HeadCross: Exploring Head-Based Crossing Selection on Head-Mounted Displays

YUKANG YAN*, YINGTIAN SHI*, CHUN YU†, and YUANCHUN SHI, Tsinghua University, China

We propose *HeadCross*, a head-based interaction method to select targets on VR and AR head-mounted displays (HMD). Using *HeadCross*, users control the pointer with head movements and to select a target, users move the pointer into the target and then back across the target boundary. In this way, users can select targets without using their hands, which is helpful when users' hands are occupied by other tasks, e.g., while holding the handrails. However, a major challenge for head-based methods is the false positive problems: unintentional head movements may be incorrectly recognized as *HeadCross* gestures and trigger the selections. To address this issue, we first conduct a user study (Study 1) to observe user behavior while performing *HeadCross* and identify the behavior differences between *HeadCross* and other types of head movements. Based on the results, we discuss design implications, extract useful features, and develop the recognition algorithm for *HeadCross*. To evaluate *HeadCross*, we conduct two user studies. In Study 2, we compared *HeadCross* to the dwell-based selection method, button-press method, and mid-air gesture-based method. Two typical target selection tasks (text entry and menu selection) are tested on both VR and AR interfaces. Results showed that compared to the dwell-based method, *HeadCross* improved the sense of control; and compared to two hand-based methods, *HeadCross* improved the interaction efficiency and reduced fatigue. In Study 3, we compared *HeadCross* to three alternative designs of head-only selection methods. Results show that *HeadCross* was perceived to be significantly faster than the alternatives. We conclude with the discussion on the interaction potential and limitations of *HeadCross*.

CCS Concepts: • **Human-centered computing → Pointing**.

Additional Key Words and Phrases: crossing selection, hands-free selection, head-based interaction

ACM Reference Format:
Yukang Yan, Yingtian Shi, Chun Yu, and Yuanchun Shi. 2020. HeadCross: Exploring Head-Based Crossing Selection on Head-Mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 1, Article 35 (March 2020), 22 pages. https://doi.org/10.1145/3380983

1 INTRODUCTION

Head-mounted displays (HMDs) are increasingly popular as the platforms for VR/AR interaction in recent years. However, as one of the most basic interaction tasks, target selection has not been fully explored on these platforms. Currently, to select a target (e.g., an application icon), users need to reach out their hands in the air and move the hands or the controllers to point to the target, and finally perform a mid-air gesture or press a button to confirm the selection. In this way, users need to raise the hands for an extended time which leads to arm fatigue [15, 53]

*These authors contributed equally to this study
†This is the corresponding author

Authors' address: Yukang Yan, yanyukanglwy@gmail.com; Yingtian Shi, shiyt16@mails.tsinghua.edu.cn; Chun Yu, chunyu@tsinghua.edu.cn; Yuanchun Shi, shiyc@tsinghua.edu.cn, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China.

2474-9567/2020/3-ART35 $15.00
https://doi.org/10.1145/3380983

and limits interaction efficiency. Furthermore, this approach becomes inconvenient or even impractical when users' hands are occupied (e.g., when holding handrails on buses) or for users with limited arm mobility.

To address these issues, we propose *HeadCross*, to select targets on HMDs with head movements instead of hand movements. Commercial HMDs support users to control the pointer with head movements [1, 2] and the users can choose the target by moving the pointer into it. However, to confirm the selection, additional interaction channel is commonly required, e.g., to press a button on the controller or to perform a hand gesture. Performing both selection and confirmation with only head movements has not been well explored or studied. *HeadCross* achieves this goal by combining the selection and confirmation in one step. With *HeadCross*, the users move the pointer to approach the target and perform the *HeadCross* gesture to select the target and confirm the selection at the same time. The *HeadCross* gesture requires users to move the pointer across the target boundary and then turn it back immediately, which is inspired by crossing-based selection methods [8, 21, 25, 33]. This "inside-outside" design aims to reject the false positives of unintentional head movements and speed up the selection process.

In this paper, we first conducted a user study (Study 1) to observe the user behavior of performing *HeadCross* gesture comparing to other head movements. We summarized the design factors and useful features to avoid false positives. Then we implemented the recognition algorithm and an SVM-based binary classifier using the data collected in Study 1. Then we evaluated *HeadCross* through two user studies. In Study 2, we compared it to the dwell-based method and two hand-based methods which are frequently used in VR/AR applications. We chose two basic target selection tasks (text entry and menu selection) as the evaluation tasks and used both VR and AR headsets as the experimental platforms. Results showed that compared to the dwell-based method, *HeadCross* achieved competitive efficiency and improved the sense of control; compared to hand-based methods, *HeadCross* reduced the fatigue. In Study 3, we compared *HeadCross* to another three head-based selection methods to explore the alternative designs of the head gestures for selection confirmation. The alternative designs included two nod-based gestures and the "inside-outside-inside" crossing gesture (IOI) inspired by EyeK [44]. Four selection methods were compared in text entry tasks on both VR and AR headsets. Results showed that *HeadCross* was perceived to be significantly faster than other alternatives while IOI was perceived to be the most accurate method. We conclude with the discussions on the applicability and limitations of *HeadCross*.

The contribution of this paper is three-fold:

- We propose *HeadCross*, a novel mono-modal head-based target acquisition method, which achieves target selection and the confirmation of selection in one gesture.
- We studied user behavior of performing *HeadCross* and extracted useful features for rejecting false positives and guidelines for improving interaction design. We discuss the interaction potential of *HeadCross*, including to leverage the contact position and entering direction as the extra channel to enrich the expressivity of the method.
- We evaluated *HeadCross* and compared it with existing widely-adopted target selection techniques on both AR and VR headsets, including the dwell-based method, hand-based and controller-based methods. We explored the alternative designs of using a head gesture as the selection confirmation and conducted a user study to compare them with *HeadCross*.

2 RELATED WORK

Users commonly interact with HMDs with speech, hand movements (e.g., with controllers), and body movements. Users can issue commands with speech [22, 26], hand gestures [47] and head gestures [64]. To navigate the in virtual environment, users can leverage virtual locomotion techniques (e.g., teleportation[14]), Walk-in-Place methods [55, 56] and joystick methods [12, 19, 20, 46, 56, 58]. For the basic task of target selection, users can perform direct touch [63], indirect pointing with hand [38, 60], gaze [36, 37, 50] or head orientation [28, 61],

and the combined use of them [27, 51, 62]. *HeadCross* aims to enable target selection with mono-modal head movements, especially to address the issues of false positives caused by unintentional movements.

2.1 Head-based Target Acquisition

Target acquisition basically requires target selection and confirmation of the selection [30]. Head movements and gestures have been intensely studied for hands-free target selection tasks, especially for users with limited hand or arm mobility [18, 24, 29, 32, 57]. Previous research (HeadTilt [40], HeadPager [54], HeadTurn [43]) enables users to select options in the menus or check-boxes by performing natural head gestures, including nodding, shaking, head tilting. Besides, head movements are also widely used for controlling and moving the pointer to the target, on the interfaces of desktop GUIs [24, 57], mobile devices [19] and VR and AR headsets [2, 10]. While moving the pointer with head movements, users benefit from the adequate high accuracy and convenience of head movements [28], compared to gaze-based or mid-air hand movements, and the accuracy can be improved by combined methods with both gaze and head orientation [31, 51]. However, another modality is commonly used for the confirmation, including button-press on controllers [28] and performing another hand gesture [2, 30, 39].

Head movements have also been proposed for confirmation of the selection[36, 52]. Pinpointing [30] tests head movements as the primary selection mode or the selection refinement mode (with a higher CD ratio of 2:1). EyeHead [51] leverages head movements for updating the pointer to a new gaze position or triggering a dwell timer. Dwell is frequently applied to achieve head-based confirmation[13], which requires to keep the pointer inside the target for a duration of 450 ms to 1000 ms to avoid false positives [66]. It enables mono-modal target acquisition by achieving both target selection and confirmation with only head movements. Different from this two-step point-and-dwell target acquisition, we propose *HeadCross* which enables one to select a target and confirm the selection with one head gesture. We regard *HeadCross* as a complementary method to achieve head-only target acquisition. It can be useful in the scenarios that users want to look at the potential targets for an extended duration of time without triggering the selection or maintain strong sense of control. More comparison details will be discussed in the following studies (Study 2 and Study 3).

2.2 Crossing-based Target Selection

The design of the *HeadCross* gesture gains inspiration from the crossing-based selection [6]. Different from Point-and-Click selection, crossing-based selection requires users to move the pointer beyond the target boundary to select it. Considering it does not require to stop the pointer over the object or to keep the pointer inside the target while clicking, it can reduce selection time [17] and lower the requirement for the movement control compared to Point-and-Click methods [45].

Crossing-based selection has been successfully applied on desktops [8, 16, 49], touchscreens [33, 34], remote screens [41] and screen-based augmented reality interfaces [65], where an extra modality (e.g., mouse press, touch, pinch gesture) is commonly required to trigger the start of the crossing action. To achieve mono-modal target selection with head movements, we modify the crossing action to be moving the pointer beyond the target boundary and reversely moving it back. The gesture is infrequently performed in users' unintentional head movements and thus can be used to reject false positives. A notable and related selection method is EyeK [49], which enables users to perform text entry with mono-modal gaze movements. To select a target with EyeK, the user first moves the pointer into the key, hover on the key until it extends the size, and then moves the pointer to the outside area and finally to come back to the key area again. Compared to *HeadCross* ("inside-outside"), EyeK requires more steps ("inside-hover-outside-inside") which require more time to perform, and if to select the same target twice in a row, it requires extra pointer movement to the outside area between the selections. The

difference may be that head orientation tracking is more accurate than gaze [28] and thus target selection can be performed in a simpler procedure with *HeadCross*.

3 STUDY1: UNDERSTANDING USER BEHAVIOR

We conduct this study to observe how users perform *HeadCross* and to identify the behavior difference between users performing *HeadCross* and other unintentional head movements. We select looking at the target (the pointer dwelling inside the target) and passing by the target (the pointer going through the target) as the unintentional head movements. We recorded the pointer trajectories to analyze the user behaviors.

3.1 Design

This study had three independent factors, which are *movement type*, *target size*, and *target density*. *Movement type* included *HeadCross*, *Dwell*, and *Pass* and was the main focus of this study. Users were instructed to perform these three types of movements on the specified target. *HeadCross* required the participants to perform the gesture to the target; *Dwell* required to keep the pointer inside the target for 400 ms; *Pass* required to move the pointer into the target boundary and leave it on the other side. We recorded the pointer trajectories of these movements. In addition to these three types of head movements, the participants also moved the pointer from target to target, and sometimes the pointer entered and left targets during this process. We also recorded these movements and labeled them as *Unintentional* passes.

We changed the *target size* and *target density* to observe the behavior changes of *HeadCross* in these different conditions. Target size had five levels measured in angular sizes, which were four to eight degrees evenly, which covered the common target sizes in current VR shooting games [3–5]. Target density had two levels, which were only one target at a time or all the targets together, where the goal will be surrounded by other distracting targets. We set *target density* to be a between-subject factor because as we tested in pilot studies, after users performed tasks in the dense layout, they learned to start *HeadCross* from very near to the target and this had changed their behaviors in the one-target condition.

3.2 Apparatus

We conducted the experiment on HTC Vive Pro [1] and implemented the test application using Unity 2018. The headset tracked head position and orientation at the frequency of 90 fps. As Figure 1 shows, the targets were located at 36 random locations with no occlusion. They were on the same sphere surface with the center at the user's headset and a radius of two meters. This ensured that the targets looked the same size to the users. The pointer was fixed in the center of the view and followed users' head rotations to move on the sphere surface. To help users observe the pointer trajectory, we used *Trail Renderer* in Unity to visualize the recent trajectory of 500 ms.

At the time of Study 1, we did not have a recognition algorithm. So to detect whether the users performed the instructed type of head movements, an experimenter watched and checked the pointer trajectory from another screen. Users press a button on the controller before they performed the head movements and the experimenter started judging the following pointer trajectory. Different tasks were instructed by different colors of the target (*HeadCross* - "Red", *Dwell* - "Blue", *Pass* - "Green"). Accepted trajectories would turn the target to be white again and a new target would appear.

3.3 Participants

We recruited 24 participants from the local campus. Their ages were from 19 to 28 (AVG = 23.58, STD = 1.91). Eleven were female and thirteen were male. Before this experiment, sixteen participants had used VR headsets and

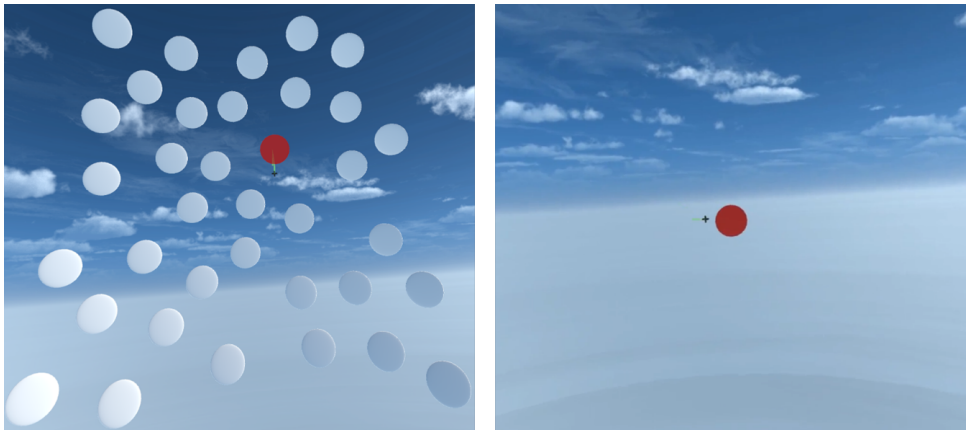

Fig. 1. The experimental interface of Study 1. The left part of the figure visualizes the condition of showing all the target together and the right part visualizes the condition of showing only the goal.

eight had experiences of controlling the pointer with head movements (while using Hololens). All participants had normal vision.

3.4 Procedure

Each user completed 3 task types × 5 target sizes × 36 target positions = 540 trials in this experiment. The trials were divided into five sessions of different target sizes in a randomized order. The order of task types in each session was also randomized. Before the experiment, the warm-up was provided for familiarizing users with different task types. Users took three-minute breaks between sessions.

3.5 Results

We analyzed the difference in user behaviors in different *movement types*. Then we analyzed how *target size* and *target density* influenced the behaviors of performing *HeadCross*. As *target density* was a between-subject factor while *movement type* and *target size* were within-subject factors, we run RM-ANOVA tests for the mixed factorial design for the analysis.

3.6 Behavior Difference

3.6.1 Entering Directions. The entering direction referred to the direction in which the pointer moved into the target. We found observable differences between the distribution of entering directions of *HeadCross* and those of other head movements. As Figure 2 shows, users preferred to perform *HeadCross* in the four basic directions in a very concentrated distribution, among which moving downward was the most frequent direction. As users reported, they felt easy to entering from above because it was similar to nodding the head onto the target. In comparison, for the head movements of other three types, the entering direction distributed more well-proportioned in all directions, and users reported that they did not pay attention to from which direction they entered the target.

3.6.2 Average Speeds. Different *movement types* also resulted in a difference in movement speeds. We measured the head movement speed by the angular speed (degrees per second) of the pointer movement. RM-ANOVA results showed that *movement types* significantly influenced the speed ($F_{2,44}$=137.79, p<0.01). The speed of *HeadCross*

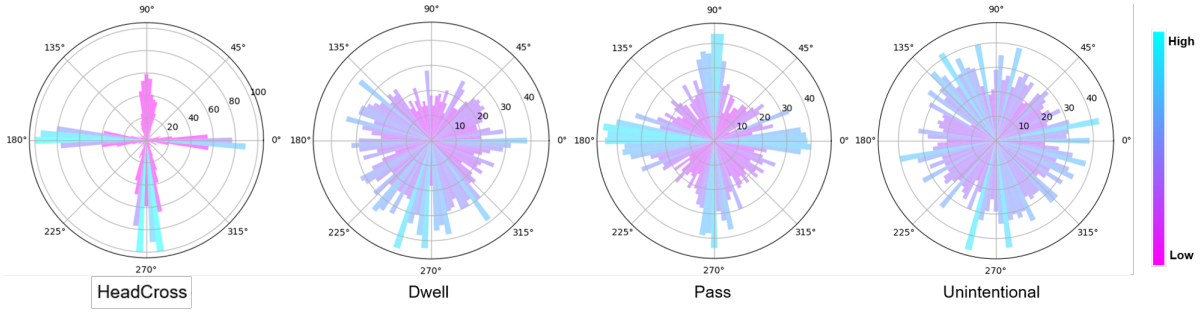

Fig. 2. The distribution of the entering directions of four types of head movements. The bars represent the frequency that the pointer moved into the target in the corresponding directions. The color and the length of the bar visualize the value of the frequency. For example, a light-blue long bar at 270 degrees on the chart of *HeadCross* shows a high frequency of the pointer moving into the target from above the target.

(AVG=18.85, STD=1.25) was between *Pass* (AVG=42.99, STD=2.94) and *Dwell* (AVG=6.14, STD=0.17). This was because *HeadCross* required to turn back inside the target, which consisted of a slow-down and a speed-up process, and it was slower than moving through the target (*Pass*) while faster than a dwelling for 400 ms inside the target (*Dwell*).

3.6.3 Speed and Direction Change Patterns. In addition to average movement speed, we analyzed the speed changing pattern of different *movement types* inside the target boundary. As the left part of Figure 3 shows in *HeadCross*, the highest speed appeared at the entering point on the target boundary (light green on two sides) and dropped at the turning point inside the target (dark blue in the middle). This pattern was distinguishable from *Dwell* which was always at low speeds, but similar to *Pass* and *Unintentional* passes. We also analyzed the movement direction changing pattern inside the target. We measured it by calculating the included angle from the pointer position to the entering point and the leaving point. If there were no direction changes, the included angle would be 180 degrees all the time and if there was a sharp turn inside the target, the included angles would be much smaller. The right part of Figure 3 visualizes the patterns of the included angles. As *HeadCross* required a sharp turn inside the target, the included angle of each point on the trajectory was small (dark blue). In comparison, *Pass* required the pointer to go through the target to the other side, the included angles were much larger, which was similar for *Unintentional* passes.

3.7 Target Size

3.7.1 Average Speed. RM-ANOVA results showed that a larger target size significantly increased the movement speed of *HeadCross* ($F_{4,88}$=50.50, p<0.01). The average speeds for five levels of target size were 16.41, 17.98, 19.05, 20.29 and 20.50 degrees per second. Except for the target size of 7 and 8 degrees, speeds of each two adjacent levels were significantly different (all p<0.01). In summary, the users moved the pointer faster as the target became larger until it was larger than 7 degrees. When the target is even larger, the size did not affect the average speed. The reason may be that users would turn the pointer far from the other side of the target, so they did not care about the actual size of the target.

3.7.2 Path Length Inside the Target. We calculated the path length of the pointer trajectory inside the target and ran RM-ANOVA tests on different target size conditions. Results showed that larger target size significantly enlarged the length ($F_{4,88}$=78.71, p<0.01), with averages of 2.80, 3.14, 3.37, 3.58 and 3.93 degrees for different target sizes. Post-hoc tests showed that differences between every two levels were significant. Users reported that when

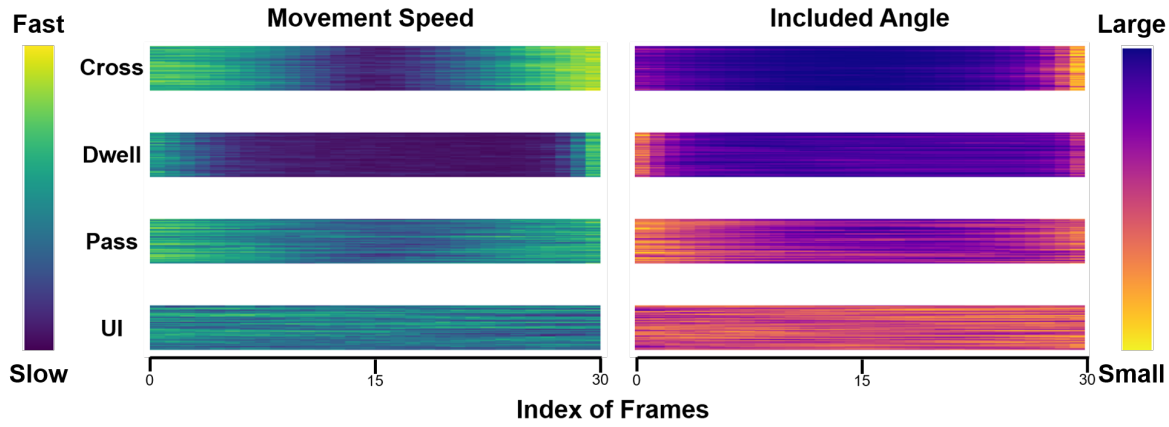

Fig. 3. The changing patterns of the pointer movement speed and moving directions with different types of head movements. For demonstration, we randomly selected 300 pointer trajectories for four types of head movements and segmented the trajectory parts that were inside the target. As they were of different lengths, we evenly sampled 30 frames out of each trajectory. We mapped the values of the movement speed and the included angle of each frame to the colors according to the color scale on the left and right bars. The indexes of the frames represent the relative rank of the frames in time and are shown along the x-axis.

the target was larger, they felt it safer to move the pointer further inside the target; when targets were small, they turned earlier to avoid overshoots. Related to the path length, the distance between entering point and exit point (*d* in Figure 4) was also significantly enlarged when the target size was larger ($F_{4,88}$=14.15, p<0.01).

3.7.3 Overshoots. RM-ANOVA results showed that the frequency (number) of overshoots was significantly affected by target size ($F_{4,88}$=5.43, p=0.001). When the target size was smaller, users were more possible to overshoot. As post-test results showed, overshoots with the target size of 4 and 5 degrees were significantly more than that of 6, 7, 8 degrees. When the target was larger than 6 degrees, the overshoot possibility dropped from 2.4 % to 1.0 % on average. So a target larger than 6 degrees may be required to avoid overshoots.

3.8 Target Density

We measured the influence of *target density* on how long and how far *HeadCross* action was started before entering the target boundaries. RM-ANOVA tests showed that the starting distance ("how far") was significantly affected by target density ($F_{1,22}$=5.84, p=0.024), while the number of frames ("how long") was not affected ($F_{1,22}$=1.51, p=0.23). So when all targets were shown, users performed *HeadCross* significantly nearer to the goal, maybe to avoid touching other targets. Symmetrically, we calculated the number of frames and the movement distance of the trajectory after leaving the target. Test results were similar: movement distance was significantly shortened when all targets were shown ($F_{1,22}$=5.27, p=0.032) while the number of frames was not affected ($F_{1,22}$=0.0, p=0.99). As users reported, in some cases, they purposely avoided entering another target after a *HeadCross*, which shortened the ending distance.

4 INSPIRATIONS FOR INTERACTION RECOGNITION AND DESIGN

In this section, we summarize the inspirations drawn from Study 1 for the recognition and design of *HeadCross* interaction. For *HeadCross* recognition, we discuss useful spatial and temporal features about the pointer trajectory; for the interaction design, we discuss the guidelines for the design of visual feedback and target appearance.

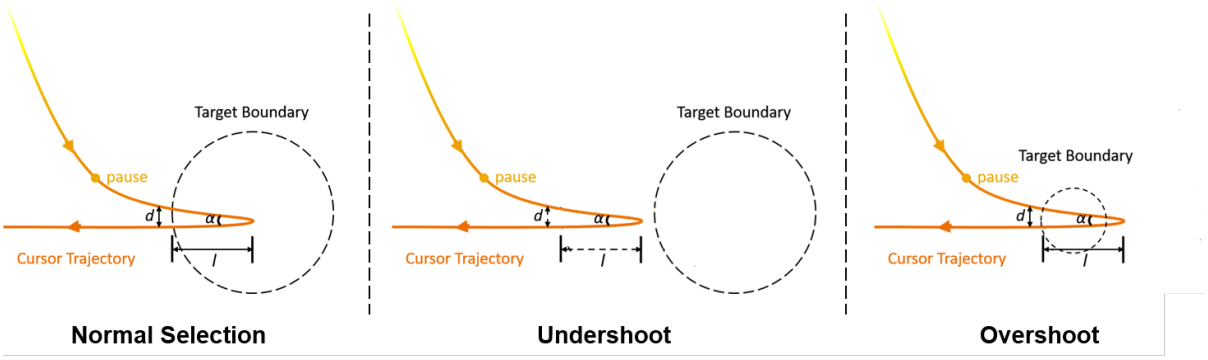

Fig. 4. The left part shows the process of a normal selection using *HeadCross*: the pointer slows down while approaching the target; then it speeds up and moves across the target boundary and crosses back. The middle part shows the undershoot condition where the pointer turns too early before entering the target. The right part shows the overshoot condition where the pointer moves too far and goes beyond the target boundary on the other side before crossing back. In all three cases, the pointer trajectories contained a sharp turn, which can be characterized by the features shown as d, l, and $\alpha(which we estimated in undershoot conditions)$.

4.1 Spatial Feature

As Figure 4 shows, a *HeadCross* gesture is to move the cursor across the target boundary and then cross back. We extract spatial features that characterize the shape of the pointer trajectory during this process. As Study 1 suggested, the distribution of the entering direction of *HeadCross* is different from other head movements and a sharp turn (small included angle) inside the target is an important feature to distinguish it from other movements. So we summarize spatial features as the entering direction, the entering and leaving positions, the length of the trajectory inside the target, the sharpness of the turn. Take the left part of Figure 4 as an example: the crossing direction is from the left, the distance between the entering and leaving positions is d and the length of the trajectory inside the target is l.

4.2 Temporal Feature

As Study 1 suggested, there is a difference in the changing patterns of the movement speed between *HeadCross* and other head movements. So we extract temporal features to characterize the speed changes. Based on the observation, a typical speed changing pattern of the pointer while users are performing *HeadCross* is as follows: the pointer slows down when it gets close to the target, then it speeds up to go across the boundary, slows down again before turning back and speeds up in the opposite direction and finally stops after leaving the target. In this process, we can expect two low-velocity regions outside the target (the start and end of *HeadCross*), one low-velocity region inside the target (the turn), between which are two speed-up and slow-down processes.

4.3 Visual Feedback

In Study 1, users reported that sometimes they did not know whether they performed the *HeadCross* gesture correctly, e.g., whether it was an overshoot. So we propose to provide visual feedback of the recent trajectory of the pointer, which is helpful to control the pointer movement and get aware of the recent movements. Considering the average duration of performing the *HeadCross* gesture, we propose to show the recent 500 ms of the pointer trajectory, which was tested to be long enough for users to observe the entire process of the gesture and also not too long to be distracting.

4.4 Target Appearance

In Study 1, we observed how participants performed the *HeadCross* gesture to targets in the circle shape. In this condition, participants more frequently crossed the target from the four basic directions than other directions. We draw two inspirations from the results: 1. we could change the target appearance to guide the participants to cross the target in comfortable directions. For example, an edge on the top will offer the affordance to cross the target from above. 2. we could map different crossing directions or positions to triggering different functions. For example, crossing a menu from four basic directions can be used to select different options on the menu. Besides, as Study 1 showed, participants rated a target size of no smaller than 7 degrees to be safe and easy to perform *HeadCross* for target selection.

5 IMPLEMENTATION

This section introduces the implementation of the *HeadCross* recognition algorithm, which consists of two steps: First, we leverage the speed changing pattern of *HeadCross* and the crossing events at the targets to filter out the potential *HeadCross* pointer trajectories; Then we extract spatial and temporal features of the pointer trajectories and classify whether they are *HeadCross* gestures or unintentional head movements by an SVM-based binary classifier.

5.1 Movement Filter

The first step of gesture recognition is to filter out the potential *HeadCross* pointer trajectories. As a *HeadCross* gesture requires the pointer to move into a target and cross back, we use the entering point and the leaving point to segment the trajectory. We predict the start point and the end point outside the target by linear regressions. We specially deal with the trajectories that may contain overshoots or undershoots while users are performing *HeadCross* gestures.

5.1.1 Trajectory Segmentation. As discussed in Section 4.2, the low-velocity regions around the start and end of a *HeadCross* trajectory would be helpful for gesture recognition of *HeadCross*. So we need to segment the pointer trajectory that contained the whole process of the gesture. As we can detect the entering point and the leaving point on the target boundary, we need to leverage the trajectory part inside the target to estimate the start point and the end point of the gesture. We intuitively hypothesize that the time duration of the starting and ending period would be linear to the time duration inside the target, considering a larger *HeadCross* movement (to a larger target) leads to a longer duration for movement in all three periods. Based on the results of Study 1, the estimated linear regression formulas are as follows:

$$Starting\_Period = 0.2 \times Inside\_Period + 0.15\ seconds \tag{1}$$

$$Ending\_Period = 0.25 \times Inside\_Period + 0.05\ seconds \tag{2}$$

5.1.2 Overshoots and Undershoots. Overshoots and undershoots will appear when users move the pointer across the whole target before turning it back or turn the pointer too early even before entering the target boundary. We specifically deal with these two types of trajectories to improve recognition performance. For overshoots, we set adaptive distance thresholds to targets of different sizes, which allow the pointer to move over the boundary but returns within the distance range. Based on the overshoot results in Study 1, we set the thresholds to be 1 degree for the target smaller than 6 degrees, and 0.5 degrees for the others. As simulated, the overshoot rate was reduced to less than 1% for the *HeadCross* data we collected in Study1. For undershoots, we can not segment the trajectory by the entering and leaving point on the target, so we directly use the speed changing pattern to filter out the potential undershoots. We apply a sliding window of 700 ms to detect three low-velocity regions and two

Table 1. The spatial and temporal features used to classifiy the pointer trajectories.

Spatial	**Explanations for the Features**
enter_exit_offset	Offset between the entering and the leaving points
start_end_offset	Offset between the start point and the end point of the gesture
turning_angle	The sharpest turn (in degrees) inside the target
curvature	The highest local curvature of the trajectory inside the target
line_shape	The standard deviation the linear regression results of the trajectory segments that are before and after the turning point (the furthest point from the entering point)
distance_inside	The furthest distance that the pointer has been from the entering point inside the target
index_difference	The index difference of the turning point and the furthest point.
Temporal	**Explanations for the Features**
duration_inside	Duration of time that the pointer stays inside the target
velocity_min	Local minimums of the pointer speeds before, during and after being inside the target
velocity_max	Local maximums of the pointer speeds in the whole process

speed-up and slow-down periods between them as the potential undershoots of *HeadCross* gestures. We also set a distance threshold of 0.5 degrees to ensure the potential *HeadCross* trajectories to be near to a target to select.

5.2 Trajectory Classification

For the pointer trajectories that have been filtered out to be potential *HeadCross* gestures, we extract spatial and temporal features from the trajectories and leverage an SVM-based binary classifier to judge whether it is qualified to be a *HeadCross* gesture.

5.2.1 Features. Guided by the inspirations drawn from Study 1, we extract spatial and temporal features from the potential *HeadCross* trajectories. Spatial features characterize the trajectory shape which should have a symmetrical "Inside-Outside" path, a sharp turn inside the target and two straight lines around the turn. Temporal features characterize the changing pattern of the pointer movement speed. To avoid the noises, we first smooth The pointer movement speeds by mean and median filter [9] before extracting temporal features. The detailed explanation of each feature is listed in Table 1. To be noted, for the overshoots within the distance threshold, we add the overshoot part of the trajectory to the calculation of *distance_inside*; for the undershoots where we can detect two high-velocity regions, we use the pointer movement distance between these two points to replace the *distance_inside* in the calculation.

5.2.2 Classifier. With the pointer trajectory data that we collected in Study1, we trained an SVM-based binary classifier. For each trajectory, we computed the features to form the input vector for the classifier. The classifier was implemented with the *scikit-learn* python library. We used 4320 (24 users × 180 trials) instances of *HeadCross* trajectories as the positive data, and data of the other two tasks and 5000 instances of unintentional movement data as the negative data. Five-fold cross-validation showed that the average recognition accuracy was 95.81% (STD=2.12%).

6 STUDY2: PERFORMANCE EVALUATION

In this study, we evaluated the user performance and experience of selecting targets with *HeadCross*. To achieve this goal, we compared *HeadCross* to three baseline methods which are frequently used on commercial VR and

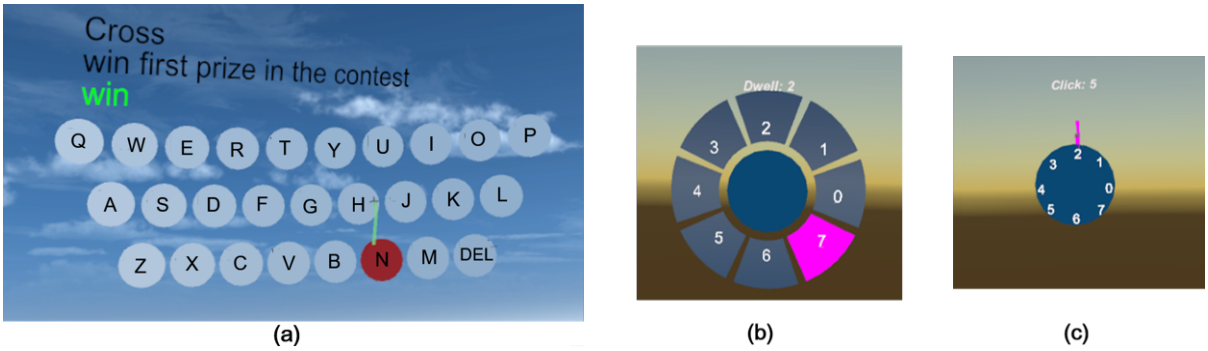

Fig. 5. The experimental interfaces of Study 2: (a) *Keyboard* task, which shows the current selection method "Cross", the target sentence and the keyboard; (b) The Marking Menu interface for *Menu* task, which shows the current selection method "Dwell" and the index of target option "2"; (c) The menu interface for *HeadCross* for *Menu* task.

AR headsets. The baseline techniques include a dwell-based head-only selection method, a hand-based mid-air gesture method, and a controller-based method. We compared them in two frequently-performed target selection tasks (text entry and menu selection) on both VR and AR headsets. To measure the user performance, we recorded the task completion time, the number of selection errors; to analyze the user experience, we collected users' scores for their subjective feelings in different aspects.

6.1 Selection Methods

We compared *HeadCross* to three baseline selection methods, which are all frequently-used on current VR and AR headsets. Mid-air hand gestures are commonly-used on AR headsets (e.g., Hololens[2]) to confirm the selections. We selected the default "Air Tap" gesture (*hand* for short) which is applied on Hololens as an AR baseline method. It was performed as "*Raise the hand in the air, make a loose fist, point up the index finger, quickly tap it down and all the way back up*". Controller-based selections (*controller* for short) are common on VR headsets using Outside-in tracking. We chose it as a VR baseline method, which required participants to press a button on the controller as the confirmation. We also tested another head-only method, which was a dwell-based selection method (*dwell* for short). It required participants to keep the pointer inside the target for a time duration beyond the threshold. This method was applied to both AR and VR applications [66]. To be noted, the input efficiency of this method was influenced by the time threshold, which was typically from 400 ms to 1000 ms[66], and it was a trade-off that shorter thresholds would speed up the input while longer thresholds would reduce more false positives. We applied 400 ms as the time threshold in this study. *HeadCross*, *hand* and *dwell* all moved the pointer with head movements and leveraged different method to confirm the selection. *Controller* first pointed to the target with the controller and then press the button to confirm the selection.

6.2 Task Design

We tested *Keyboard* task and *Menu* task to simulate the frequently-performed target selection tasks on VR and AR headsets. *Keyboard* task required participants to select keys on a keyboard to form a sentence, which simulated the text entry tasks. Considering the keys were densely located on the keyboard, during *Keyboard* task, participants needed to frequently move the pointer through different keys, which led to the high possibility of triggering false positive selections. Figure 5 (a) shows the interface of *Keyboard* task. The interface showed the required selection method, the sentence to input and the inputted characters above the keyboard. A successful selection on the keys turned them into red color for 50 ms so the participants can see the visual feedback right after they performed the selections. Correctly inputted characters were shown in green color and the wrong characters were shown in red,

which reminded users to delete it and input the correct character again. Spaces were automatically added after a complete word was inputted. The size of the key was six degrees in angle, which was tested to be appropriate for target selection tasks in Study 1. For *HeadCross*, we did not instruct the participants to avoid other targets while starting to perform the *HeadCross* gesture.

We also tested *Menu* task, which required participants to select an option out of the menu. For three baseline methods, we implemented the *Marking Menu* [11, 48, 59], which is widely-applied for menu selection tasks. As Figure 5 (b) shows, the options were shown after the menu was triggered and were located around the menu button. For *HeadCross*, we used a different menu interface. As Figure 5 (c) shows, we only maintained the menu button without the options around it. We tested to select different options by performing *HeadCross* to the same menu button but at different contact points. We used this task to test the potential of *HeadCross* to increase the interaction bandwidth, as it inputted more than one bits of information in one gesture. *Menu* tasks included one-option selection tasks and two-option selection tasks which required participants to select two options on the same menu in a row. This was a simulation for the selection task in multi-layer menus.

Participants performed 2 platforms × 2 selection tasks × 3 methods = 12 sessions in this experiment. *HeadCross* and *dwell* were tested on both headsets, *hand* was only tested on the AR headset and *controller* was only tested on the VR headset. The order of platforms and task types were counter-balanced across participants and the order of the techniques was randomized. For *Keyboard* task, the users inputted ten sentences in each session, randomly chosen from a widely-used phrase set [35]. For *Menu* task, user completed 3 rounds × 8 options = 24 one-option selections and 3 rounds × 8 randomly selected two-option pairs = 24 two-option selections.

6.3 Apparatus

We chose two state-of-the-art AR and VR headsets as the experimental platforms, Hololens [2] and Vive Pro [1]. Hololens had a field of view of about 30° × 17.5°, the sensing accuracy was about 2° in rotation and tracking frequency was 60 fps. Vive Pro had a field of view of 110 degrees, the tracking precision was reported to be less than 1mm and the tracking frequency was 90 fps.

6.4 Participants

We recruited sixteen participants from a local campus. Their average age was 23.56 (STD = 1.99). Six were female and ten were male. They scored for the familiarity to VR and AR interaction in five-point Likert scale [7] (from "1 - Poor" to "5 - Excellent"). The average score for VR familiarity was 3.63 (STD=1.02) and 2.81 (STD=0.98) for AR. All participants had normal vision.

6.5 Procedure

The experimenter first introduced two headsets, *Keyboard* task and Menu Task, and four selection methods to the participants. Then participants familiarized themselves with them in ten minutes and they were free to try the tasks with different selection methods during this time. When they reported being ready, they performed 12 sessions of tasks with breaks between every three sessions. The whole experiment took about 90 minutes. After the experiment, participants filled in questionnaires to score for perceived speed, accuracy, and fatigue in each session and we also collected their comments on the selection methods.

6.6 Analysis

We measured the interaction efficiency by the task completion time and the selection accuracy by the frequency of selection errors. For two-option selections in *Menu* task, selecting the wrong option in either step was recorded as selection errors. As only *HeadCross* and *dwell* were tested on both platforms, we conducted 2 × 2 RM-ANOVA tests to compare their interaction efficiency and selection accuracy. To compare the performance of *HeadCross*

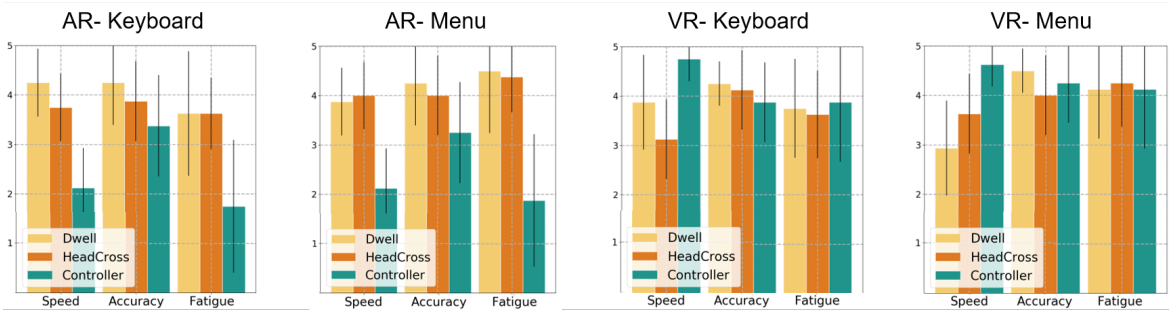

Fig. 6. Subjective scores from the participants on the perceived speed, accuracy, and fatigue while using different selection methods in Keyboard and *Menu* tasks. The error bars represent the standard deviations.

with *hand* and *controller* respectively in AR and VR conditions, we conducted paired-samples t tests. For subjective scores, we run Wilcoxon signed-rank tests between different conditions.

6.7 Keyboard Task Results

6.7.1 Speed. RM-ANOVA results showed that both *platforms* ($F_{1,15}$=5.38, p=0.035) and *selection method* ($F_{1,15}$=19.70, p<0.001) were significant factors on target selection speed. The AR headset (AVG=1.27s, STD=0.04s) required significantly more time to select a target than the VR headset (AVG=1.20s, STD=0.06s). The smaller FOV size of the AR headset cannot show the whole interface at all time, thus participants took more time to navigate the keyboard and switch between inputting and checking the inputted characters. *HeadCross* (AVG=1.36s, STD=0.07s) was significantly slower than *dwell* (AVG=1.11s, STD=0.03s). Considering *dwell* was commonly implemented with the time threshold from 450 ms to 1000 ms, we can expect *HeadCross* to be competitive to the *dwell* with the threshold of 650 ms based on the speed results.

Paired-samples t test showed that *HeadCross* (AVG=1.40s, STD=0.07s) was significantly faster than *hand* (AVG=1.72s, STD=0.10s) on AR headsets (t_15=-4.67, p<0.001). For *hand*, the switch from moving the pointer with head movements to performing the gesture with hand took more time than head-only selections of *HeadCross*. However, on VR headsets, *controller* (AVG=0.93s, STD=0.05s) was significantly faster than *HeadCross* (AVG=1.33s, STD=0.09s; t_15=5.57, p<0.001).

6.7.2 Accuracy. RM-ANOVA results showed that neither *platform* ($F_{1,15}$=3.32, p=0.09) or *technique* ($F_{1,15}$=0.89, p=0.36) significantly affected the selection accuracy. Only a few mistakes were made in these conditions (<1%). Paired-samples t tests showed that *HeadCross* (AVG = 0.76%, STD = 0.27%) made significantly fewer mistakes than *controller* (AVG=1.59%, STD=0.35%; t_15=-3.21, p=0.006). As participants reported, pressing the trigger sometimes moved the pointer away and might causes selection errors. *Controller* applied the default raycasting to point to targets, and this mechanism magnified the jitters of hand to the movements of the pointer which limited control accuracy.

6.7.3 Subjective Feeling. Figure 6 (left) summarizes the subjective scores for *Keyboard* task. Tests showed that perceived fatigue of *hand* was significantly heavier than *HeadCross* (Z=-2.89, p=0.04) and *dwell* (Z=-2.04, p=0.041). Users commented that raising the arm in the air was tiring, and became increasingly severe through the experiment. Besides, participants commented that *HeadCross* provided a strong sense of control. For example, P5 reported *"Using* dwell*, I needed to wait until the target turned color. Especially for several tasks at the beginning of the experiment, I did not know how long it would take (to trigger the selection). Using* HeadCross*, the sense of control was stronger because the gesture was 'a click performed by the head'."* Besides, as we observed, using *dwell*,

participants frequently moved the pointer out of the keyboard to avoid selecting the wrong keys by accident. So although little false positives happened for both *dwell* and *HeadCross*, there was still a difference in the user behaviors and user subjective feelings which were reflected in their comments. *HeadCross* enabled users to care less about the false positive problems and to move the pointer with fewer restrictions.

6.8 Menu Task Results

As *Menu* task had one-option and two-option selections, we run 2 methods (*HeadCross* and *dwell*) × 2 platforms × 2 levels RM-ANOVA tests on selection efficiency and accuracy.

6.8.1 Speed. Results showed that different from *Keyboard* task, *HeadCross* (AVG=1.48s, STD=0.06s) was significantly faster than *dwell* (AVG=1.73s, STD=0.05s). Both *method* ($F_{1,15}$=60.15, p<0.001) and *level* ($F_{1,15}$=19.64, p<0.001) were significant factors for speed. *HeadCross* improved the input speed by completing the activating the menu and picking the target option in one action. In comparison, *dwell* and *hand* required to trigger the menu button first and then choose the target option. A significant interaction effect between *method* and *platform* ($F_{1,15}$=12.12, p=0.003) was found and *dwell* had a larger speed decline (delta=0.40s) than *HeadCross* (delta=0.22s) changing from VR to AR headsets. *HeadCross* on AR headsets was significantly faster than *hand* for both one-option and two-option selections (t=-15.52, -20.52; p<0.001), while was significantly slower than *controller* on two-option selections on VR headsets (t=3.34, p=0.004).

6.8.2 Accuracy. Tests showed that *platform*, *method* and *level* all significantly affected the selection accuracy ($F_{1,15}$=6.22, 15.84, 8.59; p=0.025, 0.001, 0.010). Different from *Keyboard* task, *dwell* (AVG=1.9%) was more accurate than *HeadCross* (AVG=4.6%). This showed the trade-off that in this task, *HeadCross* increased input bandwidth by levering the contact points for selecting different options and improved selection speed; however, as the cost, it required more precise movement control and reduced accuracy. Using other features (e.g., crossing directions) and reducing the number of options (e.g. only using four basic directions) may improve the accuracy. Two-option selections had lower accuracy (AVG=4.3%) than one-option (AVG=2.3%) and AR platform (AVG=3.9%) made more errors than VR platform (AVG=2.7%).

6.8.3 Subjective Feeling. The perceived speed of *HeadCross* and *dwell* were tested to have no significant difference (t=0.48, p=0.63). Similar to *Keyboard* task, *hand* resulted in significantly heavier perceived fatigue than the other method (all t<-3, p<=0.001). Participants commented that *HeadCross* performed better in this task compared to *Keyboard* task, and the design of selecting different options by different contact positions was straightforward. However, P7 expressed his concern that when selecting two options on opposite sides of the menu, using *HeadCross* sometimes caused false positive problems. P5 also commented that four options around the menu would be the best for *HeadCross* as crossing the target in four basic directions would be easier to control than contact positions.

6.9 Discussion

In this experiment, we evaluated the target selection performance and user experience of *HeadCross*. For selecting targets in a dense layout (*Keyboard* task), *HeadCross* showed competitive input speed (roughly as *dwell* method with time threshold of 650 ms) and caused few false positives; while for selecting menu options, *HeadCross* picked the options by crossing the menu button at different contact positions and outperformed *dwell* and *hand* in speed, In both tasks, participants reported positively about using *HeadCross* and they frequently mentioned that the improvement in the sense of control was important for selection tasks.

We tested the controller-based method on Vive Pro, which used the Outside-in tracking and the hand-based method on Hololens, which used the Inside-out tracking. The difference was that Hololens required participants to perform the mid-air gesture (AirTap) inside the interaction area that the cameras can sense. As a result,

participants needed to keep their arms in the air for an extended time, which caused issues with arm fatigue. Similar results can be expected if we test hand-based selection methods on VR headsets using Inside-out tracking. We will test this condition in the future.

7 STUDY3: HEAD-BASED SELECTION METHODS

In this study, we explored three other alternative designs of head-based selection methods: "Point-and-Nod" (PN), "Point-Dwell-Nod" (PDN), and "Inside-Outside-Inside" (IOI) and compared them with *HeadCross*. Different from the *dwell* method, the methods that we tested in this study leveraged a head movement to confirm the selection. The comparison focused on the design of the confirmation movement, which influenced the interaction efficiency and user experience. We invited sixteen new participants to perform *Keyboard* task in Study 2 using these four selection methods. We recorded the task completion time, the number of selection errors and users' subjective scores on the experience.

7.1 Selection Methods

In this study, we compared *HeadCross* to another three head movement-based target selection methods, all of which enabled users to select targets on VR/AR headsets in a hands-free manner. The explanation of the techniques are as follows:

- *Point-Nod (PN)*: the user moves the pointer into the target (Point) and then performs a nodding gesture with head movements (Nod) to confirm the selection.
- *Point-Dwell-Nod (PDN)*: the user moves the pointer into the target and dwells for 200 ms, then the target is highlighted and the user performs a nodding gesture to confirm the selection.
- *In-Out-In (IOI)*: the user moves the pointer into the target, then the user moves the pointer out of the target boundary and back to the target again as the selection.
- *HeadCross*: the user moves the pointer into the target and turns it back as the selection.

We compared these methods to explore alternative designs of head gestures to trigger the selection confirmation. We chose PN as one baseline method as it applied the two-step selection: selecting the target by moving the pointer into it and the nodding gesture served as the confirmation. The difference between *HeadCross* and PN is that the nodding gesture in PN did not need to move across and back the target boundary, which was easier to perform and did not require precise movement control. However, as the trade-off, PN may have higher risks of false positive problems. As we tested in our pilot study, frequent false positives appeared while we were using PN and they made it difficult to select the targets efficiently. So we implemented PDN which added a dwell of the pointer inside the target before the nodding gesture. We set the time threshold of this dwell to be 200 ms, which was much shorter than *dwell* method tested in Study 2, as the nodding gesture would be performed as the final confirmation following the dwell. Inspired by EyeK [44], we implemented IOI which had a different design for crossing the target boundary. IOI required the pointer to first enter the target and then to cross back outside and to enter the target again. Compared to *HeadCross*, IOI applied the three-step "Inside-Outside-Inside" selection which would be even stronger at rejecting false positives but as the trade-off, it may slow down the selection speed.

7.2 Design

This was a two-factor within-subject experiment. The main independent factor was the target *selection methods* (HeadCross, PN, PDN, IOI) and the second factor was *platform* (VR/AR). The task was the same as *Keyboard*

task in Study 2. We recorded the task completion time to evaluate the input efficiency, the number of errors to evaluate the input accuracy and subjective scores of participants to evaluate the user experience. Participants rated for the experience in the aspects of fatigue, mental effort, learning effort, perceived speed, and perceived accuracy. We also asked participants to rank the four methods on the overall experience. Besides, we recorded the pointer movement trajectories in each task, from which we revealed insights of user behavior differences while selecting targets with different techniques.

7.3 Participants

We invited 16 participants (11 males, 5 females, age = 21.75 ± 3.68). Twelve of them had VR/AR experience before this experiment. None of them had participated in Study 2.

7.4 Results

In the sessions of PN, participants met frequent false positives (35% on average) and all participants dropped the session after trying to input 1.63 sentences in VR and 1.38 sentences in AR on average. In total, we collected 16 participant × 2 platforms × 3 selection methods × 10 sentences + 48 sentences of PN = 1008 sentences. We averaged the time to select characters of each sentence and then averaged the selection time of the sentences in each session. We ran RM-ANOVA to test the difference of *selection method* and *platform* on selection speed and accuracy. We ran Wilcoxon signed-rank test to compare the subjective scores from users. Besides, we discuss the user behavior difference referring to typical pointer trajectories that we collected from participants.

7.5 Speed

RM-ANOVA tests showed that *selection method* ($F_{3,45}$=79.07, $p < 0.01$) and *platform* ($F_{1,15}$=30.42, $p < 0.01$) were both significant factors for task completion time. Paired sample t-tests showed that *HeadCross*, PDN, and IOI were significantly faster than PN (all $p < 0.01$) and no significant difference was found between the other three selection methods. Among them *HeadCross* had the highest averaged speed (AVG = 1.37s, STD = 0.06s). The averaged speeds for four selection methods are shown in the left part of Figure 7. For *platform*, participants completed tasks significantly faster on VR headset than AR headset (AVG = 1.70s, 2.48s). As reported by participants, it was similar to Study 2 that the smaller field of view of AR required more frequent switches between selecting keys and checking the inputted characters.

7.6 Accuracy

RM-ANOVA tests showed that *selection method* ($F_{3,45}$=333.60, $p< 0.01$) and *platform* ($F_{1,15}$=12.97, $p < 0.01$) were both significant factors for task completion time. Paired sample t-tests showed that there were significant difference in selection accuracy between IOI and *HeadCross* ($p < 0.01$), *HeadCross* and PDN ($p = 0.05$), and PDN and PN ($p < 0.01$). For *platform*, participants completed tasks significantly more accurately on VR headset than AR headset.

7.7 User Experience

The right part of Figure 7 shows the averaged subjective scores. Wilcoxon signed-rank tests showed that participants perceived *HeadCross*, PDN and IOI to be significantly better than PN in all five aspects (all $p< 0.05$). In addition, participants perceived *HeadCross* to be significantly faster than PDN ($Z = -1.99$, $p < 0.05$) and IOI (Z

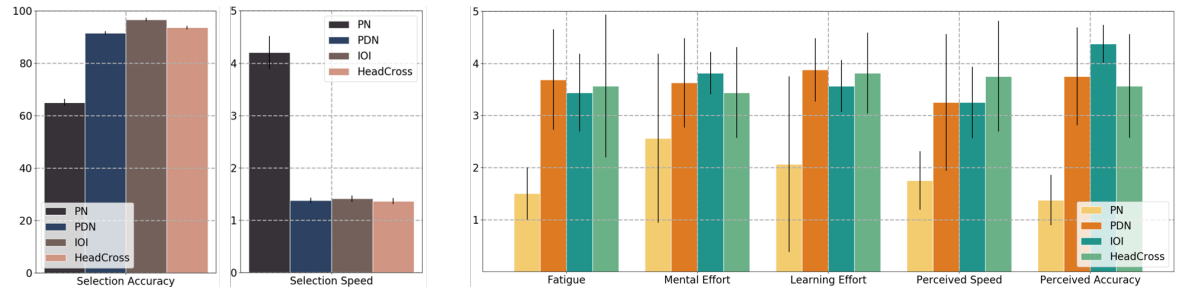

Fig. 7. Left: The averaged selection accuracy and speed of four selection methods. Right: The averaged subjective scores from the participants on the aspects of fatigue, mental effort, learning effort, perceived speed and perceived accuracy of using the selection methods. The error bars represent the standard deviations.

= -2.08, $p < 0.05$) and IOI to be significantly more accurate than *HeadCross* ($Z = -2.75$, $p < 0.01$) and PDN ($Z = -2.33$, $p < 0.05$). There were no significant differences in the other three aspects between these three selection methods. The perceived differences in selection speed and accuracy were consistent with the actual averaged speed and accuracy. For the overall preference result, the averaged rank of *HeadCross*, PDN, and IOI were the same as 1.875th and PN was ranked as the least preferred method (4th for 15 participants and 3rd for P6). Most participants thought PN to be unusable in this task. P5 thought it could be usable and fast to select among sparse targets. P4 commented that IOI was like a "double-click" with head movements, it was safer for the selection tasks, however, they felt it tedious to continuously perform IOI. P8 commented that he felt more comfortable to perform *HeadCross* vertically instead of horizontally.

7.8 User Behavior

In Figure 8, we visualized four typical examples of the pointer trajectories of using four selection methods. As shown, with PN, participants were forced to move the pointer around the keyboard layout to avoid false positives. Frequent false positives also led to frequent "Delete" option at the lower right corner. With PDN, the final nodding gestures were in much larger movements than the other three methods. This reflected that when the dwells triggered the highlight of the target, participants felt it safe and were more relaxed while performing the final nodding gestures. With IOI, participants preferred to cross the boundary in the direction of fewer targets around. As shown, for the top line, the pointer moved upwards first and for the bottom line, the pointer moved downwards first. This was because participants hoped to avoid entering other targets while performing IOI gestures. With *HeadCross*, participants preferred to start the gestures from above the targets regardless of the target positions.

7.9 Discussion

In this experiment, we explored alternative designs for the *HeadCross* gesture. The results showed that "Point and Nod" was not usable unless we added a short dwell in between; the "Inside-Outside-Inside" confirmation design was perceived as the most accurate and most powerful in rejecting false positives; the original *HeadCross* design was perceived to be the most efficient selection methods. Considering we only tested *Keyboard* task, these designs could be applied in different types of tasks with different levels of selection speed and accuracy requirements. For example, if users were on an unstable bus, they could use IOI for the target selection tasks method while if they were sitting still, they could use *HeadCross* to improve the selection speed.

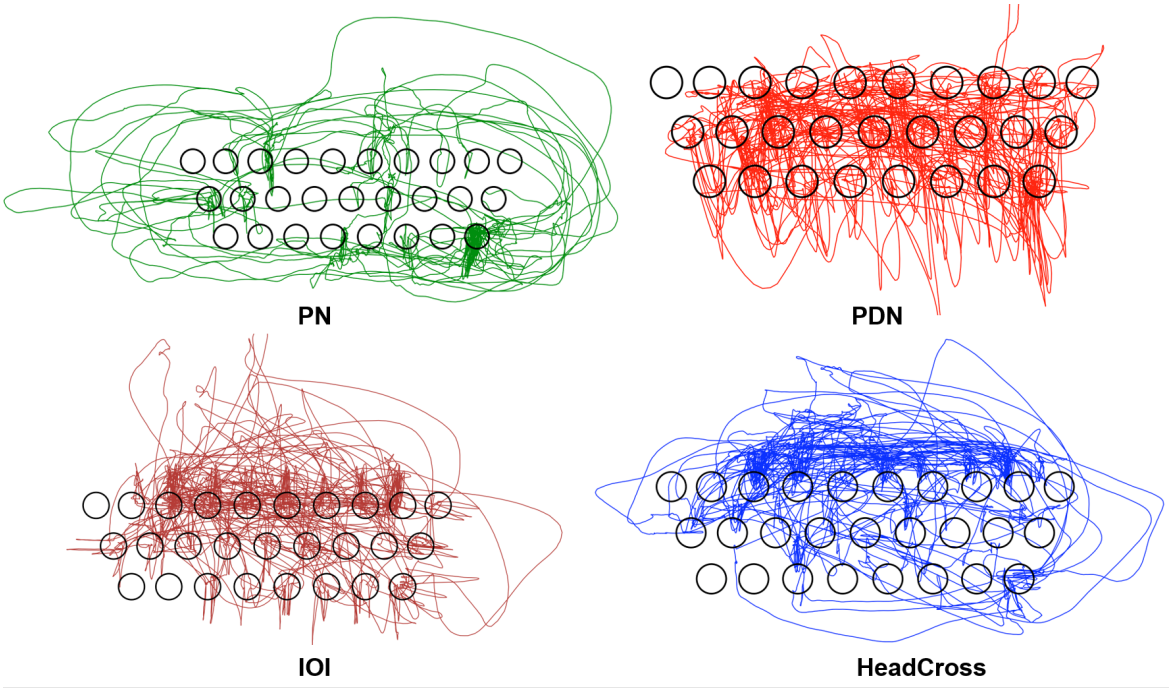

Fig. 8. Examples of pointer trajectories of the participants selecting targets with different selection methods in one session.

8 DISCUSSION

We propose *HeadCross*, a head-based target selection technique that can be applied on both VR and AR headsets. Through three user studies, we address the false positive problems of *HeadCross* and compare it to popular target selection methods and other head-based selection methods. Based on the results, we discuss the comparisons and the requirements and potential usage scenarios of applying *HeadCross*.

8.1 Head V.S. Hand

In Study 2, we compared *HeadCross* to a hand-based and a controller-based selection method. The results showed that the efficiency of *HeadCross* was in the middle of three methods while it reduced overall fatigue compared to the other two methods. *HeadCross* and *hand* moved the pointer with head movements. The only difference was the confirmation step. *HeadCross* saved time without requiring an extra channel switch from head movements to hand gestures. *HeadCross* and *controller* both used one interaction channel, however, the head movements were not as efficient as ray-casting pointing by the controller and the head gesture was also slower than the button-press action. Participants reported being increasingly exhausted while using *hand*, which also led to a decline in efficiency. Compared to hand-based methods, users perceived *HeadCross* to be more relaxing and they reported high willingness to use *HeadCross* when the hands were occupied. *HeadCross* and the other alternative designs that we tested may also improve the accessibility to AR and VR for the users with limited arm mobility.

8.2 Applicable Platform

One basic requirement for the platform to apply *HeadCross* is head orientation tracking of the users, which can be achieved by different sensing techniques including computer vision [42], computational estimation based on inertial sensor data [23]. So *HeadCross* can also be implemented for mobile phones, desktops and remote displays equipped with cameras. With mobile phones, *HeadCross* may require users to hold the phone with one hand, which is not in a totally hands-free manner. *HeadCross* also has the potential to be applied with gaze interaction. EyeK [49] has implemented a similar technique, however, due to the limitation of detection accuracy, it requires a three-step ("Inside-Outside-Inside") selection, which limits the input speed. As the gaze detection improves, we can expect to apply *HeadCross* with gaze interaction in the future.

9 LIMITATIONS AND FUTURE WORK

There are several limitations of this work, which we also consider as opportunities for future work. First, in Study 2, *HeadCross* did not outperform *controller* in selection speed on both Keyboard and *Menu* tasks. To improve the efficiency of *HeadCross*, we will explore selecting multiple targets with one *HeadCross* gesture in the future. Second, in Study 3, *HeadCross* was less accurate than another alternative design IOI. We can further improve the recognition of *HeadCross*. Compared to our current implementation, we can normalize the features, including "distance_inside", based on the target size which enables the algorithm to recognize the gestures adaptively; we can collect more data of overshoots and undershoots to understand these behaviors and build algorithms to recognize them; we can provide visual feedback of dwelling time [13] for the dwell-based method in future evaluations, which may improve user experience and obtain more realistic comparison results. Third, we only tested two extreme conditions of target density in Study 1, which were very dense layout and only one target. In the future, we can test more levels of density and study the influence it makes on user behaviors of target selection.

10 CONCLUSION

This research proposes *HeadCross*, a novel head-based target selection method for AR and VR. We first conducted a user study to observe user behaviors of performing *HeadCross* compared to head movements of passing or dwelling on the targets. Based on the results, we improved the interaction of *HeadCross* by adding the visual feedback of the pointer trajectory and implemented the recognition algorithm for *HeadCross*. Then through two user studies, we evaluated the performance of *HeadCross* and explored other alternative designs for the head gesture. Results showed that *HeadCross* achieved competitive selection speed to dwell-based methods and reduced the figure compared to hand-based methods. Compared to other alternative designs of the head gestures, *HeadCross* was perceived significantly faster than alternatives but not as accurate as IOI. To conclude, we discuss the applicability and potential improvements of *HeadCross*.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001200, the Natural Science Foundation of China under Grant No. 61521002, No. 61672314, and also by Beijing Key Lab of Networked Multimedia.

REFERENCES

[1] 2018. HTC VIVE Pro website. Website. Retrieved August 27, 2018 from https://www.vive.com/us/product/vive-pro/.
[2] 2018. Microsoft Hololens. Website. Retrieved March 7, 2018 from https://www.microsoft.com/en-us/hololens.
[3] 2019. Number Hunt. Website. Retrieved April 20, 2019 from https://store.steampowered.com/app/851770/Number_Hunt/, annote = Website URL.

[4] 2019. Serious Sam VR: The Last Hope. Website. Retrieved April 20, 2019 from https://store.steampowered.com/app/465240/Serious_Sam_VR_The_Last_Hope/.

[5] 2019. Virtual Army: Revolution. Website. Retrieved April 20, 2019 from https://store.steampowered.com/app/551610/Virtual_Army_Revolution/.

[6] Johnny Accot and Shumin Zhai. 2002. More Than Dotting the I's — Foundations for Crossing-based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. ACM, New York, NY, USA, 73–80. https://doi.org/10.1145/503376.503390

[7] I Elaine Allen and Christopher A Seaman. 2007. Likert scales and data analyses. *Quality progress* 40, 7 (2007), 64–65.

[8] Georg Apitz and François Guimbretière. 2004. CrossY: A Crossing-based Drawing Application. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 3–12. https://doi.org/10.1145/1029632.1029635

[9] GR Arce. 2005. Nonlinear Signal Processing: A Statistical Approach Wiley: New Jersey.

[10] Rowel Atienza, Ryan Blonna, Maria Isabel Saludares, Joel Casimiro, and Vivencio Fuentes. 2016. Interaction techniques using head gaze for virtual reality. In *2016 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 110–114.

[11] Gilles Bailly, Robert Walter, Jörg Müller, Tongyan Ning, and Eric Lecolinet. 2011. Comparing free hand menu techniques for distant displays using linear, marking and finger-count menus. In *Human-Computer Interaction–INTERACT 2011*. Springer, 248–262.

[12] Steffi Beckhaus, Kristopher J Blom, and Matthias Haringer. 2007. ChairIO–the chair-based Interface. *Concepts and technologies for pervasive games: a reader for pervasive gaming research* 1 (2007), 231–264.

[13] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of Eye-Gaze over Head-Gaze-Based Selection in Virtual and Augmented Reality under Varying Field of Views. In *Proceedings of the Workshop on Communication by Gaze Interaction (COGAIN âĂŹ18)*. Association for Computing Machinery, New York, NY, USA, Article Article 1, 9 pages. https://doi.org/10.1145/3206343.3206349

[14] Evren Bozgeyikli, Andrew Raij, Srinivas Katkoori, and Rajiv Dubey. 2016. Point & Teleport Locomotion Technique for Virtual Reality. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '16)*. ACM, New York, NY, USA, 205–216. https://doi.org/10.1145/2967934.2968105

[15] Marcio C. Cabral, Carlos H. Morimoto, and Marcelo K. Zuffo. 2005. On the Usability of Gesture Interfaces in Virtual Reality Environments. In *Proceedings of the 2005 Latin American Conference on Human-computer Interaction (CLIHC '05)*. ACM, New York, NY, USA, 100–108. https://doi.org/10.1145/1111360.1111370

[16] Eun Kyoung Choe, Kristen Shinohara, Parmit K. Chilana, Morgan Dixon, and Jacob O. Wobbrock. 2009. Exploring the Design of Accessible Goal Crossing Desktop Widgets. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 3733–3738. https://doi.org/10.1145/1520340.1520563

[17] Andy Cockburn and Andrew Firth. 2004. Improving the acquisition of small targets. In *People and Computers XVIIâĂŤDesigning for Society*. Springer, 181–196.

[18] Douglas A Craig and Hung T Nguyen. 2006. Wireless real-time head movement system using a personal digital assistant (PDA) for control of a power wheelchair. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. IEEE, 772–775.

[19] Andrew Crossan, John Williamson, Stephen Brewster, and Rod Murray-Smith. 2008. Wrist Rotation for Interaction in Mobile Contexts. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '08)*. ACM, New York, NY, USA, 435–438. https://doi.org/10.1145/1409240.1409307

[20] Gerwin de Haan, Eric J. Griffith, and Frits H. Post. 2008. Using the Wii Balance Board&Trade; As a Low-cost VR Interaction Device. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology (VRST '08)*. ACM, New York, NY, USA, 289–290. https://doi.org/10.1145/1450579.1450657

[21] Pierre Dragicevic. 2004. Combining Crossing-based and Paper-based Interaction Paradigms for Dragging and Dropping Between Overlapping Windows. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 193–196. https://doi.org/10.1145/1029632.1029667

[22] Jinjuan Feng and Andrew Sears. 2004. Using Confidence Scores to Improve Hands-free Speech Based Navigation in Continuous Dictation Systems. *ACM Trans. Comput.-Hum. Interact.* 11, 4 (Dec. 2004), 329–356. https://doi.org/10.1145/1035575.1035576

[23] Eric Foxlin. 1996. Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. In *Virtual Reality Annual International Symposium, 1996., Proceedings of the IEEE 1996*. IEEE, 185–194.

[24] Dmitry O Gorodnichy and Gerhard Roth. 2004. Nouse âĂŸuse your nose as a mouseâĂŹperceptual vision technology for hands-free games and interfaces. *Image and Vision Computing* 22, 12 (2004), 931–942.

[25] Hiroyuki Hakoda, Takuro Kuribara, Keigo Shima, Buntarou Shizuki, and Jiro Tanaka. 2015. AirFlip: A double crossing in-air gesture using boundary surfaces of hover zone for mobile devices. In *International Conference on Human-Computer Interaction*. Springer, 44–53.

[26] Jibo He, Alex Chaparro, Bobby Nguyen, Rondell Burge, Joseph Crandall, Barbara Chaparro, Rui Ni, and Shi Cao. 2013. Texting While Driving: Is Speech-based Texting Less Risky Than Handheld Texting?. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '13)*. ACM, New York, NY, USA, 124–130. https://doi.org/10.1145/2516540.2516560

[27] Shahram Jalaliniya, Diako Mardanbegi, and Thomas Pederson. 2015. MAGIC pointing for eyewear computers. In *Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 155–158.

[28] Shahram Jalaliniya, Diako Mardanbeigi, Thomas Pederson, and Dan Witzner Hansen. 2014. Head and eye movement as pointing modalities for eyewear computers. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks Workshops*. IEEE, 50–53.

[29] Pei Jia, Huosheng H Hu, Tao Lu, and Kui Yuan. 2007. Head gesture recognition for hands-free control of an intelligent wheelchair. *Industrial Robot: An International Journal* 34, 1 (2007), 60–68.

[30] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI âĂŹ18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 81, 14 pages. https://doi.org/10.1145/3173574.3173655

[31] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 81, 14 pages. https://doi.org/10.1145/3173574.3173655

[32] Edmund LoPresti, David M. Brienza, Jennifer Angelo, Lars Gilbertson, and Jonathan Sakai. 2000. Neck Range of Motion and Use of Computer Head Controls. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies (Assets '00)*. ACM, New York, NY, USA, 121–128. https://doi.org/10.1145/354324.354352

[33] Yuexing Luo and Daniel Vogel. 2014. Crossing-based Selection with Direct Touch Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2627–2636. https://doi.org/10.1145/2556288.2557397

[34] Yuexing Luo and Daniel Vogel. 2015. Pin-and-Cross: A Unimanual Multitouch Technique Combining Static Touches with Crossing Selection. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 323–332. https://doi.org/10.1145/2807442.2807444

[35] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971

[36] Diako Mardanbegi, Dan Witzner Hansen, and Thomas Pederson. 2012. Eye-based head gestures. In *Proceedings of the symposium on eye tracking research and applications*. ACM, 139–146.

[37] Diako Mardanbegi, Benedikt Mayer, Ken Pfeuffer, Shahram Jalaliniya, Hans Gellersen, and Alexander Perzl. 2019. EyeSeeThrough: Unifying Tool Selection and Application in Virtual Environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 474–483.

[38] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 653, 13 pages. https://doi.org/10.1145/3173574.3174227

[39] Mark R Mine. 1995. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept* (1995).

[40] Louis-Philippe Morency and Trevor Darrell. 2006. Head Gesture Recognition in Intelligent Interfaces: The Role of Context in Improving Recognition. In *Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI '06)*. ACM, New York, NY, USA, 32–38. https://doi.org/10.1145/1111449.1111464

[41] Takashi Nakamura, Shin Takahashi, and Jiro Tanaka. 2008. Double-crossing: A new interaction technique for hand gesture interfaces. In *Asia-Pacific Conference on Computer Human Interaction*. Springer, 292–300.

[42] Diederick C Niehorster, Li Li, and Markus Lappe. 2017. The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research. *i-Perception* 8, 3 (2017), 2041669517708205.

[43] Tomi Nukarinen, Jari Kangas, Oleg Špakov, Poika Isokoski, Deepak Akkil, Jussi Rantala, and Roope Raisamo. 2016. Evaluation of HeadTurn: An Interaction Technique Using the Gaze and Head Turns. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, Article 43, 8 pages. https://doi.org/10.1145/2971485.2971490

[44] Takeshi Okunaka and Yoshinobu Tonomura. 2012. Eyeke: What You Hear is What You See. In *Proceedings of the 20th ACM International Conference on Multimedia (MM '12)*. ACM, New York, NY, USA, 1287–1288. https://doi.org/10.1145/2393347.2396445

[45] Andriy Pavlovych and Wolfgang Stuerzlinger. 2009. The Tradeoff Between Spatial Jitter and Latency in Pointing Tasks. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '09)*. ACM, New York, NY, USA, 187–196. https://doi.org/10.1145/1570433.1570469

[46] Kathrin Probst, David Lindlbauer, Michael Haller, Bernhard Schwartz, and Andreas Schrempf. 2014. A Chair As Ubiquitous Input Device: Exploring Semaphoric Chair Gestures for Focused and Peripheral Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 4097–4106. https://doi.org/10.1145/2556288.2557051

[47] Siddharth S Rautaray and Anupam Agrawal. 2011. Interaction with virtual game through hand gesture recognition. In *2011 International Conference on Multimedia, Signal Processing and Communication Technologies*. IEEE, 244–247.

[48] Gang Ren and Eamonn O'Neill. 2012. 3D marking menu selection with freehand gestures. In *3D User Interfaces (3DUI), 2012 IEEE Symposium on*. IEEE, 61–68.

[49] Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: An Efficient Dwell-free Eye Gaze-based Text Entry System. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction (APCHI '13)*. ACM, New York, NY, USA, 215–220. https://doi.org/10.1145/2525194.2525288

[50] Robin Schweigert, Valentin Schwind, and Sven Mayer. 2019. EyePointing: A Gaze-Based Selection Technique. In *Proceedings of Mensch Und Computer 2019 (MuC'19)*. ACM, New York, NY, USA, 719–723. https://doi.org/10.1145/3340764.3344897

[51] Ludwig Sidenmark and Hans Gellersen. 2019. Eye&Head: Synergetic Eye and Head Movement for Gaze Pointing and Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 1161–1174.

[52] Oleg Špakov and Päivi Majaranta. 2012. Enhanced gaze interaction using simple head gestures. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 705–710.

[53] Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 265–272. https://doi.org/10.1145/223904.223938

[54] Zhenyu Tang, Chenyu Yan, Sijie Ren, and Huagen Wan. 2016. HeadPager: Page Turning with Computer Vision Based Head Interaction. In *Asian Conference on Computer Vision*. Springer, 249–257.

[55] Sam Tregillus. 2016. VR-Drop: Exploring the Use of Walking-in-Place to Create Immersive VR Games. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 176–179. https://doi.org/10.1145/2851581.2890374

[56] Sam Tregillus and Eelke Folmer. 2016. VR-STEP: Walking-in-Place Using Inertial Sensing for Hands Free Navigation in Mobile VR Environments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1250–1255. https://doi.org/10.1145/2858036.2858084

[57] Javier Varona, Cristina Manresa-Yee, and Francisco J Perales. 2008. Hands-free vision-based interface for computer accessibility. *Journal of Network and Computer Applications* 31, 4 (2008), 357–374.

[58] Jia Wang and Robert W. Lindeman. 2011. Silver Surfer: A System to Compare Isometric and Elastic Board Interfaces for Locomotion in VR. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces (3DUI '11)*. IEEE Computer Society, Washington, DC, USA, 121–122. http://dl.acm.org/citation.cfm?id=2013881.2014229

[59] Wenchang Xu, Chun Yu, Jie Liu, and Yuanchun Shi. 2015. RegionalSliding: Facilitating small target selection with marking menu for one-handed thumb use on touchscreen-based mobile devices. *Pervasive and Mobile Computing* 17 (2015), 63–78.

[60] Xuhai Xu, Alexandru Dancu, Pattie Maes, and Suranga Nanayakkara. 2018. Hand Range Interface: Information Always at Hand with a Body-centric Mid-air Input Surface. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. ACM, New York, NY, USA, Article 5, 12 pages. https://doi.org/10.1145/3229434.3229449

[61] Xuhai Xu, Chun Yu, Anind K. Dey, and Jennifer Mankoff. 2019. Clench Interface: Novel Biting Input Techniques. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI âĂŹ19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 275, 12 pages. https://doi.org/10.1145/3290605.3300505

[62] Xuhai Xu, Chun Yu, Yuntao Wang, and Yuanchun Shi. 2020. Recognizing Unintentional Touch on Interactive Tabletop. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 1 (March 2020), 27. https://doi.org/10.1145/3381011

[63] Yukang Yan, Chun Yu, Xiaojuan Ma, Shuai Huang, Hasan Iqbal, and Yuanchun Shi. 2018. Eyes-Free Target Acquisition in Interaction Space Around the Body for Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 42, 13 pages. https://doi.org/10.1145/3173574.3173616

[64] Yukang Yan, Chun Yu, Xin Yi, and Yuanchun Shi. 2018. HeadGesture: Hands-Free Input Approach Leveraging Head Movements for HMD Devices. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 4, Article 198 (Dec. 2018), 23 pages. https://doi.org/10.1145/3287076

[65] Chuang-Wen You, Yung-Huan Hsieh, and Wen-Huang Cheng. 2012. AttachedShock: facilitating moving targets acquisition on augmented reality devices using goal-crossing actions. In *Proceedings of the 20th ACM international conference on Multimedia*. ACM, 1141–1144.

[66] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, Dwell or Gesture?: Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4479–4488. https://doi.org/10.1145/3025453.3025964