, click, and right swipe gestures respectively. Finally, 3) *fine selection*: we can resort to a high-precision mode

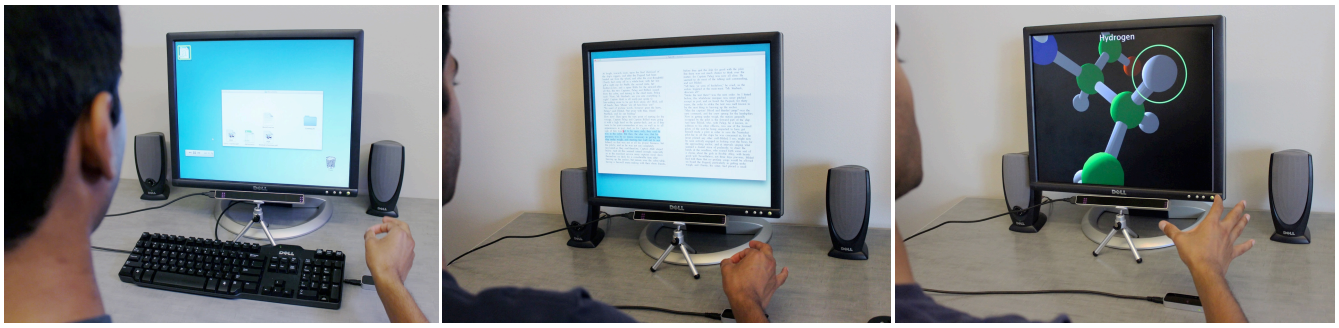

Figure 3. We built three example scenarios as a sandbox for our interaction techniques. Left-to-right: desktop, word processor, and 3D model viewer. Please also see the Video Figure.

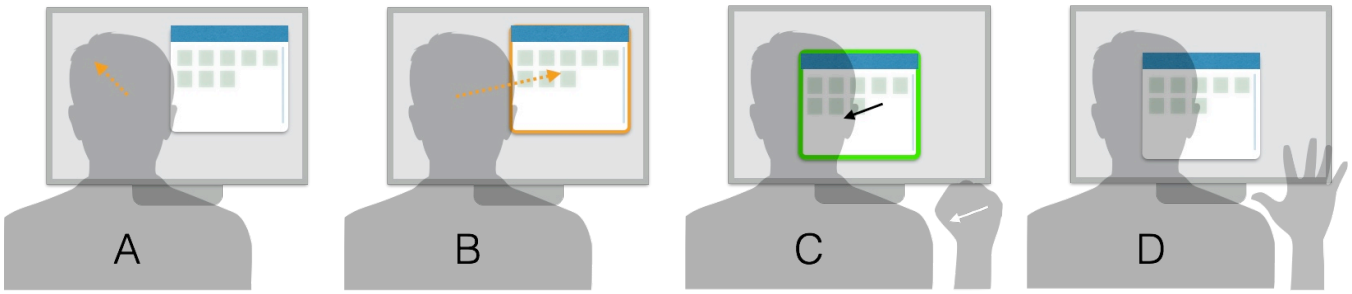

Figure 4. An example of an above gaze threshold interaction using continuous hand gesture interaction. (A) User is working. (B) User looks at window they wish to move. (C) User performs a grab gesture, after which the window tracks with the hand until (D) the fist is released.

that uses a low CD-gain manipulation gesture (e.g. pinch and drag) to adjust the selection (e.g. to precisely position a cursor within text).

4.2 Target Action Phase

Once a UI target has been accurately selected, a subsequent selection or manipulation can be performed. As noted above, continuous gaze control is uncomfortable and therefore only gesture is used for the *action phase*. These gestures manifest in one of two ways: for *discrete actions* (e.g. pinching, swiping, grabbing) or *continuous manipulation* (e.g. moving, dragging, rotating).

4.2.1 Discrete Actions

A discrete hand gesture is a single event triggered by a specific hand pose or motion. These gestures can be used for actions commonly associated with clicks and taps on traditional interfaces, e.g. opening and closing documents, activating buttons, and dismissing dialogs. Typical discrete gestures include closing or opening the fingers (e.g., touching two fingers for "pinching", making a fist for "grabbing"), threshold-crossing movements with the hand (e.g., moving horizontally for "swiping", vertically for "clicking"), or rapid movements with one or more fingers (e.g., "flicking" movements). Discrete hand gestures are therefore suited to the purpose of target selection such as button press or triggered actions (e.g. see music player play/pause example).

4.2.2 Continuous Manipulation

Continuous manipulation with the hands consists of relative movements and provides continuous positional information in three dimensions. We preface these manipulations with an activation gesture (e.g., pinching, grabbing), which establishes the origin point for the manipulation and enables clutching. Common continuous manipulations include moving the hand while pinching or grabbing (for e.g., "dragging" items or "panning" a scene), translating and rotating the hands in 3D (to manipulate a virtual camera or object), and spreading two hands for "zooming". Continuous manipulation gestures are therefore suited to the purpose of fluid target positioning and manipulation.

4.3 Taxonomy

Using this interactive breakdown, a 2x2 taxonomy naturally arises (Figure 2). On one axis, we have the *target acquisition phase*, which specify targets either above or below its accuracy threshold. On the other axis, the *target action phase* consists of discrete gesturing and continuous manipulation. Figures 4 through 7 offer examples for each quadrant of the taxonomy.

5. EXAMPLE APPLICATIONS

Using our taxonomy as a guide, we created a series of exemplary interaction techniques to illustrate the potential and flexibility of our approach. To contextualize these interactions, we embed them within three illustrative scenarios – a desktop environment, word processor, and 3D object viewer (Figure 3). See the Video Figure for a real-time demonstration. We can envision these techniques augmenting both classic desktop computing experiences (e.g., with a keyboard), as well as large wall displays, kiosks, or distant displays, where keyboards and mice are impractical.

For each scenario, we show how these interaction techniques can be used to e.g., select, navigate, and otherwise manipulate content in rapid and expressive ways. Each application is a fully implemented prototype (i.e., all described actions are available at every moment in time), though obviously not all possible functions are implemented. Finally, we note that the particular designs of these applications were adjusted to demonstrate a variety of functionalities, and do not represent idealized systems. Designing a practical

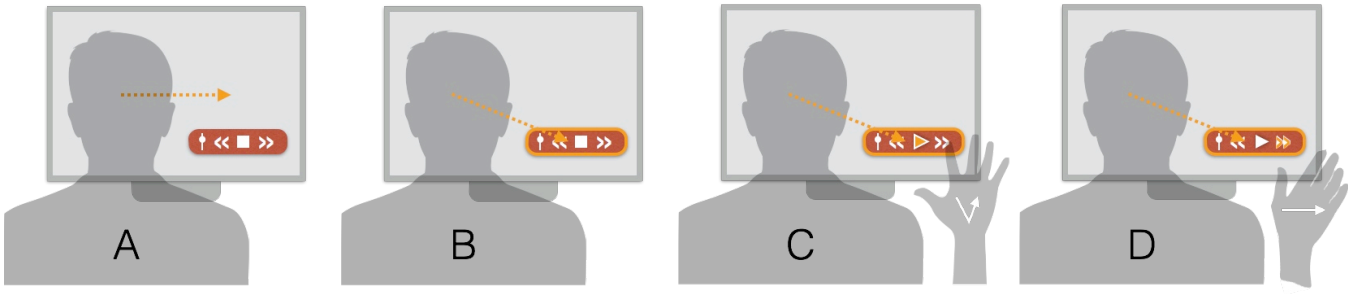

Figure 5. An example of a below gaze threshold interaction using discrete hand gesture. (A) While user is working, he wishes to play some music. (B) User looks at the music player play button (currently stopped). (C) Buttons on the music player are too small for gaze targeting alone, so gestures are employed to resolve the ambiguity. In this case, a downward bounce of the hand toggles between pause and play. (D) User advances to the next song with a right hand swipe.

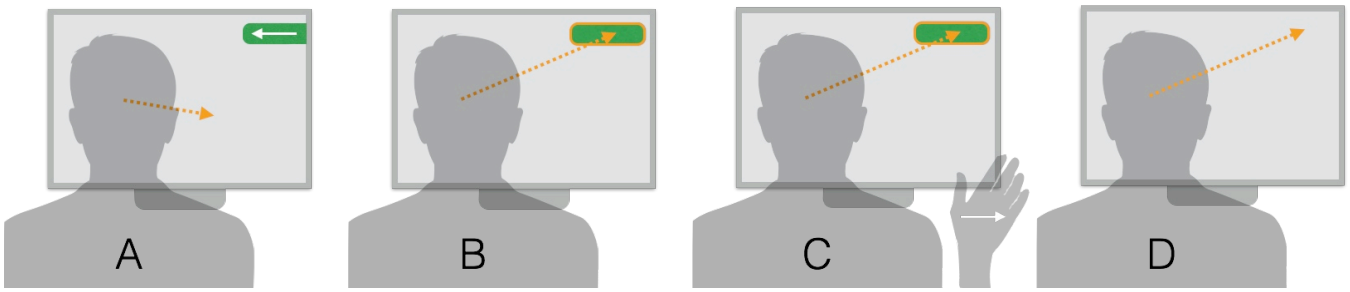

Figure 6. An example of an above gaze threshold interaction using discrete hand gesture. (A) While user is working, a notification window slides in from the top right. (B) User looks at (i.e., reads) notification. (C) User decides to dismiss notification, and performs a rightward hand swipe. (D) Notification is dismissed.

gaze+gesture system would require an in-depth examination of how all of the gestures would interplay across applications to form cohesive and intuitive experiences. We provide these example scenarios not necessarily as end-user applications, but more as a vehicle to frame to the individual application techniques, offer a familiar context, and to provide a label by which to refer to the individual interactions (such as in Figure 2).

5.1 Desktop Scenario

We included several GUI elements in our desktop scenario (Figure 3, left). Foremost, we have icons and windows. To disambiguate between file and window selection, two different gestures are used. A "grab" gesture (making a fist) selects the gazed-at window (Figure 4, B), which can then be moved by translating the hand in space (Figure 4, C) and closed with a downward movement. Alternatively, by using a "pinch" gesture with the thumb and index finger, the gazed-at file is selected. Once selected, the file can be dragged by translating the hand, or opened with an upward motion. Of note, if the gaze area contains no windows or files, the action is ignored. Lastly, we also implemented a "pick and place" method where instead of dragging a file to a destination, the user can simply pinch to select the file, look at where they want the file to go, and release the pinch, instantly transporting it.

We also provide a minimized music player, the buttons of which (purposely) lie below the gaze accuracy threshold (Figure 5). To trigger functionality, different gestures must be used. Here, we use left and right swipes to move to the previous and next track respectively. A downward hand bounce toggles pause and play. Pinching plus lateral translations of the hand adjusts the volume.

Finally, the scenario also includes slide-in notifications, which, once looked at, can be opened with a left swipe or dismissed with a right swipe (Figure 6).

5.2 Word Processor Scenario

We created a basic text editor which uses gaze+gesture. Users can move the text cursor by gazing at the target location and pinching to coarsely position the cursor (Figure 7, A and B). Users then drag to perform fine adjustment on the cursor position, allowing character level positioning (Figure 7, C). After placing the cursor, the user may select text in two ways. 1) To select a small region of text, the user can briefly unpinch and then drag to select. 2) To select a large region of text, the user can unpinch, gaze elsewhere, and then double-pinch and drag to define the end position of the text selection.

Once text has been selected, the user can flick two fingers upwards to summon a contextual menu, displaying common operations such as bold, italicize, highlight, copy, cut, etc. Gazing at a menu item highlights it, and flicking the fingers upwards again selects the operation. Alternatively, flicking the fingers downwards cancels the menu. Because the menu is only made visible on demand and selection is done with a quick glance, it can be as large as needed.

5.3 3D Model Viewer Scenario

To demonstrate how gaze+gesture works in 3D contexts, we created a molecule viewer. Users can select, view and explore different molecules, and inspect details within the structure (e.g. atoms).

A toolbar on the right side of the screen provides access to a database of molecules. The user summons it by gazing at the right edge of the screen, using a grab gesture, and pulling leftward. Users select a molecule from the toolbar by gazing at the desired item and pinch-dragging it to the main viewing area. The toolbar can likewise be closed by gazing, grabbing and swiping right.

When looking at molecules, the closest atom to the gaze location is highlighted. Users can translate and rotate the molecule in 3D

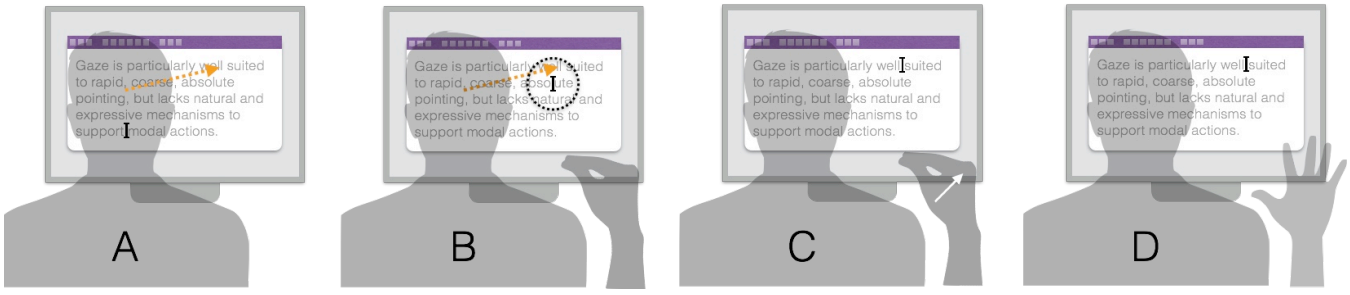

Figure 7. An example of a below gaze threshold interaction using continuous hand gesture interaction to reposition a cursor. (A) The user wishes to reposition the cursor, and gazes at the destination. (B) He pinches his hand, and the cursor is moved to the approximate gaze location (uncertainty shown as a dotted circle). (C) By translating the pinched hand, the user can finely manipulate the cursor position. (D) When satisfied with the position, the pinch is released. The user can e.g., resume typing.

space by grabbing with one or two hands. A bimanual two-finger pinch gesture is used to zoom the entire view about the gaze point. This allows for much smoother navigation to the specific section of the model within the user's focus. Finally, a single pinch is used to bring up additional information about an atom.

5.4 Evaluation

To better understand the pointing performance of gaze+gesture, we ran a Fitts-style study. Of note, the diversity of study designs and targeting tasks make direct comparisons between pointing techniques difficult, especially when they are multimodal. Replicating others' apparatuses is equally problematic. As such, we endeavored to include a suite of comparative techniques, drawn from the literature and popular use to better ground our results and discussion [3, 10, 15, 21, 25, 30, 39].

Specifically, we include two gaze-driven techniques, one gesture-only technique, the mouse, and the trackpad. Although we described several techniques using gaze+gesture, we chose to evaluate our post-gaze, fine adjustment method. We hypothesized this would allow for rapid distance traversals whilst simultaneously permitting small-scale targeting (i.e., best of both worlds), which is where current gaze and gesture systems struggle.

Of note, there is an ongoing debate over whether Fitts' Law applies to gaze input, with research arguing both for [25, 39, 41] and against [5, 33]. Thus we caution that the raw numerical results generated from the study may not represent the intrinsic performance index needed to enable between-paper comparisons. Nonetheless, they do serve as a useful *relative* measure of performance within this study, with which we use to assess our results.

5.5 Participants

Twenty-two participants were recruited. Three participants' data were dropped: two due the inability of the Leap SDK to recognize participants' pinching gesture, and one that was a performance outlier (>3 SD above mean). Out of the 19 participants analyzed, 8 were female and 17 were right-handed; mean age was 25.8. None had prior experience with an eye tracker or free hand gesture control. Due to calibration issues with the Eye Tribe Tracker, we could not recruit users who wore glasses.

5.6 Conditions

In addition to testing gaze+gesture, we evaluated five additional input methods that we felt were interesting and relevant points of comparison or strong baselines. Appropriate gain values were informed by pilot studies.

- ***Mouse****:* The mouse has repeatedly been shown to be an excellent pointing device (e.g. [1, 3]), and as such, serves as a gold standard baseline. Targets were selected by clicking.
- ***Trackpad****:* Though it is less efficient than a mouse, the trackpad is an equally ubiquitous input mechanism, and so also serves as an aggressive baseline. Targets were selected by clicking down on the pad.
- ***Gaze+Dwell****:* Gaze coupled with dwell for selection has been used in e.g., [15, 21, 25, 39]. We use a 500ms dwell time, which was shown to have the highest performance out of dwell-based methods in [22]. Gaze controls absolute cursor location; no clutching is required.
- ***Gaze+Blink****:* Gaze plus blink for selection was used in e.g., [10, 30]. We evaluated a range of blink durations from the literature [30] during piloting, and settled on a blink duration of 100ms as a trigger as it was found to have the lowest false positive rate and overall fastest time. Gaze again controls absolute cursor location, without clutching.
- ***Gesture+Dwell****:* Many systems have used free-space pointing or translation plus a dwell for selection (see e.g., [12, 31]). We used a 500ms dwell for selection. The Leap's field of view is insufficient to address the entire screen at a useable CD gain. Thus, a clutching mechanism is needed, for which we use a pinch gesture. When pinched, 1cm of hand movement corresponds to ~40px of cursor movement. When un-pinched, the hand is not tracked, enabling clutching.
- ***Gaze+Gesture****:* When the hand is un-pinched, the approximate gaze position is shown as a blue outlined box superimposed on screen (Figure 8). When the hand is pinched, cursor appears at the center of the blue box, and the blue outline is hidden. The cursor can then be controlled by gesture, with 1cm of hand movement corresponding to ~20px of cursor movement.

5.7 Apparatus and Calibration

For conditions utilizing gaze, gesture, or both modalities, we use the apparatus described in the Implementation section (Figure 1). For the mouse and trackpad conditions, the input device was placed on the table. Users were seated 55 cm in front of a 19" (48 cm) 1280 x 1024 LCD monitor. At this distance, 32px corresponded to approximately 1° of visual arc. Our study software mediated all input for cursor control, displayed graphics, and logged study data.

No calibration was needed for gesture tracking. Calibrating the gaze tracker consisted of looking at nine targets on the screen, taking about 14 seconds to complete. The calibration procedure was repeated until the calibration routine reported a "four star" or "five star" (best possible) calibration result, corresponding to maximum inaccuracies of < 0.7° and < 0.5° respectively (as claimed by the Eye Tribe documentation). Users usually had to calibrate one to three times to achieve this result. In practice, we found accuracy to be closer to ±3°. Though participants were asked to hold their position and posture as best possible, we did not institute any strict rules.

5.8 Procedure

We use the ISO 9241-9 Fitts' study standard procedure [14] as it is widely used to produce targeting performance data comparable across studies. Each participant tested all six input methods (described above) in a random order. For each input method, users were presented with all combinations of target widths (1.5, 3.0 and 4.5cm) and target distances (7.5, 15, 22.5, 30cm). Each block pre-

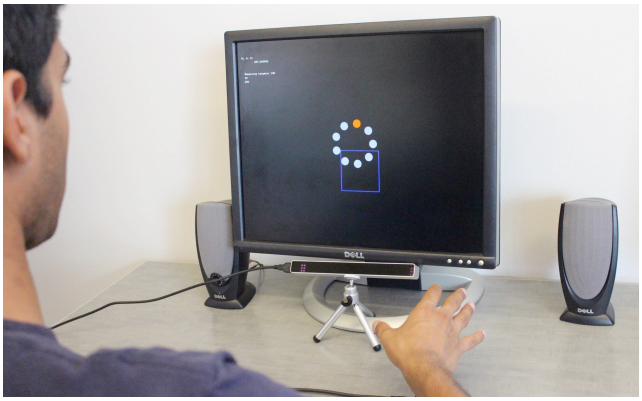

Figure 8. Gaze+gesture pointing technique. The blue square indicates the approximate gaze position.

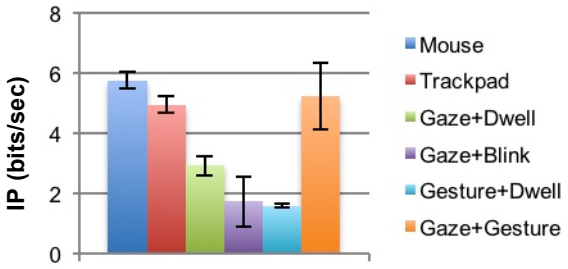

Figure 9. Index of Performance by input method. Error bars represent standard error.

sented nine circular targets in a multidirectional targeting task yielding nine movement times (trials) per block (Figure 8). Target width corresponded to target diameter while target distance corresponded to arrangement diameter. The order of the blocks was randomized within each input method. The target was denoted in red, while all others were grey. If users were unable to select the target within five seconds, the trial was skipped and logged as a time-out.

With each new input method, users were given a brief explanation and were allowed to practice for no more than five minutes. Users were permitted to rest after each block if they wished. The total number of trials generated was 19 participants x 6 input methods x 3 widths x 4 distances x 9 trials = 12,312, including time-outs.

6. RESULTS

We assessed both the Fitts-derived index of performance (IP) as well as the number of timed-out trials to evaluate the scalability of the techniques to small targets.

6.1 Performance

Figure 9 shows the mean IP across the six input methods. Repeated measures analysis of variance showed a significant main effect for difference between the mean of input methods ($F_{5,108} = 9.31$, $p < .0001$). From greatest to least, the mean indices of performance in bits/sec were: mouse, 5.77; gaze+gesture, 5.23; trackpad, 4.95; gaze+dwell, 2.91; gaze+blink, 1.72; and gesture+dwell, 1.57.

We then performed a Tukey HSD test to identify significant differences. The results show that there is no significant performance difference between the gaze+gesture, mouse, and trackpad conditions, which is a positive result given that we included the latter two as gold standards. However, these three methods all significantly outperform gaze+blink and gesture+dwell ($p < 0.05$). Only mouse and trackpad significantly outperform gaze+dwell.

6.2 Scalability to Small Targets

We hypothesized that supplementing gaze targeting with a low-CD-gain gesture suffix would enable users to more readily access smaller targets. To assess this, we look not at IP, but at the percentage of trials with timeouts at different size targets (i.e., the trials in which the user was not able to click the target within five seconds). As expected, the error rate increases as target size decreases (although neither the mouse nor trackpad conditions had any timeouts, and are thus not shown in Figure 10). The timeout rate is comparable across input methods with targets of size 3.0 and 4.5cm. With 1.5cm targets, methods relying on gaze alone (gaze+dwell and gaze+blink) dramatically increase in error. However, those relying on gesture (gesture+dwell and gaze+gesture) have a softer slope, with gaze+gesture performing best.

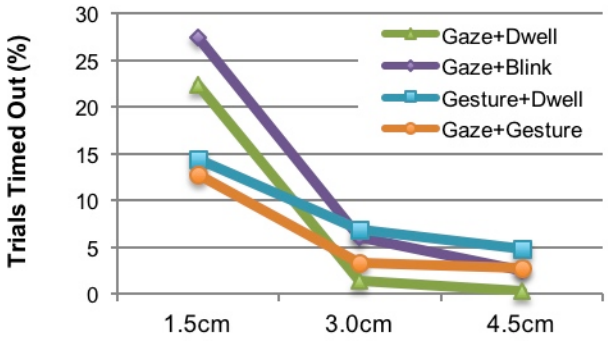

Figure 10. Percentage of timed-out trials by target size.

7. CONCLUSION

We have presented a set of interaction techniques combining gaze and free-space gesture to support common digital activities. We used a taxonomic breakdown to guide our development, organized along two basic interactive dimensions: whether the size of a desired object is above or below the eye tracker's accuracy threshold and whether the subsequent operation involves a discrete hand pose or continuous manipulation. For interactions with objects smaller than the eye tracker threshold, we proposed a synergistic gaze plus fine grain hand manipulation technique. We found that this technique was robust to sensing inaccuracies in commodity hardware, suggesting our approach could be used on low cost, off-the-shelf components. We compare the aforementioned precision pointing method against five baseline techniques; results suggest the technique could enhance systems using gaze or gesture alone.

Today, end-user gaze-augmented systems are rare. However, the increasing prevalence of high-resolution front-facing cameras on laptops, phones and tablets (especially the 2014 introduction of the face-tracking Amazon Fire Phone, which features four, front-facing, high speed infrared cameras) hints that gaze tracking could become ubiquitous on consumer devices in the future. Moreover, researchers continue to make progress on calibration-free eye trackers [32, 46], which would make it possible to create practical walk-up or pick-up and use gesture-augmented systems. Combined with other input modalities, powerful new interaction paradigms are possible. We believe that our example scenarios and interaction techniques underscore this opportunity, pointing the way towards more expressive and efficient free-space interactions.

8. REFERENCES

[1] Bérard, F., Ip, J., Benovoy, M., El-Shimy, D., Blum, J.R. and Cooperstock, J.R. Did "Minority Report" Get It Wrong? Superiority of the Mouse over 3D Input Devices in a 3D Placement Task. In *Proc. INTERACT '09*, 400-414.

[2] Bolt, R.A. Put-That-There: Voice and Gesture at the Graphics Interface. In *Proc. SIGGRAPH '80*, 262-270.

[3] Card, S.K., English, W.K. and Burr, B.J. 1978. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics* 21(8), 601-613.

[4] Chen, L. Incorporating gesture and gaze into multimodal models of human-to-human communication. In *Proc. NAACL-DocConsortium '06*, 211-214.

[5] Chi, C.F. and Lin, C.L. 1997. Speed and accuracy of eye-gaze pointing. *Percept. Mot. Skills*, 85(2), 705-718.

[6] Cockburn. A., Quinn, P., Gutwin, C., Ramos, G. and Looser, J. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *Int. J. Human-Computer Studies*, 69, 401-414.

[7] Connolly, K.J. 1998. Psychobiology of the Hand. Cambridge University Press.

[8] Erol, A., Bebis, G., Nicolescu, M., Boyle, R., and Twombly, X. 2007. Vision-based hand pose estimation: A review. *Comput. Vis. Image Und.*, 108(1), 52-73.

[9] The EyeTribe. Eye Tribe Tracker. `http://theeyetribe.com`

[10] Grauman, K., Betke, M., Lombardi, J., Gips, J. and Bradski, J.R. 2003. Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Univers. Access. Inform. Soc.* 2(4), 359-373.

[11] Hales, J., Rozado, D. and Mardanbegi, D. Interacting with objects in the environment by gaze and hand gestures. In *Proc. PETMEI '13*, 1-9.

[12] Hardenberg, C. and Bérard, F. Bare-Hand Human-Computer Interaction. In *Proc. PUI '01*, 1-8.

[13] Hutchinson, T. E., White, K. P. Jr., Martin, W. N., Reichert, K. C. and Frey, L. A. 1989. Human-computer interaction using eye-gaze input. *IEEE Trans. Syst., Man, Cybern.* 19(6), 1527-1534.

[14] ISO/DIS 9241-9. 1998. Ergonomic Requirements for Office Work with Visual Display Terminals, Non-keyboard Input Device Requirements.

[15] Jacob, R. J. K. What you look at is what you get: eye movement-based interaction techniques. In *Proc. CHI '90*, 11-18.

[16] Jones, B., Sodhi, R., Forsyth, D., Bailey, B., and Maciocci, G. Around device interaction for multiscale navigation. In *Proc. MobileHCI '12*, 83-92.

[17] Kim, H-J., Kim, H., Chae, S., Seo, J. and Han, T-D. AR pen and hand gestures: a new tool for pen drawings. In *CHI EA '13*, 943-948.

[18] Kratz, S., Rohs, M., Guse, D., Müller, J., Bailly, G. and Nischt, M. PalmSpace: continuous around-device gestures vs. multitouch for 3D rotation tasks on mobile devices. In *Proc. AVI '12,* 181-188.

[19] Kumar, M., Paepcke, A. and Winograd, T. EyePoint: Practical Pointing and Selection Using Gaze and Keyboard. In *Proc. CHI '07*, 421-430.

[20] Leap Motion, Inc. `https://www.leapmotion.com`

[21] MacKenzie, C. L., and Iberall, T. 1994. The Grasping Hand. Advances in Psychology, Vol. 104. Elsevier Science B.V. Amsterdam, The Netherlands.

[22] MacKenzie, I. S. 2012. Evaluating eye tracking systems for computer input. Gaze interaction and applications of eye tracking: Advances in assistive technologies, 205-225.

[23] Mateo, J.C., Agustin, J.S. and Hansen, J.P. Gaze Beats Mouse: Hands-free Selection by Combining Gaze and EMG. In *CHI EA '08*, 3039-3044.

[24] Microsoft Corp. Kinect. `http://www.xbox.com/en-US/kinect`

[25] Miniotas, D. Application of Fitts' law to eye gaze interaction. In *CHI EA '00*, 339-340.

[26] Morimoto, C. H. and Mimica, M. R. M. 2005. Eye gaze tracking techniques for interactive applications. *Comput. Vis. Image Underst.* 98(1), 4-24.

[27] Nevalainen, S. and Sajaniemi, J. Comparison of Three Eye Tracking Devices in Psychology of Programming Research. In *Proc. PPIG '04,* 151-158.

[28] Pfeuffer, K., Alexander, J., Chong, M.K. and Gellersen, H. Gaze-touch: Combining Gaze with Multi-touch for Interaction on the Same Surface. In *Proc. UIST '14*.

[29] Pouke, M., Karhu, A., Hickey, S., Arhippainen, L. Gaze tracking and non-touch gesture based interaction method for mobile 3D virtual spaces. In *Proc. OzCHI '12*, 505-512.

[30] Schiffman, H.R. 2001. Sensation and Perception: An Integrated Approach. New York: John Wiley & Sons, Inc. p. 70.

[31] Schwaller, M. and Lalanne, D. Pointing in the Air: Measuring the Effect of Hand Selection Strategies on Performance and Effort. In *SouthCHI '13*, 732-747.

[32] Shih, S. W., Wu, Y. T., & Liu, J. A calibration-free gaze tracking technique. In *Proc. ICPR '00*, 201-204.

[33] Sibert, L.E. and Jacob, R.J. Evaluation of Eye Gaze Interaction. In *Proc. CHI '00*, 281-288.

[34] Slaney, M., Rajan, R., Stolcke, A. and Parthasarathy, P. Gaze-enhanced speech recognition. In *Proc. ICASSP '14,* 3236-3240.

[35] Špakov, O., Isokoski, P. and Majaranta, P. Look and Lean: Accurate Head-Assisted Eye Pointing. In *Proc. ETRA '14*, 35-42.

[36] Starner, T., Auxier, J., Ashbrook, D. and Gandy, M. The Gesture Pendant: A Self-illuminating, Wearable, Infrared Computer Vision System for Home Automation Control and Medical Monitoring. In *Proc. ISWC '00*, 87-94.

[37] Stellmach, S., and Dachselt, R. Look & touch: gaze-supported target acquisition. In *Proc. CHI '12*, 2981-2990.

[38] Stellmach, S. and Dachselt, R. Still looking: investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proc. CHI '13*, 285-294.

[39] Ware, C. and Mikaelian, H.H. An evaluation of an eye tracker as a device for computer input. In *Proc. CHI '87*, 183-188.

[40] Wigdor, D. and Wixon, D. 2011. Brave NUI World: Designing Natural User Interfaces for Touch and Gesture. Morgan Kaufmann, 97-104.

[41] Wilson, A. and Cutrell, E. FlowMouse: a computer vision-based pointing and gesture input device. In *Proc. INTERACT '05*, 565-578.

[42] Wilson, F. 1998. The Hand: How Its Use Shapes the Brain, Language, and Human Culture. Pantheon Books, New York.

[43] Yoo, B., et al. 3D user interface combining gaze and hand gestures for large-scale display. In *CHI EA '10*, 3709-3714.

[44] Zhai, S., Morimoto, C. and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. In *Proc. CHI '99*, 246-253.

[45] Zhang, X., Chen, X., Wang, W., Yang, J., Lantz, V. and Wang, K. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In *Proc. IUI '09*, 401-406.

[46] Zhu, Z., & Ji, Q. 2004. Eye and gaze tracking for interactive graphic display. *Mach. Vis. Appl., 15*(3), 139-148.

[47] Zimmerman, T., Lanier, J., Blanchard, C., Bryson, S. and Harvill, Y. A hand gesture interface device. In *Proc. CHI '87*, 189-192.