Headbang: Using Head Gestures to Trigger Discrete Actions on Mobile Devices

Sebastian Hueber
RWTH Aachen University
Aachen, Germany
hueber@cs.rwth-aachen.de

Christian Cherek
RWTH Aachen University
Aachen, Germany
cherek@cs.rwth-aachen.de

Philipp Wacker
RWTH Aachen University
Aachen, Germany
wacker@cs.rwth-aachen.de

Jan Borchers
RWTH Aachen University
Aachen, Germany
borchers@cs.rwth-aachen.de

Simon Voelker
RWTH Aachen University
Aachen, Germany
voelker@cs.rwth-aachen.de

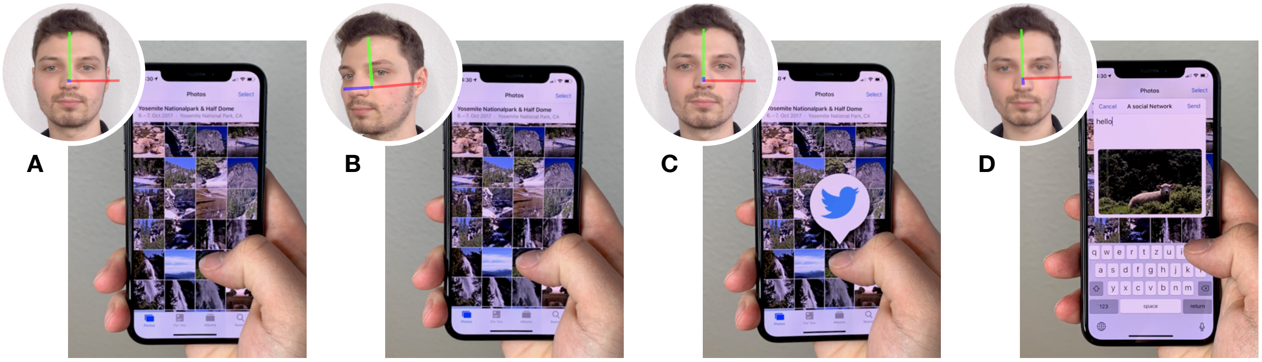

Figure 1: (a) To share an image to a social media application with a Headbang gesture, a user touches an image with his finger and (b) immediately moves his head slightly away from the device and back again. (c) As soon as the gesture is detected, an indicator above the image displays the selected action. (d) To confirm this action the user lifts the finger from the image.

ABSTRACT

We present Headbang, an interaction technique that enriches touch input on handheld devices through slight head movement gestures. This way, users can easily execute shortcuts, like Copy, Paste, or Share, to on-screen targets while touching them. Headbang utilizes the capabilities of commodity smartphones to track the user's head with their front facing cameras. We evaluated Headbang in two studies and show that the system can be reliably used while sitting and walking and offers a similar accuracy and speed as touch interaction.

CCS CONCEPTS

- **Human-centered computing → Gestural input**.

KEYWORDS

head tracking; head gestures; mobile devices; pie menus

ACM Reference Format:
Sebastian Hueber, Christian Cherek, Philipp Wacker, Jan Borchers, and Simon Voelker. 2020. Headbang: Using Head Gestures to Trigger Discrete Actions on Mobile Devices. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '20), October 5–8, 2020, Oldenburg, Germany.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3379503.3403538

1 INTRODUCTION

Touch input is prevalently used in mobile devices. However, its expressiveness is limited. Simple inputs such as tapping and swiping are usually already occupied with the semantics of selection and scrolling. Options that are familiar from desktop interaction, such as right-click or keyboard shortcuts do not exist, and many common tasks require longer sequences of selections that hardly benefit from multitouch [20]. The limited screen space on mobile devices increases this issue, as there is not enough space to present large toolbars. Therefore, most modern smartphone OS make use of time-based inputs to show context menus.

To alleviate this issue, a variety of touch techniques have emerged, such as stroke gestures [1], multitouch gestures [12], force touch input [6], and utilizing further sensors built into smartphones and tablets, such as motion sensors for tilt input [2].

MobileHCI '20, October 5–8, 2020, Oldenburg, Germany

ACM ISBN 978-1-4503-7516-0/20/10…$15.00
https://doi.org/10.1145/3379503.3403538

Advanced camera systems and processing power in modern mobile devices allow tracking the user's body and head. These tracking capabilities allow to augment touch input with with head movements to increase its expressiveness. For example, performing shortcuts on specific items, or enhancing one handed use by removing the need for specific menu buttons on the screen edges.

Head movement is a common social communication method [16, 24] that has also been used to interact with interactive systems. For example, to move a cursor on a desktop computer [11], as command gestures while using head-mounted displays [41], or to interact with mobile devices [8]. However, formerly additional hardware external to the primary device was needed for tracking.

While interacting with their phones, users are focusing on the screen and their head is oriented towards the device, making vision-based head control an interesting field of research. We present Headbang, an interaction technique that allows users to trigger actions by slightly rotating their head into different directions. As Headbang works with the visual input from the built-in front facing camera, it does not require additional hardware and saves value screen space for toolbars.

After discussing related work, we present our Headbang interaction technique and its implementation in more detail. In our studies Headbang gestures were detected reliably while sitting and walking and offer a promising alternative to context menus. We close with a discussion and a collection of use cases for the Headbang interaction technique.

The main contributions of this paper are as follows: First, we provide a quantification of the obtainable head-tracking precision on a commodity smartphone and its robustness against walking. Second, we present the Headbang interaction technique to increase expressiveness of touch input and propose different use cases. Third, we provide a comparison of Headbang with alternatives such as device tilting.

2 RELATED WORK

As the detection of touch input is only binary, i.e. detected or not detected, its expressiveness is limited. Together with the fat finger problem and limited screen space, mobile devices make it impossible to use toolbars as known from the desktop. Instead, developers opt to use menus that are accessed by mode switches, e.g. by a long press.

2.1 Touch Gestures

A variety of different touch techniques have been introduced to mitigate the issue of the limited expressiveness of touch input. Several interaction techniques use single-figure stroke gestures to access menus or shortcuts [1, 20, 43]. However, several of these gestures, such as swiping or flicking, have become standard system gestures for navigation and interaction and therefore cannot be used to access additional shortcuts. Furthermore, the menus themselves require large enough screen space for every item for reliable touch input.

A common way to customize interaction is to use multi-finger touch gestures [12]. These gestures are intuitive but not feasible in many situations since users often interact with their smartphones using only the thumb for input [3, 13, 15]. Force input is another method to access menus or shortcuts [6]. However, it requires significant learning [5], is difficult to control while moving [40], and often not available on current smartphones. Chen et al. [4] explored the combination of touch and in-air input. However, additional hardware makes these concepts difficult to use in a real world-scenario.

2.2 Tilting Interfaces

Tilting input has been well explored since the 1990s [30]. As it does not need a display for input, tilting allows for usage on small devices and is especially promising for smartphone UIs that offer small screen space. For triggering actions, pie menus have proven a good mapping of device rotation to a menu item [2, 35]. Other explored usage areas for tilting input include scrolling [27, 31] and text entry [28, 38]. However, these use cases are outperformed by touch input on smartphones.

2.3 Head Tracking

Modern mobile devices can also track the users eyes, yet tracking can be very inaccurate, especially when users are moving [17]. Head gestures are a common communication method when people interact with each other. For example, nodding and shaking are used to express *yes* or *no* [16], or more complex messages such as acknowledgment or disinterest [24]. In HCI, head gestures were also explored to be used as input technique [21], especially for users with limited arm mobility, e.g., to control wheelchairs [7, 22].

Still, head movement can also be a useful additional input method for able-bodied users [10, 23]. It has been used as continuous input to move the cursor on desktop computers [11, 29], or to change the viewport in a 3D-application [14]. Discrete operations are also promising application areas for head gestures. HeadTurn [26, 44] allows users to change numeric values by rotating their head left or right, HeadPager [33] enables users to turn pages by leaning their heads to the left or the right area, and HeadNod [25] allows users to quickly answer *yes* or *no* in a dialogue by nodding or shaking their head.

Head mounted devices can use the tracked head position in many ways, such as authenticating users based on their head gestures [42] or as a method to select moving targets by following their trajectories [9]. Yukang et al. [41] conducted an elicitation study in which they explored what kind of head gestures could be used to create hands-free input while using an HMD device. They found that head gestures that involved turning the head in one direction and back again were preferred by the users and are also distinguished reliably from normal head movement. According to Kytö et al. [19], head-based selection is easy to control and more accurate but slower than eye-based selection while using an AR headset.

On handheld devices, head movement has also been explored. Crossan et al. [8] explored how accurately users can select a target on a smartphone while walking using head tilting to control the cursor. They found that absolute cursor control was faster and more accurate than velocity cursor control in a static context but significantly worse while moving. Williamson et al. [39] used shake sensors and compared head gestures with wrist and device tilting gestures and showed that head gestures have a similar accuracy to wrist or device motion gestures. However, they also showed

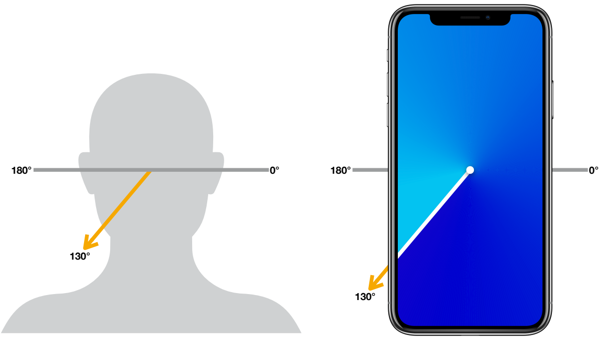

Figure 2: We measure rotation clockwise with 0° pointing to the right. The interface depicts the app used in study 1, with the line representing a desired rotation of 130°.

that users felt uncomfortable to do head gestures while having a conversation with other people.

The aforementioned mobile device approaches require motion that are attached to the users to detect the head movement. This additional hardware makes these approaches difficult to use the the real-world. However, due to recent advances in computer vision on mobile devices, modern smartphones are capable of tracking the users head with the front camera of the device [36] .

3 INFERRING A ROTARY ANGLE FROM MOBILE HEAD TRACKING

While interacting with their phones, users are focusing on the screen and their head is oriented towards the device. As users usually hold their device at about 50cm distance the front facing camera can comfortably record the head during interaction. Therefore, we can use it to infer input from the user's head posture. By shooting a ray cast from the head, the directional movement of the intersection point on the device plane can be interpreted as rotational input. We defined this rotational input clockwise with the 0° angle pointing to the right, as depicted in Figure 2.

Our implementation is based on ARKit[1] which provides us with a position and orientation of the user's head in the coordinate system originating at the device's front camera. The head's rotary yaw and pitch are then transferred into a point on the device plane by intersecting a ray from the head with a plane that is defined by the camera's orientation vector as normal. Using the device plane instead of a plane orthogonal to the head has two advantages. First, a typical posture holding a smartphone, implies bowing the head towards the phone, which limits further movement towards the chin. Therefore a downwards head movement requires less movement. Second, the system can be used when even while the phone is tilted while the mapping of reaching the topmost item remains intact. The angle is finally inferred by analyzing the path the head's intersection point moves along.

3.1 Enabling correction of wrong inputs

We store the point that was measured when the interaction started and a buffer of the 60 most recent points. As the device is using a sampling rate of 60Hz, i.e. we analyze the movement occurring in a movement of at most 1 second. The point in this buffer that has the farthest distance to the starting point is then used to calculate the angle on the device, converting the 3D rotation into a 2D one.

This approach makes corrections possible while the angle is determined. For example, a clockwise rotation is tracked without a noticeable delay.

This interaction technique suits the modality of head position well, as the intersection of the head (hence, absolute accuracy) is limited, but the accuracy of relative movements is easy to control and, thus, accurate. In fact the measured data was stable enough to have no need for further filtering which would only slow down tracking.

3.2 Measuring Head Tracking Accuracy

We conducted a study with 8 participants (22–28 years, $M = 25.25$, $SD = 1.75$, 3 female) to determine the deviation of measured angles.

Apparatus and Task. An iPhone X was given to the participants that they held at a typical position they hold their phone, i.e. with a distance of roughly 50cm. The study app presented the participants a white line ranging from the center of the screen to its border with an angle. We tested all multiples of 10 degrees in a random order and with two repetitions, i.e. a total of 72 trials per participant. Even though the perceived mapping of motion on the device to a head gesture is subjective, we visually indicated the gesture to perform. On touch down, a dot moved linearly paced along the path with a speed of 2.4cm/sec. Participants were asked to target the dot with their head, then touch and hold anywhere on the screen and follow the dot along the path with their head. Once the measured intersection point moved farther than 4.8cm, the system gave haptic feedback, participants lifted their finger and continued to the next angle. The interface is depicted in Figure 2.

Results. Since we were interested in the accuracy that is achievable with this system including the human as error factor, we did not provide feedback. The movement speed was required to follow the dot was left for interpretation of the participant. However, trials always took less than two seconds to complete. The average offset between the targeted angle and the angle that was measured from our system was 11.67°. The spread of the data had a large standard deviation of 8.13°. While a sixth of the measured samples had a very high accuracy of less than 3°error, the maximum error we measured was 35.4°.

4 HEADBANG INTERACTION TECHNIQUE

Headbang allows to trigger an action on a specific object such as sharing a photo to Twitter by rotating the head slightly. To do so, the user touches the photo she wants to share (Fig. 1.a) and then immediately rotates her head slightly away from the screen and back again (Fig. 1.b). The connected action is then displayed on the screen (Fig. 1.c) and the user can lift her finger to perform the action (Fig. 1.d). Since in touch interfaces an action is typically triggered by lifting the finger from an object and not by touching

---

[1] https://developer.apple.com/documentation/arkit

it, the Headbang interaction sequence does not overload existing interaction concept of handheld touch devices and can co-exist with common touch-based interaction technique such as tap, long press, swipe, or drag.

Using head gestures allows to add further actions without having to rely on multi-touch or swipe gestures that have become standard system commands. As users only have to move their head slightly, it is still possible to keep the eyes on the screen to keep the context. Furthermore, the back and forth movement can be easily performed and can be distinguished from normal movement [41]. The head gesture takes place between tapping and releasing the object. This makes it possible to give information about the currently determined action before the user confirms this action by releasing the finger. For example, tapping and holding an image and performing the head gesture already shows a popout indicating that 'share to Twitter' has been detected. If this is the action the user wants to perform, she can lift the finger. Alternatively, it is possible to cancel the action by sliding the finger outside of the selected image before lifting it similarly as canceling a button press on current mobile operating systems. This provides an easy way to cancel unwanted actions and also enables new users to use this system without the risk of immediately performing unwanted actions. Expert users do not have to wait for the visual indication and instead can touch a target, perform the head gesture, and release the finger from the screen before the head gesture is detected. However, using this faster mode of Headbang prevents the users to cancel their actions.

Various natural mappings seem promising for Headbang inputs. Cultural mappings can imitate common head gestures: A down movement can be used for the confirmation of a message as it imitates a nod; a sideways gesture could be used for rejections. Spatial mappings can be inspired by the direction of the action or their typical icon representation in the UI. For example, forwarding, replying, sharing, or printing an email with head gestures to the right, left, top or bottom.

We implemented Headbang with different numbers of directions and evaluated the performance in the two following user studies. Furthermore, we implemented several example applications using this technique and present them in the 'Use Cases' section.

5 INVESTIGATING TRACKING ROBUSTNESS

To understand how well our envisioned interaction technique works we conducted a study in which participants triggered Headbang actions with 4 and 8 directions both while sitting and walking. For the study we recruited 12 participants aged between 19 and 65 years ($M = 36.3, SD = 14.9$, 5 female).

5.1 Apparatus and Task

For the study we used an iPhone X that detects the user's head as described above. Users were asked to perform the back-and-forth Headbang gestures, as described in the section above, in a sitting position and while walking. Similar to the walking condition by Crossan et al.[8], the users were instructed to walk a figure-of-eight across a three by four meter rectangle. We placed obstacles for users to walk around to make sure they still had to pay attention to where they were walking.

Our participants were asked to perform Headbang gestures in the direction of an arrow in the UI. Feedback was deactivated for the study. Each user performed each gesture twelve times. As we investigated both 4 and 8 directions, this results in a total of 288 gestures across all four conditions for each participant. To counterbalance possible learning effects, conditions were assigned conditions in a latin-square design.

At the start of each trial, users were shown a button in the bottom third of the screen with an arrow pointing into the direction of the gesture they should perform. The button position was varied among four different positions, and the different gestures were pseudo-randomly distributed over the buttons to mimic a more natural interaction. We made sure that all buttons could easily be reached in one-handed portrait mode. To start a trial, users had to press-and-hold the button and then perform the Headbang gesture. After performing the gesture, users had to release the button to start the next trial.

5.2 Variables

Since we were mostly interested in how reliable the system could distinguish in which direction the users turned their heads, we used CONTEXT [sitting, walking] and AREAS [four, eight] as independent variables. Additionally, we analyzed if the detection rate differs between the different areas. For this we used DIRECTION as additional independent variable. In the four area condition the directions were defined by multiples of 90°, i.e. right, bottom, left, and top. In the eight area condition we used multiples of 45°respectively. As dependent variables, we measured *Success* [0,1] if the system was able to identify the correct area and the task completion *Time* [s] for each trial.

5.3 Results

The overall success rate was 95.22% (SD: 8.38%) with an average task completion time of 1.39s (SD: 0.73s) across all trials and users. For a more detailed analysis, we used McNemar and Cochran's Q tests for the dichotomous *Success* data and we conducted a repeated-measures ANOVA on the logtransformed *Time* data.

CONTEXT had a significant main effect on ***Success*** ($Q(1) = 5.19$, $p < .023$). The success rate in the sitting condition (96%) was significantly higher than in the walking condition (94%). Also, the AREAS had a significant main effect on ***Success*** ($Q(1) = 6.47$, $p < .011$). The success rate in the four-area condition (96%) was significantly higher than in the eight-area condition (94%). There was also a AREAS × CONTEXT interaction effect ($Q(3) = 13.71$, $p < .003$). Post hoc tests revealed that the four-areas sitting condition (97%) had a significantly higher success rate than the eight-areas walking condition (94%). Figure 3 shows the results for this interaction.

In the four-area condition DIRECTION had a significant main effect on ***Success*** ($Q(3) = 48.303$, $p < .001$). Post-hoc tests revealed that the bottom direction had a significantly lower success rate (91%) than the other direction (top: 99%, left: 98%, right: 100%). In the eight-area condition DIRECTION had a significant main effect on ***Success*** ($Q(7) = 74.220$, $p < .001$). Post-hoc tests revealed that the bottom-right (45°) direction (86% success rate) had a significantly lower success rate than the other directions.

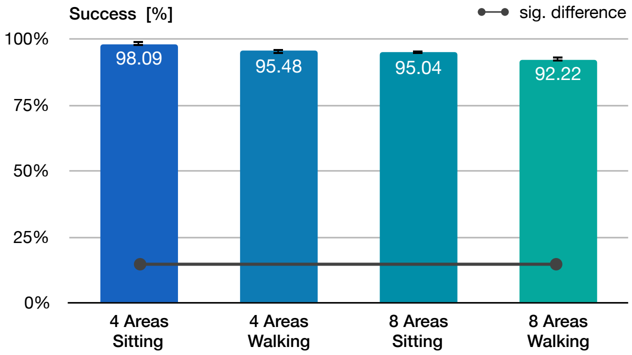

Figure 3: ***Success*** **[%] by Areas × Context. Whiskers denote 95% CI.**

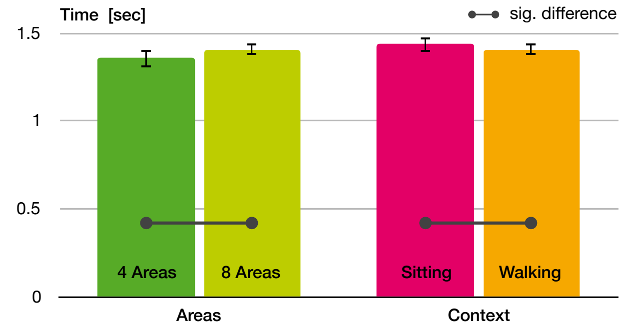

Figure 4: ***Time*** **by number of Areas (left) and by Context (right). Whiskers denote 95% CI.**

Context had a significant main effect on ***Time*** ($F_{1,3441}$ = 30.518, p <.001). Users were significantly faster in the four-area condition (1.29s) than in the eight-area condition (1.35s). Also Areas had a significant main effect on ***Time*** ($F_{1,3441}$ = 31.602, p <.001). Users were significantly faster in the walking condition (1.28s) than in the sitting condition (1.35s). Figure 4 shows the results of both main effects.

5.4 Discussion

With an overall success rate of 95%, our evaluation shows that Headbang gestures can be used reliably for input. Yet the success rate is significantly different between the different conditions. The higher success rates in the four area conditions had to be expected, as the cone of a single section is twice as large as in the eight area condition. Likewise, the difference in success between the sitting and walking condition is no surprise. In the sitting condition, both the device and the user are static, which allows the system to track the user's head more accurately.

The result of the bottom direction having a slightly lower success rate in the four-area condition can be explained by how the users hold and look at the device. We observe that users hold their smartphone not directly in front of the head but much lower to have a comfortable arm position. This means that the users already rotate their heads downwards to look at the content display on the smartphone. An even further downwards rotation of the head to select the bottom area could be limited due to the neck muscles or discomfort for the users. Therefore participants overshoot to the top when performing a back-and-forth gesture. A similar effect, however not significantly, also appears in the eight-area condition. Here, all three bottom directions have a lower success rate than the other areas. However, only the lower right direction was significantly different from the other areas. The reason for this could be that most of the users (10 out of 12) were right-handed and therefore did hold the device on the right side. In this case, the users had to move their head further to the right to select the lower bottom direction.

Figure 5: The segments were equally sized in the pie menus and oriented so that four segments were exactly aligned with the horizontal and vertical axis, i.e. the 90° steps.

The study also showed significant differences for the task completion time between the conditions. While walking participants were around 30ms faster than while sitting. This could be participants shifted their attention away from the phone earlier to not run into an obstacle while walking. However, this difference is minimal and probably not noticeable in a real-world use case.

6 USING HEAD TRACKING TO TRIGGER ACTIONS IN MENUS

As we have seen, Headbang can reliably be used to trigger actions on items of interest. However, the investigated interaction technique did not contain any visual feedforward that presents the user all available actions. Due to the limited screen space available on mobile devices, the selection of an action is often selected from a context menu. The size of these menus can become quite large. On iOS 13, for instance, we can find context menus with 11 items in the Files app, 8 items in the Mail app, or 10 items in the Music app. In touch interaction, tapping and swiping are already occupied, resulting in the need to hold an item of interest for a specific duration to bring up the menu. As users can quickly recall spatial positions [32], Headbang seems to be a promising modality for menu items. Thus, we wanted to compare Headbang with touch and tilting input for menus.

Headbang menu. The Headbang menu is a pie menu whose selected item is controlled with head tilting. Upon putting a finger on an item on the screen, the camera system activates. When tilting the head slightly, i.e. by approximately 10°, a pie menu appears and the item that corresponds to the current angle is highlighted. The selection is changed by rotating the head and confirmed by lifting the finger. With a diameter of 3cm the pie menu offers a compact menu visualization that takes less screen space than the list menu at the cost of omitting labels. While maintaining its size, the pie can also depict labels for menus with few items (<7). For large menus labels in the pie are possible by increasing its diameter, while still being smaller than a full list and occupying 25% of the phone screen.

For comparison we also implemented a pie menu for device tilt and touch input. The device tilt condition behaves exactly like the Headbang menu and shows up when holding an item and tilting the device. As recommended by Teather et al. [34], we used the absolute device rotation for input. In the touch condition the menu appeared after a long press of 400ms, which is 20% faster than the iOS default. The selection in the menu was then selected by swiping in the direction of the item and confirmed by lifting the finger inside or outside of the menu.

As a baseline we implemented a list-style menu with touch targets that are 2.8×0.7cm large, adopting the same size as the system menus in iOS 13. While in real-world applications users often have to scroll through the menu, we made sure that all items were always visible on screen, limiting the maximal number of items in this condition to 16.

We used a new set of twelve participants in this study aged from 20 – 31 years ($M = 25.5, SD = 3.34$, 3 female).

6.1 Apparatus and Task

For the study we used the same iPhone model as in the previous study but extended our implementation with the different menu techniques as described above.

The interface presented the participant an emoji and a red box with the size of an app icon ($1cm^2$) in whose context menu the depicted emoji had to be selected. Moreover, an arrow next to the emoji pointed to the position it will appear in the menu once it becomes visible. This hint was provided to mitigate the search time during the interaction and to take into consideration that users create muscle memory for actions they operate frequently. We asked our participants to select the menu items as quickly as possible without compromising accuracy while using the phone with one hand only.

While all presented techniques work with an arbitrary number of items, we picked three different menu sizes with 8, 12, and 16 elements for evaluation. We found 12 items to be a reasonable number from our observation of system menus and due to its benefit of a mapping known from the hour marks of a clock. For each menu size, we selected 8 different items at the representative angles 35°, 80°, 120°, 167°, 210°, 260°, 305°, and 350°. With three repetitions this led to 288 selections for each participant (8 items × 3 menu sizes × 3 repetitions × 4 menu types). To counterbalance possible learning effects we used latin squares for both the menu size and input conditions.

Before each recording and the questionnaire participants were allowed to test the input techniques. They tested all four input conditions with the same menu size before switching to the next study phase. The new menu size had a new order of input conditions. Our participants ranked their preference between the four techniques for each menu size before switching to the next phase. Further questions were filled out at the end. There was no monetary compensation for participation.

6.2 Variables

As we conducted the study in order to find out whether Headbang can be used as a reliable modality for selecting menu actions, we used the INPUT TECHNIQUE [Headbang, Device Tilt, Touch Pie, Touch List] and the MENU SIZE [8, 12, 16] as independent variable. As dependent variables we measured *Success* [0,1] if the correct items was selected, and the task completion *Time* [s] for each trial. In addition we measured the angles obtained from head or device tilting when initiating the gesture, i.e. before feedback, and on selection, i.e. when feedback was visible. When calculating the offset to the angle representing the center of the target menu segment this results in *Initiation Offset* [°] and *Selection Offset* [°] respectively. From the questionnaires we measured *Preference* in an enforced ranking from 1–4.

6.3 Results

In this study we were most interested in the participants' performance depending on the INPUT TECHNIQUE used. Therefore, we will focus our analysis on this main effect and related interaction effects. We conducted a repeated-measures ANOVA on the (log-transformed) *Time* data.

MENU SIZE had a significant main effect on ***Time*** ($F_{2,2833} = 43.856$, $p < .001$). Tukey HSD post hoc pairwise comparisons were all significant. Menus containing 8 items were the fastest (1.273 sec). On average, menus with 12 items were 15% slower (1.472 sec) and 16 items were 29% slower (1.646 sec) than menus with 8 items. We were not able to find a significant effect of INPUT TECHNIQUE on ***Time*** ($F_{3,2833} = 1.755$, $p = .154$).

There was also a MENU SIZE × INPUT TECHNIQUE interaction effect on ***Time*** ($F_{6,2833} = 3.901$, $p < .001$). Again we used Tukey HSD post hoc pairwise comparisons for further analysis. On average, Device Tilt with 8 menu items was the fastest condition (1.094 sec), and it was significantly faster than all other conditions except from Headbang with 8 menu items and Touch Pie with 8 menu items. Larger menu sizes took longer to operate, ranging from 1.637 sec (Touch Pie) to 1.669 sec (Headbang). However, Headbang with 16 menu items was only significantly slower than Device Tilt with 8 menu items and Touch Pie with 8 items. When comparing Headbang with the Touch List there were no significant differences independently of the menu size. Figure 6 shows the measured times for all conditions.

For the analysis of the *Success* values we calculated the *Success rate* as the share of successful trials per condition and user. We then conducted a repeated-measures ANOVA on the calculated success rates.

MENU SIZE had a significant main effect on the ***Success Rate*** ($F_{2,99} = 9.004$, $p < .001$). Tukey HSD post hoc pairwise comparisons revealed that the success rate significantly drops from 98.5% to 94.0% when doubling the number of menu items from 8 to 16.

INPUT TECHNIQUE also had a significant main effect on ***Success Rate*** ($F_{3,99} = 7.997$, $p < .001$). Tukey HSD post hoc pairwise comparisons show that Device Tilt (92.2%) was significantly worse than Headbang (97.5%) and Touch List (99.2%). Furthermore, the Touch List also had a significantly higher success rate than Touch Pie (95.4%).

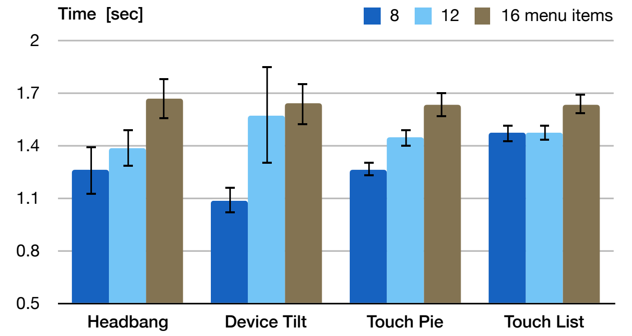

Figure 6: *Time* [sec] by Input Technique × Menu Size. On average it took 1.5sec to open the menu and select an item. Whiskers denote 95% CI.

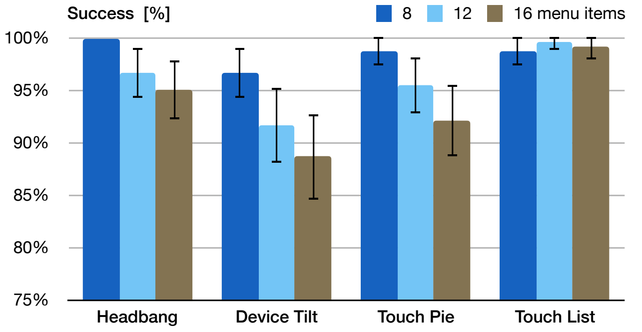

Figure 7: Success Rate [%] by Input Technique × Menu Size. Whiskers denote 95% CI.

6.4 Discussion

Overall, we measured similar times to activate (with an initial tilt or a long press) and select an item across all techniques. That the operation duration became longer with growing menu sizes in any condition had to be expected as the segments in the pie menus become shorter and in the list menu a larger distance has to be traveled with the thumb. Apart from Device Tilt, all input techniques delivered high success rates making them feasible for usage in mobile devices.

In addition to the similar performance, the rankings of our participants were unsettled, too: The Likert scale data from the questionnaires including preference, comfort, and easiness yielded similar ratings across all conditions with no sig. effects found by using a Friedman test. Other than [39] our participants did not find the head controlled input awkward, but rather similarly unconventional to device tilt. This might be a result of Headbang requiring only subtle head rotations for input. The qualitative feedback we received from our participants on what they liked and disliked about the techniques was similar, too. Four participants enjoyed that Headbang requires less homing than Device Tilt, as the head is typically in a more neutral position when initiating the gesture then the wrist. However, three participants also stated that they found reaching the upper targets with Headbang less comfortable than with Device Tilt. Five participants stated that they perceived 16 menu items as too much with all input conditions.

The low success rate measured in the Touch Pie came to a surprise, as it was possible to swipe the finger out of the menu, achieving large target sizes that should have been use reliably. As a conclusion, for touch-only systems we recommend using the Touch List over the Touch Pie.

When using Headbang, items at the bottom and top of the pie menu, which are selected with nodding gestures, were selected around 400 ms faster than items at the left and right, i.e. shaking gesture items. We presume this originates in users looking downwards to the device and thus already performing a rotary pitch in their resting position.

While the effect of feedback was noticeable, the study further supports that Headbang can be used without feedback. The average *Selection Offset* we measured across all trials was 6.59°(SD = 5.08°). While the *Initiation Offset* was four times as large (20.49°) it still offers enough precision for menu sizes of 8.

In conclusion, with its high accuracy and while not being slower as touch, the cost of implementing Headbang as menu technique is quite low. One advantage of Headbang over the Touch List is the reduced amount of screen space needed, thus reducing occluded content. Moreover, all participants were familiar with touch screen menus, but not with head input. Nonetheless was Headbang already slightly faster in small and medium-sized menus than Touch List and we would expect users to become even faster with training.

Therefore we believe that Headbang is a useful menu selection technique in scenarios where users cannot reach over the whole screen, e.g. one-handed smartphone and tablet use, or in-car controls that are operated while driving. It allows for compact menus with many options preserving the context, which is favorable for e.g. drawing and image editing apps.

7 USE CASES

Headbang is an interaction technique that can be used in a wide variety of application domains and use cases. To underscore its utility, we envisioned and developed several use cases and applications we believe to be particularly interesting.

Sharing digital content between different applications is a common task on a smartphone. An example of this is to share an image from the image library with a social media application such as Twitter or Instagram. To do so, users typically have first to select the image, click the share button, and then select the app to which the image should be shared. With Headbang the users select the image and perform a Headbang gesture to directly share the image to the designated application (Fig. 1).

We also developed a simple text editor, shown in Figure 9, that enables the user to use Headbang to trigger actions on selected text or the current text cursor location. For example, to copy selected text, she performance the gesture to the top; to replace the text with another text, she performs the gesture to the bottom. She can also make the text bold, underlined, or italics by using Headbang gestures in different directions.

Headbang gestures can also be used for a multi-level pasteboard in which users can store digital objects by selecting them and then perform a Headbang gesture in the direction in which the object

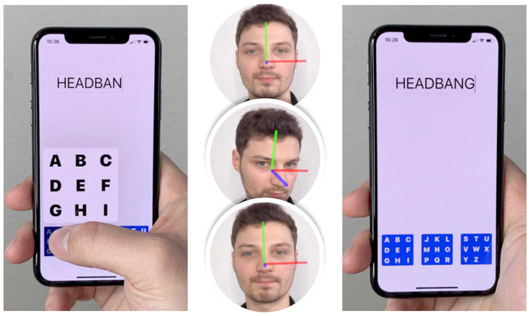

Figure 8: In this example, the user uses an accessibility keyboard for users with tremors who have difficulties to select small buttons. To type the letter G the user touches the left button (left) and then performs a Headbang gesture to the lower left (middle). After lifting the finger from the button the selection is confirmed (right)

Figure 9: In this example, the user selected a part of the text that he wants to cut. After selecting the text with his finger (left) he performs a Handbang gesture to the left (middle) to cut the text (right).

should be stored. This approach allows users to use their spatial memory to retrieve in which direction they stored which object. To retrieve the item, users perform the same Headbang gesture again while placing the finger on an add button or an area at which the object should be placed.

Another use case for the Headbang interaction technique is to use it as accessibility feature for users with tremors who have difficulties typing on an on-screen keyboard [37]. For this use case, we developed an on-screen keyboard that consists of three large buttons that are much easier to select than the typical keyboard buttons (Fig. 8). Each button encodes nine characters, eight for the Headbang gesture and one (the middle) for touching and releasing the button without performing a Headbang gesture. This can be used to type letters selecting one of the buttons and then performing the Headbang gesture in the direction of the designated letter.

We also envision hands-free use cases in which the users perform Headbang gestures to trigger actions that are not applied to a currently selected object but rather global actions. A simple example for this is to use Headbang gestures to turn a page in a digital cookbook while preparing the meal. To execute this gesture, the user has to look directly at the device and then execute the Headbang gesture. The same approach can be used for a variety of different use cases in which the users would like to interact with the device hands-free.

8 CONCLUSION

With Headbang we presented an interaction technique that increases the expressiveness of touch input by using head gestures as an additional input channel. With an overall success rate over 95% our studies showed that the Headbang technique can be reliably detected by a commodity smartphone while the users are sitting or walking. We have shown that the execution time of a head gesture is influenced by the number of menu items, whether the user is walking or standing, and the target location. As Headbang was not slower than touch input in our studies, the costs of using it are low while offering the advantage of needing less screen space and enhancing one-handed uses. This enables Headbang gestures to be used in a variety of everyday tasks to trigger common actions from a menu or without visual feedforward or as an accessibility feature for users who have difficulties selecting small targets on a touch screen. A limitation of Headbang could be that according to Khamis et. al [18] the user's head is not always visible to the front camera of the device while it is used. This could lead to the problem that the device cannot recognize the user's head gestures. However, they also point out that if users are aware that the device needs to track their faces they probably would hold their device differently when they interact with it.

8.1 Future Work

As future work we want to explore this interaction technique in more detail and investigate its use as accessibility feature based on the already presented use cases. An evaluation with users who have tremors will help us understand how the Headbang interaction technique can be further improved. Like most user interfaces, Headbang can also suffer from accidental activation. While it is already possible to discard the menu by swiping to prevent accidental selections, mechanisms to prevent accidental activation remain an interesting area. Moreover, our implementation of Headbang is currently activated by touch input, resulting in a relatively low power consumption. The opportunities Headbang offers to accomplish completely hands-free input are thus not uncovered yet. For a hands-free interaction we want to investigate different activation methods, including dwell-time based head resting, blinking, other facial gestures, and voice input. These approaches come with their own caveats including battery consumption, user acceptance and accuracy issues. We also want to explore the social acceptance of interacting with a smartphone using head gestures in public spaces further, as our qualitative data suggested no dislike for this input modality.

ACKNOWLEDGMENTS

This work was partly funded by the German B-IT Foundation.

REFERENCES

[1] Caroline Appert and Shumin Zhai. 2009. Using Strokes As Command Shortcuts: Cognitive Benefits and Toolkit Support. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 2289–2298. https://doi.org/10.1145/1518701.1519052 Boston, MA, USA.

[2] Mathias Baglioni, Eric Lecolinet, and Yves Guiard. 2011. JerkTilts: Using Accelerometers for Eight-Choice Selection on Mobile Devices. In *Proceedings of the 13th International Conference on Multimodal Interfaces (ICMI '11)*. ACM, New York, NY, USA, 121–128. https://doi.org/10.1145/2070481.2070503 Alicante, Spain.

[3] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. 2012. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 39–48. https://doi.org/10.1145/2371574.2371582 San Francisco, California, USA.

[4] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. 2014. Air+Touch: Interweaving Touch & In-Air Gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 519–525. https://doi.org/10.1145/2642918.2647392 Honolulu, Hawaii, USA.

[5] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. 2019. ForceRay: Extending Thumb Reach via Force Input Stabilizes Device Grip for Mobile Touch Input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 212:1–212:10. https://doi.org/10.1145/3290605.3300442 Glasgow, Scotland UK.

[6] Christian Corsten, Simon Voelker, Andreas Link, and Jan Borchers. 2018. Use the Force Picker, Luke: Space-Efficient Value Input on Force-Sensitive Mobile Touchscreens. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 661:1–661:12. https://doi.org/10.1145/3173574.3174235 Montreal, Canada.

[7] D. A. Craig and H. T. Nguyen. 2005. Wireless Real-Time Head Movement System Using a Personal Digital Assistant (PDA) for Control of a Power Wheelchair. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE, New York, NY, USA, 772–775. https://doi.org/10.1109/IEMBS.2005.1616529

[8] Andrew Crossan, Mark McGill, Stephen Brewster, and Roderick Murray-Smith. 2009. Head Tilting for Interaction in Mobile Contexts. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, 6:1–6:10. https://doi.org/10.1145/1613858.1613866 Bonn, Germany.

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

[10] Yulia Gizatdinova, Oleg Špakov, Outi Tuisku, Matthew Turk, and Veikko Surakka. 2018. Gaze and Head Pointing for Hands-Free Text Entry: Applicability to Ultra-Small Virtual Keyboards. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) *(ETRA '18)*. Association for Computing Machinery, New York, NY, USA, Article 14, 9 pages. https://doi.org/10.1145/3204493.3204539

[11] Dmitry O. Gorodnichy and Gerhard Roth. 2004. Nouse 'Use your Nose as a Mouse' Perceptual Vision Technology for Hands-Free Games and Interfaces. *Image and Vision Computing* 22, 12 (Oct. 2004), 931–942. https://doi.org/10.1016/j.imavis.2004.03.021

[12] Uta Hinrichs and Sheelagh Carpendale. 2011. Gestures in the Wild: Studying Multi-Touch Gesture Sequences on Interactive Tabletop Exhibits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 3023–3032. https://doi.org/10.1145/1978942.1979391 Vancouver, BC, Canada.

[13] Nambu Hirotaka. 2003. Reassessing Current Cell Phone Designs: Using Thumb Input Effectively. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 938–939. https://doi.org/10.1145/765891.766081 Ft. Lauderdale, Florida, USA.

[14] Thibaut Jacob, Gilles Bailly, Eric Lecolinet, Géry Casiez, and Marc Teyssier. 2016. Desktop Orbital Camera Motions Using Rotational Head Movements. In *Proceedings of the 2016 Symposium on Spatial User Interaction (SUI '16)*. ACM, New York, NY, USA, 139–148. https://doi.org/10.1145/2983310.2985758 Tokyo, Japan.

[15] Amy K. Karlson, Benjamin B. Bederson, and Jose L. Contreras-Vidal. 2008. Understanding One-Handed Use of Mobile Devices. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology* 1, 1 (2008), 86–101. https://doi.org/10.4018/978-1-59904-871-0.ch006

[16] Viktoria A. Kettner and Jeremy I. M. Carpendale. 2013. Developing Gestures for No and Yes: Head Shaking and Nodding in Infancy. *Gesture* 13, 2 (Jan. 2013), 193–209. https://doi.org/10.1075/gest.13.2.04ket

[17] Mohamed Khamis, Florian Alt, and Andreas Bulling. 2018. The Past, Present, and Future of Gaze-Enabled Handheld Mobile Devices: Survey and Lessons Learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI '18*. ACM, New York, NY, USA, 1–17. https://doi.org/10.1145/3229434.3229452 Barcelona, Spain.

[18] Mohamed Khamis, Anita Baier, Niels Henze, Florian Alt, and Andreas Bulling. 2018. Understanding Face and Eye Visibility in Front-Facing Cameras of Smartphones Used in the Wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 280:1–280:12. https://doi.org/10.1145/3173574.3173854 Montreal QC, Canada.

[19] 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, 81:1–81:14. https://doi.org/10.1145/3173574.3173655 Montreal QC, Canada.

[20] Yang Li. 2010. Gesture Search: A Tool for Fast Mobile Data Access. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, New York, NY, USA, 87–96. https://doi.org/10.1145/1866029.1866044 New York, New York, USA.

[21] 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 Arlington, Virginia, USA.

[22] Tao Lu, Kui Yuan, Huosheng H. Hu, and Pei Jia. 2007. Head Gesture Recognition for Hands-Free Control of an Intelligent Wheelchair. *Industrial Robot: The International Journal of Robotics Research and Application* 34, 1 (Jan. 2007), 60–68. https://doi.org/10.1108/01439910710718469

[23] Diako Mardanbegi, Dan Witzner Hansen, and Thomas Pederson. 2012. Eye-based Head Gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 139–146. https://doi.org/10.1145/2168556.2168578 event-place: Santa Barbara, California.

[24] Evelyn Z. McClave. 2000. Linguistic Functions of Head Movements in the Context of Speech. *Journal of Pragmatics* 32, 7 (June 2000), 855–878. https://doi.org/10.1016/S0378-2166(99)00079-X

[25] 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 Sydney, Australia.

[26] 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, 43:1–43:8. https://doi.org/10.1145/2971485.2971490 Gothenburg, Sweden.

[27] I. Oakley and S. O'Modhrain. 2005. Tilt to scroll: evaluating a motion based vibrotactile mobile interface. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. IEEE, New York, NY, USA, 40–49. https://doi.org/10.1109/WHC.2005.138

[28] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: Accelerometer-Supported Text Entry for Very Small Devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (Paris, France) *(UIST '02)*. Association for Computing Machinery, New York, NY, USA, 201–204. https://doi.org/10.1145/571985.572013

[29] Robert G. Radwin, Gregg C. Vanderheiden, and Mei-Li Lin. 1990. A Method for Evaluating Head-Controlled Computer Input Devices Using Fitts' Law. *Human Factors* 32, 4 (Aug. 1990), 423–438. https://doi.org/10.1177/001872089003200405

[30] Jun Rekimoto. 1996. Tilting Operations for Small Screen Interfaces. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* (Seattle, Washington, USA) *(UIST '96)*. Association for Computing Machinery, New York, NY, USA, 167–168. https://doi.org/10.1145/237091.237115

[31] H. H. Sad and F. Poirier. 2009. Evaluation and Modeling of User Performance for Pointing and Scrolling Tasks on Handheld Devices Using Tilt Sensor. In *2009 Second International Conferences on Advances in Computer-Human Interactions*. IEEE, New York, NY, USA, 295–300. https://doi.org/10.1109/ACHI.2009.15

[32] Joey Scarr, Andy Cockburn, and Carl Gutwin. 2013. Supporting and Exploiting Spatial Memory in User Interfaces. *Foundations and Trends® in Human–Computer Interaction* 6, 1 (2013), 1–84. https://doi.org/10.1561/1100000046

[33] Zhenyu Tang, Chenyu Yan, Sijie Ren, and Huagen Wan. 2017. HeadPager: Page Turning with Computer Vision Based Head Interaction. In *Computer Vision – ACCV 2016 Workshops (Lecture Notes in Computer Science)*, Chu-Song Chen, Jiwen Lu, and Kai-Kuang Ma (Eds.). Springer International Publishing, Cham, Switzerland, 249–257.

[34] Robert J. Teather and I. Scott MacKenzie. 2014. Position vs. Velocity Control for Tilt-Based Interaction. In *Proceedings of Graphics Interface 2014* (Montreal, Quebec, Canada) *(GI '14)*. Canadian Information Processing Society, CAN, 51–58.

[35] Feng Tian, Lishuang Xu, Hongan Wang, Xiaolong Zhang, Yuanyuan Liu, Vidya Setlur, and Guozhong Dai. 2008. Tilt Menu: Using the 3D Orientation Information of Pen Devices to Extend the Selection Capability of Pen-Based User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1371–1380. https://doi.org/10.1145/1357054.1357269

[36] Simon Voelker, Sebastian Hueber, Christian Corsten, and Christian Remy. 2020. HeadReach: Using Head Tracking to Increase Reachability on Mobile Touch

Devices. In *To appear in the Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu Hawai'i, USA) *(CHI '20)*. ACM, New York, NY, USA. https://doi.org/10.1145/3313831.3376868

[37] Chat Wacharamanotham, Jan Hurtmanns, Alexander Mertens, Martin Kronenbuerger, Christopher Schlick, and Jan Borchers. 2011. Evaluating Swabbing: A Touchscreen Input Method for Elderly Users with Tremor. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 623–626. https://doi.org/10.1145/1978942.1979031 Vancouver, BC, Canada.

[38] Daniel Wigdor and Ravin Balakrishnan. 2003. TiltText: Using Tilt for Text Input to Mobile Phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (Vancouver, Canada) *(UIST '03)*. Association for Computing Machinery, New York, NY, USA, 81–90. https://doi.org/10.1145/964696.964705

[39] Julie R. Williamson, Stephen Brewster, and Rama Vennelakanti. 2013. Mo!Games: Evaluating Mobile Gestures in the Wild. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction (ICMI '13)*. ACM, New York, NY, USA, 173–180. https://doi.org/10.1145/2522848.2522874 Sydney, Australia.

[40] Graham Wilson, Stephen A. Brewster, Martin Halvey, Andrew Crossan, and Craig Stewart. 2011. The Effects of Walking, Feedback and Control Method on Pressure-based Interaction. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 147–156. https://doi.org/10.1145/2037373.2037397 event-place: Stockholm, Sweden.

[41] Yukang Yan, Chun Yu, Xin Yi, and Yuanchun Shi. 2018. HeadGesture: Hands-Free Input Approach Leveraging Head Movements for HMD Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable, and Ubiquitous Technologies (IMWUT '18)* 2, 4 (Dec. 2018), 198:1–198:23. https://doi.org/10.1145/3287076

[42] Shanhe Yi, Zhengrui Qin, Ed Novak, Yafeng Yin, and Qun Li. 2016. GlassGesture: Exploring Head Gesture Interface of Smart Glasses. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. IEEE, New York, NY, USA, 1–9. https://doi.org/10.1109/INFOCOM.2016.7524542

[43] Chi Zhang, Nan Jiang, and Feng Tian. 2016. Accessing Mobile Apps with User Defined Gesture Shortcuts: An Exploratory Study. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 385–390. https://doi.org/10.1145/2992154.2996786 Niagara Falls, Ontario, Canada.

[44] Oleg Špakov, Poika Isokoski, Jari Kangas, Jussi Rantala, Deepak Akkil, and Roope Raisamo. 2016. Comparison of Three Implementations of HeadTurn: A Multimodal Interaction Technique with Gaze and Head Turns. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction (ICMI '16)*. ACM, New York, NY, USA, 289–296. https://doi.org/10.1145/2993148.2993153 Tokyo, Japan.