GazeSwipe: Enhancing Mobile Touchscreen Reachability through Seamless Gaze and Finger-Swipe Integration

Zhuojiang Cai*
State Key Lab. of VR Technology and Systems
Beihang University
Beijing, China
caizhuojiang@buaa.edu.cn

Jingkai Hong*
State Key Lab. of VR Technology and Systems
Beihang University
Beijing, China
jingkai_hong@buaa.edu.cn

Zhimin Wang
State Key Lab. of VR Technology and Systems
Beihang University
Beijing, China
zm.wang@buaa.edu.cn

Feng Lu†
State Key Lab. of VR Technology and Systems
Beihang University
Beijing, China
lufeng@buaa.edu.cn
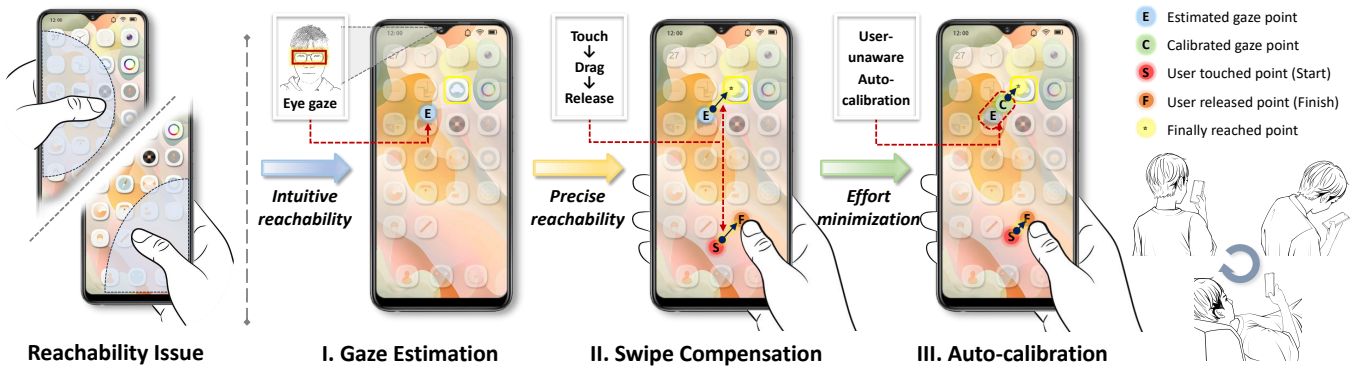
Figure 1: To address the issue of thumb reachability on smartphones and tablets when held with one hand, we introduce the gaze-based pointing, enabling users to intuitively reach any position by simply gazing at the target. By combining this with the finger-touch input, users can precisely interact with targets through swipe gestures. Additionally, we introduce a user-unaware auto-calibration method that eliminates the need for explicit gaze calibration, enhancing gaze accuracy during use and making interactions more seamless and efficient.

Abstract

Smartphones with large screens provide users with increased display and interaction space but pose challenges in reaching certain areas with the thumb when using the device with one hand. To address this, we introduce GazeSwipe, a multimodal interaction technique that combines eye gaze with finger-swipe gestures, enabling intuitive and low-friction reach on mobile touchscreens. Specifically, we design a gaze estimation method that eliminates the need for explicit gaze calibration. Our approach also avoids the use of additional eye-tracking hardware by leveraging the smartphone's built-in front-facing camera. Considering the potential decrease in gaze accuracy without dedicated eye trackers, we use finger-swipe gestures to compensate for any inaccuracies in gaze estimation. Additionally, we introduce a user-unaware auto-calibration method that improves gaze accuracy during interaction. Through extensive experiments on smartphones and tablets, we compare our technique with various methods for touchscreen reachability and evaluate the performance of our auto-calibration strategy. The results demonstrate that our method achieves high success rates and is preferred by users. The findings also validate the effectiveness of the auto-calibration strategy.

*Both authors contributed equally to this research.

†Feng Lu is the corresponding author.

CCS Concepts

• **Human-centered computing → HCI design and evaluation methods**; **Interaction techniques**.

Keywords

Interaction Technique, Eye Tracking, Reachability, Mobile Devices

1 Introduction

In recent years, the rapid advancement of smartphone and tablet technology has been characterized by a significant increase in screen sizes. While these larger displays have undoubtedly enriched user experiences by providing more space for content presentation and interaction, they have also introduced a series of ergonomic challenges, particularly concerning one-handed usage [8, 15, 21]. As screen sizes approach or exceed the 5.5-inch threshold, users find it increasingly difficult to reach certain areas of the screen with one hand [52]. This often requires frequent adjustments to the grip, making operations more cumbersome and time-consuming, and increasing the risk of accidental drops.

Imagine you are holding your smartphone with one hand while the other is occupied by carrying a bag, holding onto a bus or subway handrail, or steering a shopping cart, and you need to tap a button at the top of the screen. Alternatively, picture yourself walking through the office, one hand holding a coffee cup while the other operates a tablet, or even writing on a whiteboard with one hand while trying to use a tablet with the other. In these situations, reaching distant parts of the screen becomes difficult and unstable. These scenarios highlight the reachability issues of large touchscreens when used with one hand, emphasizing the need for a more convenient and efficient solution.

To address the reachability issue, several solutions have been proposed. Screen Transform Techniques, such as Sliding Screen [21], TiltReduction [3], and One-handed Mode features on various smartphones [12], adjust the user interface by scaling or shifting it to the lower areas of the screen, bringing content within the thumb's comfortable reach. While these methods can enhance reachability and improve grip stability, they come at the cost of reducing the usable screen area. Scaling the interface often results in smaller elements, leading to imprecise operations, while shifting the interface to display only a part of the content makes it inconvenient for continuous operations across different regions of the screen. Moreover, since the thumb's reachable area remains fixed regardless of screen size, the required degree of scaling or shifting increases with larger displays, making these techniques impractical for tablets.

Another category of solutions, Cursor-based Techniques, such as BezelCursor [29], ForceRay [5], and HeadReach [52], enable users to control a virtual cursor to reach targets anywhere on the screen via touch. While these strategies avoid the limitations of Screen Transform Techniques, their interaction experiences are often less intuitive and require long thumb movements, making the interaction more effortful and time-consuming.

Given that gaze information naturally and intuitively reflects user intent, increasing research is integrating it into mobile device interactions [7, 18, 32]. For example, gaze has been used to enhance mobile voice assistants, improve text editing efficiency, and facilitate cross-device collaboration [31, 44, 53]. CursorShift [39] achieved gaze and touch interaction on tablets in a controlled lab environment. However, this method requires expensive external eye-tracking devices and additional explicit calibration processes, making it impractical for everyday use on mobile devices and insufficient for effectively solving reachability problems. Recently, Apple has also introduced gaze tracking on iPhones and iPads as an accessibility feature for people with disabilities, allowing users to gaze at a target and dwell to confirm interactions [2]. However, this feature requires calibration before use, and after calibration, changes in head position may lead to inaccurate performance. Moreover, the reliance on gaze alone for unimodal interaction makes it susceptible to the Midas touch problem. Therefore, it cannot serve as a viable solution for the everyday reachability issue.

To this end, we propose GazeSwipe, a multimodal interaction technique that combines eye gaze with finger-swipe gestures to enhance reachability on smartphones and tablets in an intuitive and low-effort manner. With GazeSwipe, users simply gaze at the target and use a thumb swipe to adjust the gaze cursor and complete the interaction. Our approach offers three key advantages: 1) it requires no additional hardware, utilizing the front-facing camera for gaze estimation; 2) it employs a user-unaware auto-calibration strategy, eliminating the need for explicit gaze calibration and improving gaze accuracy during use; and 3) Since gaze is fast and intuitive, while finger-swipe is subtle and effortless, the combination delivers a seamless and low-friction interaction experience for everyday use on smartphones and tablets.

We conduct two user studies to evaluate our interaction technique. In the first study, we examine the impact of different calibration strategies on GazeSwipe. The results show that our auto-calibration strategy significantly improves gaze estimation accuracy while eliminating the annoying explicit calibration process and enhancing robustness to head movements in everyday use. In the second study, we compared GazeSwipe with other interaction techniques addressing reachability issues on smartphones and tablets. The results demonstrate that our method achieves high success rates and is preferred by users on both devices. Specifically on tablets, our method outperforms others across all subjective and objective metrics, except for a moderately longer completion time.

In summary, this paper makes the following contributions:

- We propose GazeSwipe, a novel multimodal interaction technique that integrates eye gaze with finger swipes, providing an intuitive and low-friction method for enhancing reachability on smartphones and tablets.
- We propose a user-unaware auto-calibration method that eliminates the need for explicit gaze calibration while enhancing gaze accuracy during use, thereby enabling smooth interactions.
- We conducted extensive user studies to evaluate the auto-calibration strategy and compare GazeSwipe with other reachability techniques, demonstrating its promising performance on both smartphones and tablets.

2 Related Work

2.1 Reachability Techniques on Mobile Devices

As smartphones continue to increase in screen size, there is a growing challenge of thumb reachability to certain areas of the screen while holding the device with one hand. To address this issue and

enable users to interact with targets in these unreachable areas more conveniently, researchers have explored various reachability techniques. We categorize these techniques into screen transform techniques and cursor techniques for discussion.

2.1.1 Screen Transform Techniques. The screen transform technique involves resizing or moving screen regions to bring unreachable areas closer to the thumb. Many smartphones already implement such features, such as One-handed Mode on Android devices [12]. This mode is activated by swiping up from one of the bottom corners, causing the interface to shrink towards that corner, thereby allowing the thumb to reach previously unreachable areas. Similarly, MovingScreen [48] and Sliding Screen [21] utilize edge swipes to to move the screen closer to the thumb. ThumbSpace [16] creates a pop-up view at the thumb touch position to display the entire screen, while TapTap [42] shows a portion of the screen near the thumb touch position in the pop-up view. Chang et al. [3] proposed tilt-based solutions for reachability issues, including TiltSlide, which shifts the screen towards the tilt direction, and TiltReduction, which reduces the screen size when the device is tilted. Le et al. [26] suggested moving the interface by sliding the index finger on a touchpad located on the back of the device.

However, these methods often shrink or partially display the interface, making it harder to accurately tap targets and affecting display visibility. Tilt-controlled approaches may be impractical in certain postures, such as when lying down. Additionally, methods using back touchpads require additional hardware.

2.1.2 Cursor Techniques. Another approach, besides moving the screen closer to the thumb, is to control the cursor to interact with unreachable screen areas. BezelCursor [29] and ExtendedThumb [25] trigger cursor mode through bezel swipes or double-taps, enabling users to drag the cursor to the desired target. TiltCursor [3] triggers cursor mode when the device is tilted. Ikeda et al. [14] proposed a hover-based method for cursor manipulation. CornerSpace [57] permits users to position a remote cursor in unreachable screen zones for swift access. ForceRay [5] utilizes the force-sensitive screen to adjust cursor distance from the thumb based on pressure intensity. Dual-Surface Input [56] employs an extra touchpad on the device's back to select targets beyond the thumb's reach. HeadReach [52] enables cursor control via head movements. Esteves et al. [9] allow users to select predefined targets outside the screen by specifying sliding trajectories on the screen.

Although these cursor-based techniques can help users accurately reach targets, the long thumb movement distances may lead to user fatigue. Additionally, the use of force input, head posture, and custom sliding trajectories to assist in addressing reachability issues may not be entirely intuitive, requiring a learning curve.

2.2 Gaze Estimation Techniques

Gaze estimation technique involves predicting the point of gaze (PoG) or gaze direction from images or videos captured by a camera. These methods can be categorized into model-based and appearance-based techniques.

2.2.1 Model-based Techniques. Model-based techniques estimate gaze by fitting a 3D eye model from images. Some methods utilize specialized RGB-D cameras [11, 47, 55] or infrared cameras [34]

to obtain 3D facial landmarks, 3D pupil location, iris contour, and corneal reflection, enabling more accurate fitting of the eye model. However, these methods are not suitable for use on mobile devices due to the requirement for additional equipment.

Other methods utilize regular RGB cameras [30, 49, 50, 54] to capture images for fitting the eye model. They use detected 2D facial landmarks to fit a 3D face model to obtain head pose and detect the 2D pupil center [49] or iris contour [30]. Wang et al. [54] proposed a deformable eye-face model, while Ververas et al. [50] proposed using neural network regression to predict dense 3D eye meshes, enhancing estimation robustness. Nevertheless, these methods are generally less robust in unconstrained environments.

2.2.2 Appearance-based Techniques. Appearance-based techniques directly map appearance images to 2D gaze points or 3D gaze directions, making them suitable for use on mobile devices. For estimating 2D gaze points, Krafka et al. [23] introduced a mobile device-compatible gaze estimation technique, utilizing the GazeCapture dataset to train a convolutional neural network. This network takes cropped eye and face images as input, eliminating the need for additional head pose information. Bao et al. [1] enhanced gaze estimation accuracy by adaptively integrating binocular features and guiding eye feature extraction with facial cues. Huynh et al. [13] proposed iMon, a mobile-specific gaze tracking system, which significantly improves estimation precision by addressing gaze uncertainty due to minor eye movements, refining eye images to reduce motion blur, and correcting individual focus differences. Park et al. [36] proposed a method integrating visual saliency to address accuracy challenges in traditional eye-tracking caused by individual differences and data scarcity.

For estimating 3D gaze directions, several methods have been proposed. Fischer et al. [10] introduced RT-GENE, a real-time 3D gaze estimation system designed for natural environments, leveraging advanced image processing and feature extraction techniques. Kellnhofer et al. [17] presented Gaze360, which combines a 360° panoramic camera setup with multiple SLR cameras to collect comprehensive data for accurate 3D gaze direction estimation. Zhang et al. [58] developed ETH-XGaze, a large-scale dataset and accompanying techniques tailored for precise 3D gaze estimation under extreme head pose and gaze variations. Cheng et al. [4] proposed GazeTR, one of the first methods to adopt the vision transformer architecture for gaze estimation, enhancing accuracy through a hybrid approach that integrates CNN feature extraction. Nagpure et al. [33] introduced GazeNAS, employing neural architecture search to create lightweight and efficient 3D gaze estimation models.

Despite significant progress, appearance-based techniques still cannot provide precise gaze estimation for fine-grained interaction on mobile devices in unconstrained environments. In our work, we employ a lightweight 2D gaze estimation method that enables direct usage on mobile devices without requiring gaze point projection.

2.3 Gaze-based Interaction Techniques

In this paper, we propose a gaze-based dual-modal interaction technique that combines gaze and touch inputs on mobile touchscreens. This section reviews gaze-only and gaze-based multi-modal interaction methods, with a particular focus on those on mobile devices.

2.3.1 Gaze-only Interaction. Gaze-only interaction on computer displays and mobile touchscreens has been explored in many studies. Previous work has classified gaze-based interactions into two categories: implicit and explicit [27, 35]. Implicit interactions trigger actions based on the user's passive gaze behavior, such as triggering page scrolling by analyzing gaze trajectories during reading [28]. Explicit interactions, more relevant to our work, rely on intentional gaze movements to trigger actions. Common techniques in explicit gaze interaction include dwell time, pursuit, and gaze gestures [6, 35, 51]. For instance, Apple's eye-tracking accessibility feature on iOS uses dwell time to trigger interactions [2].

Although gaze-only systems hold significant potential, they face critical challenges. One major issue is the Midas touch problem, where unintended gaze fixations inadvertently trigger actions on the screen, leading to false inputs. In addition, some methods may cause user fatigue, making them less suitable for prolonged use.

2.3.2 Gaze + Voice / Motion / Mouse. Other studies have explored combining gaze with other modalities for interaction on mobile touchscreens [27]. Mayer et al. [31] introduced WorldGaze, a pioneering technology aimed at enhancing mobile voice assistants through gaze interaction. Zhao et al. [60] proposed a hands-free text correction method that combines gaze and voice on smartphones. Kong et al. [22] utilized both gaze and hand motion for controlling smartphone interfaces. These studies collectively underscore the crucial role of user visual attention information provided by gaze in mobile device interactions.

Some gaze and motion-based interaction techniques in XR employ auto-calibration methods. Kytö et al. [24] used gaze in AR headsets, refining interactions through head or hand movements and collecting temporal interaction samples for calibration. Similarly, Sidenmark et al. [43] explored gaze-hand coordination calibration strategies in VR headsets. While these strategies have proven successful in XR, they have not been applied to mobile touchscreen scenarios, where optimizations such as considering changes in head posture are important.

Additionally, Sugano et al. [46] explored gaze estimation with auto-calibration in desktop PC environments, considering head movement. While there are some similarities with our approach, key differences arise from the different devices used (smartphones and tablets vs. desktop PCs). Additionally, our auto-calibration strategy collects data from finger swipe release positions (vs. from mouse clicks [46]), which is specifically tailored to our interaction method. This combination of gaze, finger swipe, and auto-calibration ensures smooth and precise interactions for everyday use.

2.3.3 Gaze + Touch. The integration of gaze and touch has been increasingly explored in recent research. Rivu et al. [40] proposed a method for text editing that combines gaze and touch. Pfeuffer et al. [38] explored gaze and touch interaction for multi-user touchscreen collaboration. Voelker et al. [53] utilized gaze and touch for cross-device collaboration. Rivu et al. [41] introduced an enhanced button interaction method based on gaze. Some works [19, 20] employ the combination of gaze and touch for authentication on mobile devices.

While these studies cover a variety of applications, some research has specifically explored the use of gaze with touch swipes for refinement across different devices. Stellmach et al. [45] proposed a method for controlling a distant display using gaze and a handheld touchscreen. Pfeuffer et al. [37] demonstrated interactions on tablets using a combination of pen, touch, and gaze. The work most closely related to ours is CursorShift [39], which uses a Tobii EyeX tracker for gaze estimation and combines it with touch input on tablet devices. However, these approaches focus on different contexts from ours. While they demonstrate how gaze and touch can collaborate for interaction, the need for a large and fixed eye tracker, along with additional calibration processes, makes these methods impractical for smartphones and tablets in everyday mobile scenarios and for addressing reachability challenges.

3 Motivation and Challenges

Given the limitations of Screen Transform Techniques—such as wasted screen space, imprecise operations, and difficulty in extending to tablets—a more efficient and low-friction Cursor-based Technique might be a better reachability solution on both smartphones and tablets. Based on this motivation, we aim to develop a gaze cursor technique that intuitively and efficiently overcomes the reachability issue. However, this technique still faces the following three challenges:

Q1: Is it possible to implement gaze cursor interaction without additional hardware or a calibration process? Previous work has attempted to implement gaze and touch interaction on tablets, but it relied on external, expensive eye trackers and required a cumbersome 9-point calibration process [39]. This limits its practicality for everyday mobile use and its ability to effectively address the reachability issue. Can we instead leverage the front-facing camera of a mobile device, without requiring calibration, to provide an effortless and effective reachability solution?

Q2: If there is no calibration process, how can the gaze cursor be made more accurate and stable? The gaze estimation results are visualized as a gaze cursor, and its accuracy directly affects the distance the user needs to adjust the cursor. Without additional calibration processes and relying solely on the front-facing camera, the accuracy of gaze estimation is typically low. So, how can we improve the accuracy and stability of the gaze estimation results?

Q3: How can we design a gaze cursor-based interaction method that is natural and seamless for users? Even with professional external eye trackers, gaze cursors often lack the precision required for fine-grained interactions. Thus, despite improvements in gaze estimation accuracy and stability using the front-facing camera, users may still need to adjust the cursor. Given this challenge, how can we develop a smooth and user-friendly interaction method?

4 GazeSwipe

Building on the motivation and challenges outlined above, we propose GazeSwipe, an intuitive and low-effort gaze-and-swipe interaction technique that requires no calibration and can be used in everyday scenarios. Our approach is presented in three parts: mobile gaze estimation, user-unaware auto-calibration, and gaze-and-swipe interaction.

4.1 Mobile Gaze Estimation

GazeSwipe relies on the gaze estimation method to estimate the user's real-time gaze point on the screen, enabling quick and accurate pointing of the intended interaction target. To achieve a
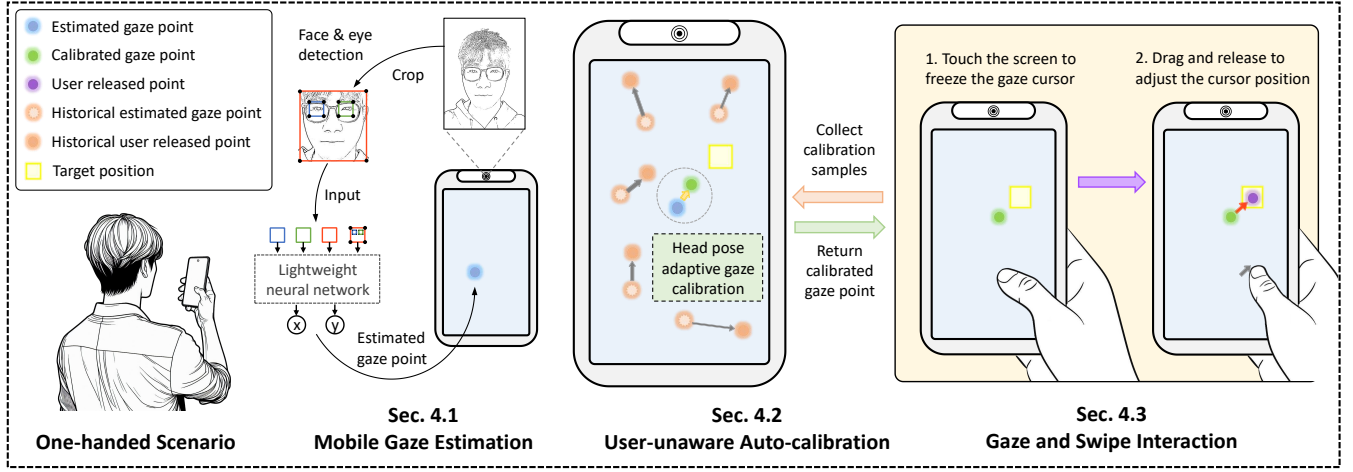
Figure 2: Overview of the GazeSwipe method. Our approach is designed to tackle the reachability challenges encountered in one-handed mobile touchscreen usage scenarios. It involves feeding facial images captured by the front-facing camera into a lightweight neural network to estimate the gaze point and display a cursor accordingly. Additionally, the system performs user-unaware auto-calibration based on historical interaction samples, improving gaze accuracy over time. To complete interactions, users gaze at the target and fine-tune the gaze cursor position using a "touch-drag-release" approach.

plug-and-play functionality without requiring calibration, we employ a deep learning approach that estimates the gaze point directly from facial images captured by the device's front-facing camera.

Specifically, our approach employs a lightweight neural network [1] to estimate the gaze point coordinates on the screen plane based on RGB images, with the camera as the origin and distances measured in centimeters. This neural network operates in real-time on most mainstream smartphones and achieves competitive performance on the GazeCapture [23] and MPIIFaceGaze [59] datasets, a gaze error of 3.9 cm on the MPIIFaceGaze dataset. On our experimental devices, the accuracy is 2.9 cm for the phone and 3.5 cm for the tablet, with an average frame rate of 10-15 fps.

The gaze point is then transformed into screen coordinates according to the camera and screen parameters, and a one-euro filter is applied to eliminate gaze jitter. Let $\mathcal{I}$ represent the image captured by the front camera, and G_E represent the estimated gaze point in the screen coordinate system. Our gaze estimation process can be formulated as:

$$G_E = H(T \cdot f_\theta(\mathcal{I})) \tag{1}$$

where $f_\theta(\cdot)$ is the neural network, and θ represents its parameters. The transformation matrix T converts the coordinates from the camera-origin system to the screen-origin system, and $H(\cdot)$ represents the filter used to smooth the estimated gaze points.

However, the estimated gaze point alone is insufficient for precise pointing of fine-grained interaction elements. Therefore, we introduce an auto-calibration method to further enhance gaze accuracy, which is detailed in Section 4.2, and an interaction method that combines gaze with swipe gestures, described in Section 4.3.

4.2 User-Unaware Auto-Calibration

Existing gaze tracking techniques typically require individual calibration before use. In conventional gaze interaction techniques,

an explicit 9-point calibration strategy is one of the mainstream methods. Users are required to sequentially gaze at 9 target points displayed on the screen for a few seconds each before using the device. This process captures eye parameters or adjusts the mapping relationship to improve the accuracy of the gaze estimation. However, this explicit calibration process is cumbersome and significantly reduces the user experience. Furthermore, in mobile touchscreen use cases, the user may frequently move, causing them to gradually deviate from the initial calibration environment during subsequent use. These deviations may be due to changes in head posture, variations in the relative position between the face and the camera, or changes in lighting conditions. As a result, calibration information may become invalid, leading to increased gaze estimation errors over time. While recalibration can mitigate these deviations, frequent calibration is impractical in daily use.

To overcome the limitations of explicit gaze calibration strategies on mobile touchscreens, we propose a user-unaware auto-calibration method. This method is based on the insight that the interaction process itself can serve as the calibration process, enabling implicit, dynamic, and incremental gaze calibration during interactions. Our auto-calibration method, similar to explicit calibration, requires gathering samples of estimated gaze points and their corresponding ground truth values. We observed that the user's interaction behavior provides calibration samples, as the location of the cursor at the moment the thumb is released can be regarded as the ground truth for the gaze point. Therefore, the user-unaware auto-calibration strategy continuously collects calibration data during interactions and dynamically adjusts the gaze estimation based on historical interaction samples.

Specifically, each time the user releases their thumb after completing an interaction, a calibration sample is recorded. Each calibration sample represents the position of the cursor at the moment the thumb is released, that is, the ground truth of the gaze point

G_{gt}^i, and the gaze point estimated by the neural network G_E^i. The user-unaware auto-calibration aims to correct the current estimated gaze point G_E^t based on these samples, reducing the distance the user needs to drag and improving the interaction experience.

To achieve this, we initially designed an efficient auto-calibration strategy (Strategy 1) that fulfills our goals. Subsequently, we considered additional factors specific to mobile scenarios to further enhance the performance of the auto-calibration, leading to Strategy 2. The user study is discussed in detail in Section 6.

4.2.1 Calibration Strategy 1. In the t-th interaction, our strategy corrects the current estimated gaze point G_E^t by applying distance-inverse weighting to the offsets of historical samples $\{dG^i | i = 1, 2, \ldots, t-1\}$, resulting in the calibrated gaze point G_C^t:

$$G_C^t = G_E^t + \sum_{i=1}^{t-1} \lambda_t^i dG^i, \qquad (2)$$

where dG^i is the offset vector of the i-th sample, and λ_t^i is the distance-inverse weight of the i-th sample relative to the current prediction. These terms are defined as follows:

$$\begin{cases} dG^i = G_{gt}^i - G_E^i, \\ \lambda_t^i = \dfrac{1}{S_t \| G_{gt}^i - G_E^i \|}. \end{cases} \qquad (3)$$

Here, $\| G_{gt}^i - G_E^i \|$ represents the Euclidean distance between two points, and $S_t = \sum_{j=1}^{t-1} \frac{1}{\| G_{gt}^j - G_E^j \|}$. This auto-calibration strategy effectively corrects the biases in the gaze estimation output by the network, which are influenced by user-specific and environment-specific factors.

4.2.2 Calibration Strategy 2. Head movements often affect the accuracy of gaze calibration. Therefore, we account for this factor in our auto-calibration strategy. Specifically, the user's estimated head pose relative to the camera is represented by a normalized direction vector H_E^i. This information is obtained during the preprocessing stage of gaze estimation and retained in the calibration samples.

The calibrated gaze point G_C^t is determined with an additional weight h_t^i based on the head pose, and is expressed as:

$$G_C^t = G_E^t + \sum_{i=1}^{t-1} \lambda_t^i h_t^i dG^i, \qquad (4)$$

where $h_t^i = H_E^t \cdot H_E^i$ represents the cosine of the angle between the current head pose direction and the head pose direction of the i-th sample.

This calibration strategy is more robust to head movements relative to the phone, as demonstrated in our subsequent studies. Ultimately, our interaction technique adopts this strategy for user-unaware auto-calibration, reducing the average drag distance in GazeSwipe, thereby improving the user experience.

Our calibration strategies post-process the neural network output, weighting the results based on historical gaze points and head pose. This process does not involve updating the neural network's parameters. We found that biases caused by factors such as head pose changes are likely low-dimensional. Considering the real-time

constraints of resource-limited mobile devices, post-processing alone delivers satisfactory performance.

4.3 Gaze and Swipe Interaction

GazeSwipe leverages gaze for pointing at target elements, allowing the cursor to intuitively and quickly approach the target. It's important to note that previous studies have found that even with professional external eye trackers and additional calibration steps, gaze precision is not always sufficient for precise target selection. Therefore, manual adjustment of the cursor to fine-tune its position is necessary.

Our approach combines gaze with finger-swipe gestures, enabling users to control the gaze cursor through a seamless "touch-drag-release" motion. GazeSwipe displays the gaze cursor, which snaps to the nearest element in real-time on the screen. Users can tap anywhere on the screen with their thumb to lock the cursor, drag their thumb to adjust the cursor toward the target, and release it at the intended location to confirm the interaction. Additionally, if the gaze cursor is already accurately snapped to the target without requiring a swipe, a simple tap can confirm the interaction.

Efforts have been made to ensure GazeSwipe is smooth and user-friendly. Since gaze estimation errors increase with swipe distance, larger errors can make the operation more difficult and time-consuming. Therefore, incorporating gaze calibration methods to reduce estimation errors is essential. Our auto-calibration approach not only addresses this issue but also eliminates the need for explicit calibration processes. Moreover, the gaze cursor analyzes the layout and automatically snaps to the nearest element, bringing it closer to the target. These strategies make interactions more efficient and effortless.

GazeSwipe is activated and deactivated using a custom gesture. In our implementation, users can activate GazeSwipe by double-tapping the screen edge, which initiates gaze estimation and displays the gaze cursor. The same gesture can be used to deactivate it and return to the standard touchscreen mode.

To enhance the interaction experience and facilitate continuous operations, GazeSwipe remains compatible with standard touch-screen gestures while activated. Specifically, users can double-tap to trigger a short tap at the finger position or use a long swipe to scroll the page. This design emphasizes GazeSwipe's ability to facilitate smooth interactions across different regions of the screen.

5 Evaluation Design

5.1 Research Objectives

In the forthcoming user study, participants will simulate typical usage scenarios with smartphones and tablets, enabling an evaluation of our method in an unconstrained environment. This evaluation will involve both quantitative and subjective comparisons. Our objective is to address the following research questions:

RQ1: How does the performance of mobile gaze estimation using front-facing cameras fare in interaction? In the context of using built-in front-facing RGB cameras on handheld smartphones and tablets, we need to investigate the effectiveness of gaze estimation in our interaction method, which will be explored in Study 1.

RQ2: How do different calibration techniques affect the interaction experience? Leveraging the insight that the interaction process
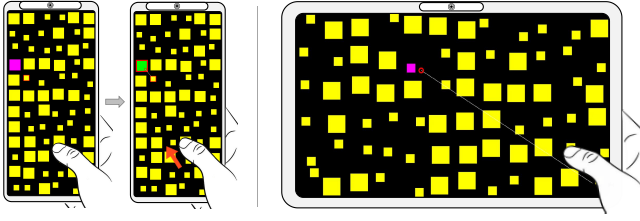
Figure 3: User interface during the studies. A 12×6 grid of square targets is randomly generated, comprising of two different sizes. Left: Smartphone interface (shown with GazeSwipe). Participants fixate their gaze on a target, causing the cursor to appear near it. They then swipe the cursor to the target, which turns green once reached. Right: Tablet interface (shown with Pure Cursor).

itself serves as the calibration, we propose a user-unaware auto-calibration method. The effectiveness of this method compared to explicit calibration will be investigated in Study 1.

RQ3: How does the performance of GazeSwipe compare to other reachability solutions? Compared to Direct Touch, One-Handed Mode, and Cursor-based Techniques without gaze input, can GazeSwipe provide a more seamless and accurate interaction experience in addressing reachability issues? Moreover, how does its performance differ between smartphones and tablets? These questions will be investigated in Study 2.

5.2 Participants

We recruited 15 participants (2 female, 13 male) aged between 21 and 29 years ($M = 24.33, SD = 2.26$). All of them was right-handed, and 12 of them wore glasses. Participants rated their experience with smartphone usage ($M = 4.87, SD = 0.35$), tablet usage ($M = 4.0, SD = 1.0$), and gaze interaction ($M = 3.53, SD = 0.83$) on a scale from 1 (no experience) to 5 (expert).

5.3 Apparatus

The user studies were conducted on two devices. For the smartphone, we used the Redmi K60, an Android device with a 6.67" screen, 1440×3200 px resolution, and a Qualcomm Snapdragon 8+ Gen1 SoC. For the tablet, we used the Xiaomi Pad 6, featuring an 11" screen, 2880×1800 px resolution, and a Snapdragon 870 SoC. The front cameras on both devices are positioned at the top center of the screen. Additionally, we tested the effectiveness of GazeSwipe on other Android devices with varying screen sizes, camera positions, and computational capabilities. Our method performed smoothly across all of these devices.

5.4 User Interface

The user interface is illustrated in Fig. 3. In the user study, the smartphone screen, with a resolution of 2268×1080 pt, is divided into a 12×6 grid, where each cell measures 189×180 pt. Each grid cell contains a randomly positioned yellow square element, either 50×50 pt or 100×100 pt in size. During the experiment, one randomly selected element is highlighted in magenta, and participants are

instructed to perform a short tap interaction on this target element using the specified interaction method.

For the Pure Cursor interaction method, a red circle represents the cursor. When the user's thumb touches the screen, the cursor initially appears at the touch point. As the thumb is swiped, both the starting cursor and the current cursor are displayed, with a directional line connecting them. Releasing the thumb triggers the interaction at the position of the current cursor. For GazeSwipe, the starting cursor automatically snaps to the nearest element, which is highlighted with a red outline around its edges. If the current cursor is positioned over the magenta target element, the element turns green.

5.5 Evaluation Metrics

5.5.1 Objective Metrics. In Study 1, we evaluated gaze estimation error across different calibration strategies, whereas in Study 2, we examined the latter three objective metrics across the four reachability techniques:

Gaze Estimation Error. This metric is determined by calculating the difference between the cursor coordinate when the user releases their thumb and the estimated gaze point. Gaze estimation error directly reflects the accuracy of gaze guidance, which in turn affects the drag distance during user interaction.

Thumb Movement Distance. This metric quantifies the distance traveled by the user's thumb from initial screen touch to release during each interaction.

Completion Time. This metric measures the duration from the user's initial thumb touch to its release during each interaction.

Success Rate. This metric quantifies the proportion of interactions where the user accurately taps within the designated target area, relative to the total number of interactions conducted.

5.5.2 Subjective Metrics. In Study 2, we also collected participant feedback through a questionnaire regarding their experiences with the four reachability techniques. The questionnaire assessed aspects such as ease of use, unreachability, grip stability, and fatigue. Participants also ranked their preferences for the four techniques.

## 6 Study 1: Calibration Strategies

In this study, our objective was to assess the efficacy of using the front-facing camera for gaze estimation in reachability interactions on smartphones and tablets, as well as to evaluate the accuracy of various gaze calibration strategies. To achieve this, participants were instructed to hold the mobile devices with one hand and perform experiments on the specified user interface, testing four calibration techniques on both the smartphone and tablet.

6.1 Comparative Setups

Participants interacted with four different calibration techniques for gaze-and-swipe interaction:

No calibration (NC). This configuration uses the output of the gaze estimation method without calibration, serving as the baseline to evaluate the fundamental capability of the neural network in targeting elements on smartphones and tablets.

Explicit calibration (EC). This involves an explicit 9-point calibration process before usage to improve gaze estimation accuracy.
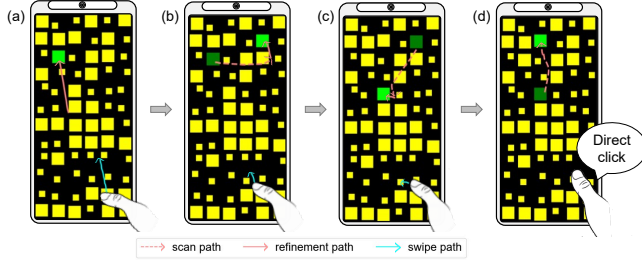
Figure 4: Example of gaze scanpaths collected during interactions. (a) At system startup, gaze errors are relatively large, requiring longer finger swipes. (b-c) After a few interactions, the length of finger swipes progressively decreases. (d) With prolonged use, the gaze can directly snap to the target, and a simple tap is sufficient to confirm the interaction.

Auto-calibration Strategy 1 (AC1). This implements the user-unaware auto-calibration method proposed in Section 4.2, collecting samples during interactions to automatically refine gaze estimation.

Auto-calibration Strategy 2 (AC2). This strategy integrates head pose factors into the auto-calibration process, providing more robust and accurate gaze calibration.

6.2 Procedure

The 15 participants followed this procedure for the study:

Preparation Phase: Participants first completed a personal information questionnaire, which collected details such as age, gender, handedness, use of glasses, and prior experience with smartphone operation, tablet operation, and eye-tracking interaction. Afterward, they were given a 2-minute period to freely familiarize themselves with the gaze-and-swipe interaction method.

Experimental Phase: Each participant was assigned to complete 4 calibration setups × 64 targets × 2 device types = 512 interaction trials. At the beginning of each experiment setup, the program cleared previous calibration data and randomly generated target layouts. Target selection during the experiment was randomized by the program. Participants were instructed to interact naturally and relaxedly with the targets according to the provided instructions. The duration of each participant's study session lasted approximately 30 minutes. Additionally, one participant was selected to conduct an extended experiment with 128 target selections on the smartphone, which was used to observe how gaze estimation error changes across more interactions for the four calibration strategies.

6.3 Results

The gaze estimation error for the four calibration strategies is illustrated in Fig. 5, with smartphone results on the left and tablet results on the right. For smartphones, a repeated-measures ANOVA revealed a significant effect of calibration strategy on gaze estimation error ($F_{3,56} = 5.26, p < 0.005, \eta_p^2 = 0.22$). Post hoc pairwise comparisons using the Tukey HSD method indicated that the mean gaze error followed the order NC>EC>AC1>AC2, with significant differences between NC-AC1 and NC-AC2 ($p < 0.01$).

For tablets, a repeated-measures ANOVA test also showed a significant effect of calibration strategy on gaze estimation error
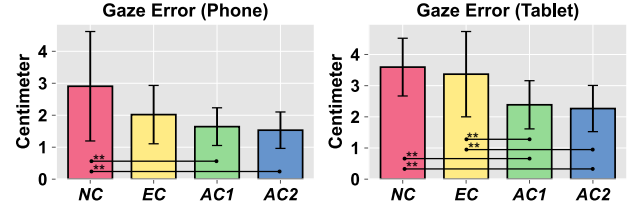
Figure 5: Gaze error for the four calibration strategies. Results on the smartphone (left) and on the tablet (right). The four strategies include: No Calibration (NC), Explicit Calibration (EC) using a 9-point calibration process, Auto-calibration Strategy 1 (AC1) without head pose, and Auto-calibration Strategy 2 (AC2) incorporating head pose. Error bars represent standard deviations, with significant differences marked by ** ($p < 0.05$).
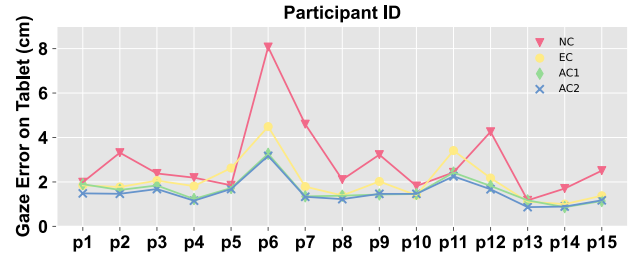
Figure 6: Gaze error for each participant. The error varies across participants before calibration but stabilizes after calibration, particularly with the two auto-calibration strategies.

($F_{3,56} = 7.06, p < 0.001, \eta_p^2 = 0.274$). Post hoc pairwise comparisons using the Tukey HSD method revealed that the mean gaze error followed the same order (NC>EC>AC1>AC2), with significant differences observed between NC-AC1, NC-AC2, EC-AC1, and EC-AC2 ($p < 0.05$).

When comparing smartphones and tablets, the gaze error was consistently about 0.5 cm higher on tablets, both before and after calibration. This discrepancy may be attributed to the fact that tablet screens are typically viewed from a greater distance, resulting in lower resolution facial images for gaze estimation. Additionally, at the same angular error, a greater viewing distance on tablets translates into a larger projected error on the screen. Nevertheless, due to the larger screen and UI elements on tablets, several participants reported in the subsequent study that gaze pointing felt more accurate on tablets than on smartphones.

We also observed differences in gaze error among individuals under the four different calibration strategies. As shown in Fig. 6, Participant P6 exhibited significant gaze error under No Calibration, but after applying the auto-calibration strategies, the gaze error reduced to less than half of the original value. This demonstrates the effectiveness and robustness of our calibration strategy.

As illustrated in Fig. 7, we examined the variation of gaze error over the number of interactions for the four calibration strategies. NC consistently maintained high gaze error (around 4 cm, not
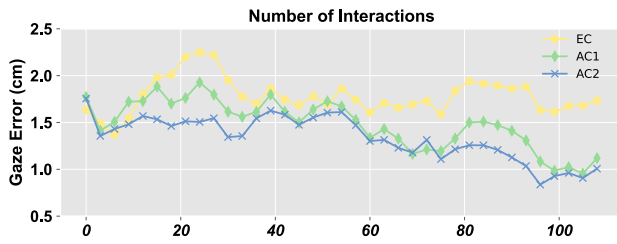
Figure 7: Gaze error of one participant over the number of interactions, with a sliding window of 16 and a step size of 4. As the number of interactions increases, AC1 and AC2 consistently collect the latest calibration samples, resulting in lower gaze errors compared to EC. Additionally, AC2, which accounts for head pose, demonstrates even lower gaze errors.

plotted in the figure). Although EC initially had lower gaze error compared to the auto-calibration methods, both AC1 and AC2 eventually outperformed EC as the number of interactions increased. The gaze error of EC fluctuated within a certain range, which we attribute to changes in user posture over time. These posture changes could lead to deviations from the initial calibration conditions, affecting the accuracy of gaze estimation. In contrast, AC1 and AC2 automatically recalibrated using the latest interaction samples, stabilizing gaze error at relatively low levels after a certain number of interactions. Since AC2 benefits from posture information in historical calibration samples, it was able to interpolate calibration offsets based on similar postures in the past. This enabled AC2 to generally achieve lower gaze error than AC1.

6.4 Discussion

In Study 1, our results addressed the first two research questions. Regarding RQ1, we observed substantial gaze errors in the uncalibrated output of the gaze estimation method for certain users, with some individual error distances approaching the width of the screen, indicating that the uncalibrated method could not consistently provide reliable gaze pointing. For RQ2, we found that applying calibration strategies significantly improved the quality of gaze estimation. Moreover, the auto-calibration strategies outperformed the explicit 9-point calibration method, as they continuously accumulated new samples during interaction to enhance the most recent results. Additionally, between the two auto-calibration strategies, the results indicated an advantage of AC2 over AC1. Therefore, we selected AC2 as the calibration strategy for GazeSwipe.

7 Study 2: Reachability Techniques

In this study, we quantitatively evaluated the effectiveness of GazeSwipe in addressing reachability issues on mobile touchscreens compared to other techniques. We selected Strategy 2 as the auto-calibration method for GazeSwipe and compared it with three other techniques: Direct Touch, One-handed Mode, and Pure Cursor.

7.1 Comparative Setups

We compared the following four techniques on the smartphone, and since the thumb cannot directly touch certain areas when holding the tablet with one hand, we compared only the latter three techniques on the tablet:

Direct Touch (DT). While holding the phone with one hand, users' thumbs typically reach only a portion of the screen, but they can adjust their grip to touch otherwise unreachable areas. We include this Direct Touch method as a baseline technique.

One-handed Mode (OM). This mode shrinks the user interface to a size fully reachable by the thumb and displays it in the bottom right corner of the screen (or bottom left for left-handed users). This feature allows users to directly touch areas that would normally be outside their thumb's reach and is commonly available on many Android smartphones.

Pure Cursor (PC). A cursor-based technique without gaze guidance, similar to BezelCursor [29], is employed. The main difference between this method and GazeSwipe is the initial position of the cursor. Here, the cursor starts at the user's touch position and extends linearly along the direction of the swipe by a certain factor, triggering interaction upon releasing the thumb.

GazeSwipe (GS). Our proposed technique utilizes user-unaware auto-calibration Strategy 2 to deliver robust gaze estimation results.

In the study, we randomized the order in which participants experienced each technique to ensure fairness in the comparison.

7.2 Procedure

Each participant was required to go through the following process:

Preparation Phase: After completing the personal information collection, participants were given 2 minutes to freely practice the four interaction techniques.

Experimental Phase: Each participant completed 4 techniques × 32 targets × 2 device types = 256 trials. Similar to the experimental phase in Study 1, the program randomly generated target layouts and selected targets for each trial. Participants were instructed to interact naturally with the targets using the specified techniques, simulating their typical smartphone usage. The calibration data from the preparation phase was cleared before the start of this phase, ensuring that no pre-calibration data was used in GazeSwipe. Participants were allowed to take short breaks after completing a set of trials if desired, and the total experiment duration for each participant was within 30 minutes.

Subjective Evaluation: After completing a set of trials for each technique, participants were asked to rate their level of agreement on a Likert scale (1: totally disagree; 5: totally agree) with statements including "The technique was easy to apply", "I had difficulty tapping on certain areas of the screen", "I maintained a stable grip during the interaction", and "I felt fatigued after using this technique for a while". Upon completing all experiments, they were asked to rank their preferences for the four techniques. Since our study specifically focuses on mobile touchscreen reachability, we prioritize metrics that reflect system usability and user workload based on the task context. Similar subjective questionnaires and metrics have been employed in several prior studies [5, 52].

7.3 Results on the Smartphone

Our interest is to understand the impact of various reachability solution techniques on user performance. We first analyze user performance on the smartphone. We conducted repeated-measures
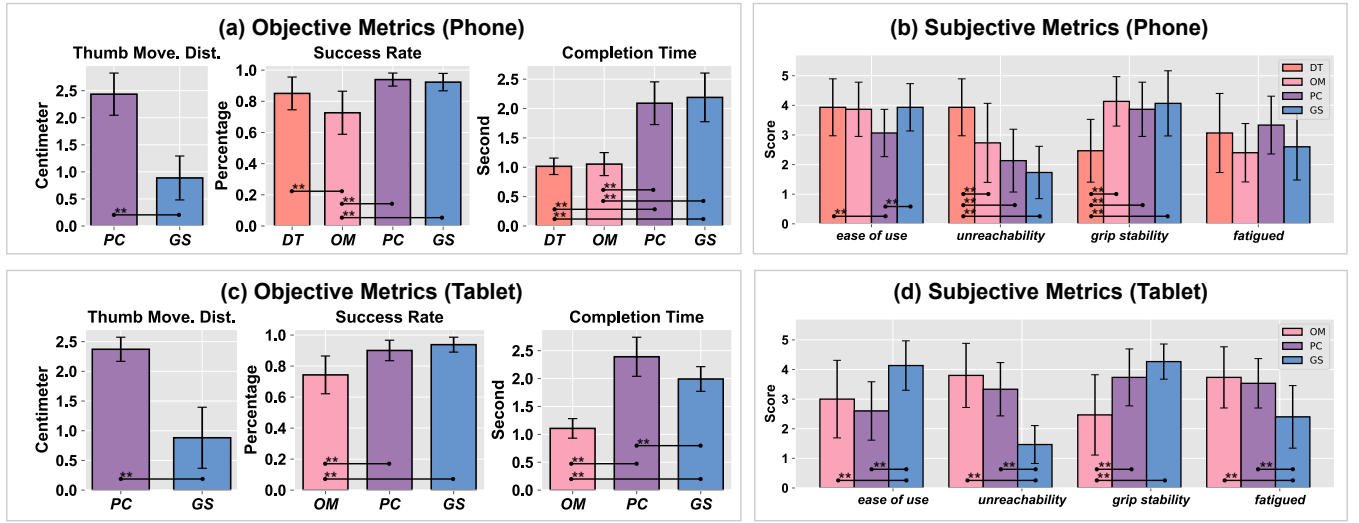
Figure 8: Objective and subjective results of Study 2. Panels (a-b) depict the results on the smartphone, while panels (c-d) illustrate the results on the tablet. The four techniques include: Direct Touch (DT), One-handed Mode (OM), Pure Cursor (PC), and GazeSwipe (GS). Error bars denote the standard deviation, with significant differences indicated by ** ($p < 0.05$).
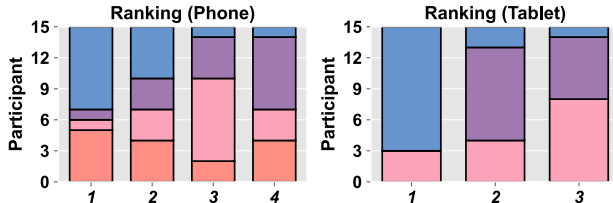

Figure 9: Preference ranking of the four reachability techniques on both smartphone and tablet (tablet excludes DT). A rank of 1 indicates the most preferred method. It can be seen that GazeSwipe (GS) received the highest preference on both devices.
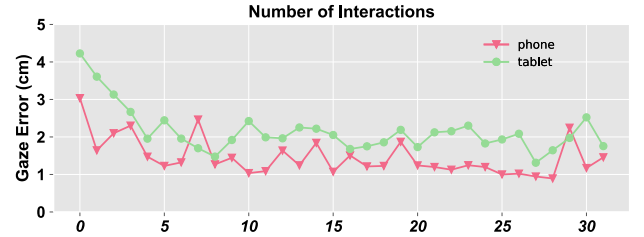

Figure 10: Mean gaze error of all participants over the number of interactions (no sliding window) in Study 2 for both phone and tablet. The gaze error is relatively high at the beginning, but decreases to a lower level as the number of interactions increases on both devices.

ANOVA and post hoc pairwise comparisons to evaluate and analyze the objective and subjective metrics recorded in Study 2, including thumb movement distance, success rate, completion time, ease of use, unreachability, grip stability, fatigue, and preference ranking.

7.3.1 Objective Measures. The thumb movement distance is shown in the left column of Fig. 8 (a). Since DT and OM do not involve swiping, this metric is not applicable to them. Thus, we only compare PC and GS. Repeated-measures ANOVA revealed a significant effect of technique on **thumb movement distance** ($F_{1,28} = 113.567, p < 0.001, \eta_p^2 = 0.802$). Tukey HSD post hoc pairwise comparisons indicated significant differences between the two techniques ($p < 0.001$). We found that PC requires a significantly longer thumb movement distance (2.436 cm) compared to GS (0.887 cm). This indicates that gaze pointing greatly reduces the swiping distance required for users, alleviating user fatigue and enabling more precise target selection.

The success rate is shown in the center column of Fig. 8 (a). Technique has a significant impact on **success rate** ($F_{3,56} = 16.094, p < 0.001, \eta_p^2 = 0.463$). Tukey HSD post hoc pairwise comparisons indicated significant differences between DT-OM, OM-PC, and OM-GS ($p < 0.001$). The results indicate that, compared to DT (85.1%) and OM (72.6%), the success rates of PC (93.9%) and GS (92.3%) are significantly higher.

The completion time is shown in the right column of Fig. 8 (a). Technique has a significant impact on **completion time** ($F_{3,56} = 67.832, p < 0.001, \eta_p^2 = 0.784$). Tukey HSD post hoc pairwise comparisons are all significant ($p < 0.001$) except DT-OM and PC-GS. As expected, DT (1.016 s) and OM (1.054 s) are the fastest due to their lack of swiping process, and participants' familiarity with the direct tapping interaction. However, as mentioned earlier, their success rate is significantly lower. PC (2.091 s) ranks third in terms of completion time, while GS (2.190 s) takes the longest, but the difference in completion time between GS and PC is insignificant.

7.3.2 Subjective Measures. Fig. 8 (b) illustrates the results of the user survey. The four indicators correspond to the four survey questions presented in Section 7.2. Repeated-measures ANOVA shows that technique has a significant effect on **ease of use** ($F_{3,56} = 3.54, p < 0.05, \eta_p^2 = 0.159$), and post hoc pairwise comparisons reveal significant differences between DT-PC and PC-GS ($p < 0.05$), with the sequence DT>GS>OM>PC. Technique also has a significant effect on **unreachability** ($F_{3,56} = 11.975, p < 0.001, \eta_p^2 = 0.391$). Significant differences were found between DT-OM, DT-PC, and PC-GS, with the order GS<PC<OM<DT, suggesting that DT suffers from severe reachability issues. In addition, technique has a significant effect on **grip stability** ($F_{3,56} = 9.586, p < 0.001, \eta_p^2 = 0.339$). Significant differences were found between DT-OM, DT-PC, and DT-GS, indicating that users felt significantly less stable when using DT, while the other three methods provided better grip stability. However, technique did not have a significant effect on **fatigue** ($F_{3,56} = 2.199, p > 0.05, \eta_p^2 = 0.105$). For the **preference ranking**, users' average preferences are ranked as GS>DT>OM>PC, as shown in Fig. 9.

7.4 Results on the Tablet

We then evaluated the metrics on the tablet. Due to the significant difference in screen size between the tablet and the smartphone, the metrics exhibited distinct characteristics.

7.4.1 Objective Measures. Fig. 8 (c) illustrates the results of objective measures on the tablet. Repeated-measures ANOVA revealed a significant effect of technique on **thumb movement distance** ($F_{1,28} = 109.211, p < 0.001, \eta_p^2 = 0.796$). Tukey HSD post hoc pairwise comparisons indicated significant differences between the two techniques ($p < 0.001$). Similar to the smartphone results, PC requires a significantly longer thumb movement distance (2.372 cm) compared to GS (0.882 cm). Technique also has a significant impact on **success rate** ($F_{2,42} = 22.359, p < 0.001, \eta_p^2 = 0.516$). Tukey HSD post hoc pairwise comparisons revealed significant differences between OM-PC and OM-GS ($p < 0.001$). The results indicate that, compared to OM (74.3%), the success rates of PC (90.1%) and GS (93.8%) are significantly higher. Due to the larger screen size of the tablet, PC requires a longer cursor displacement with the same thumb movement distance, which increases the likelihood of errors. Technique has a significant impact on **completion time** ($F_{2,42} = 96.123, p < 0.001, \eta_p^2 = 0.821$). Tukey HSD post hoc pairwise comparisons are all significant ($p < 0.001$). OM (1.107 s) remains the fastest, while GS (1.993 s) and PC (2.391 s) are slower.

7.4.2 Subjective Measures. Fig. 8 (d) illustrates the results of subjective measures on the tablet. Repeated-measures ANOVA shows that technique has a significant effect on **ease of use** ($F_{2,42} = 8.42, p < 0.001, \eta_p^2 = 0.286$). Post hoc pairwise comparisons reveal significant differences between OM-GS and PC-GS ($p < 0.05$), with the ranking of GS>OM>PC. Technique also significantly impacts **unreachability** ($F_{2,42} = 28.697, p < 0.001, \eta_p^2 = 0.577$). Significant differences were found between OM-GS and PC-GS, with the order GS<PC<OM, indicating that GS has better reachability compared to the other two methods. Additionally, technique significantly affects **grip stability** ($F_{2,42} = 12.352, p < 0.001, \eta_p^2 = 0.37$). Significant differences were found between OM-PC and OM-GS, with the

ranking GS>PC>OM, suggesting that PC and GS offer better grip stability than OM. Technique also significantly influences **fatigue** ($F_{2,42} = 8.089, p < 0.001, \eta_p^2 = 0.278$), with the order GS<PC<OM. Significant differences were observed between OM-GS and PC-GS, indicating that GS is notably less fatiguing on the tablet. For **preference ranking**, users' average preferences are ranked as GS>OM>PC, as shown in Fig. 9.

7.5 Discussion

Following the completion of Study 2, we discussed RQ3, comparing the performance differences between our proposed GazeSwipe technique (GS) and the other three techniques: DT, OM, and PC.

7.5.1 Direct Touch vs. GazeSwipe. Direct Touch (DT) serves as the baseline in this study, exhibiting significant reachability issues as evidenced by participants' subjective feedback, indicated by higher unreachability ratings. This necessitates frequent grip adjustments, leading to feelings of instability and fatigue, along with a lower success rate than cursor-based method. However, DT still ranked second in preference on smartphones, just after GazeSwipe. This may be attributed to its familiar operation and faster completion times.

7.5.2 Pure Cursor vs. GazeSwipe. Pure Cursor (PC) addresses DT's reachability issues using cursor manipulation, which proves to be an effective solution. Results show higher success rates and a perceived sense of greater reachability and grip stability compared to DT. However, PC's cursor extends from the thumb touch point, requiring longer thumb movement distances and resulting in higher fatigue compared to GazeSwipe. On tablets, controlling the cursor across the large screen with small thumb movements proves even more challenging. As a result, PC received the lowest preference ranking on both devices.

7.5.3 One-handed Mode vs. GazeSwipe. One-handed Mode (OM) tackles reachability issues by shrinking the screen, a feature employed in many Android smartphones. On smartphones, the results show that OM and GazeSwipe receive similar ratings in ease of use, grip stability, and fatigue. However, OM falls short of both GazeSwipe and even DT in terms of success rate, as the reduced element size makes thumb tapping more difficult. OM does have an advantage in completion time over GazeSwipe, as it, like DT, does not require swiping.

On tablets, GazeSwipe demonstrates a more pronounced advantage over OM, outperforming it in all metrics except for completion time. This is likely due to the large screen requiring a greater reduction in interface size for OM, making precise taps harder. On the other hand, enlarging the interface to improve accuracy leads to reduced reachability and grip stability, increasing fatigue.

Interestingly, when we asked the 15 participants whether they prioritized completion time or success rate during interactions, 11 preferred success rate, while 4 considered both equally important. This suggests that users may tolerate moderately longer completion times if the interaction method offers greater precision.

7.5.4 Smartphone vs. Tablet. When comparing the performance of GazeSwipe on smartphones and tablets, some participants reported

that the interaction felt more effective on tablets, with several attributing this to more accurate gaze pointing on the larger screen. Others noted that, in contrast, One-handed Mode and Pure Cursor were less ideal solutions on tablets.

From the objective metrics, although the gaze estimation error after calibration was slightly higher on tablets compared to smartphones, the larger UI elements on tablets allowed for more precise gaze-based target selection. Additionally, the success rate and completion time for GazeSwipe were similar on both devices, with a slight advantage observed on tablets.

Subjectively, GazeSwipe did not show significant differences compared to One-handed Mode across the four subjective metrics on smartphones. However, on tablets, GazeSwipe significantly outperformed One-handed Mode in all four metrics, indicating that it provides the best reachability experience on tablets.

In terms of preference ranking, GazeSwipe was the most favored solution on both smartphones and tablets, highlighting its intuitive and seamless interaction approach as a effective option across both device types.

7.5.5 Overall. In summary, GazeSwipe effectively addresses the reachability issues inherent in Direct Touch, significantly enhancing user experience compared to Pure Cursor. Furthermore, GazeSwipe outperforms or matches One-handed Mode in all aspects except for completion time, with this advantage being even more significant on tablets.

8 Limitations, Future Work and Conclusion

8.1 Limitations and Future Work

While our method demonstrates effectiveness, it still has some limitations. Firstly, since GazeSwipe does not rely on explicit calibration, the gaze point may be less accurate initially, although it improves in accuracy after several interactions. Secondly, due to the use of visual methods, our technique may not be suitable for dark environments. This could potentially be improved by implementing automatic screen brightness adjustment during interaction to provide adequate illumination. Thirdly, some participants observed that the starting position of the gaze-guided cursor could be located anywhere around the target, leading to varying drag directions with each interaction, which might increase their response time. This could possibly be addressed by introducing an offset to the initial position to maintain consistent drag directions. Furthermore, our method lacks user recognition and does not store historical calibration information, which could limit its usability in practical applications. Lastly, for more complex applications, such as typing, the interaction method remains too slow to be practical in real-world scenarios.

As the combination of gaze, finger swipe, and auto-calibration provides a robust solution for reachability issues on mobile devices, future work could focus on further improving the accuracy of gaze estimation or designing more efficient auto-calibration strategies. Potential directions for enhancing auto-calibration strategies include making better use of additional samples collected during a single interaction, seamlessly integrating the gaze estimation network with the calibrator, and considering additional factors such as time, lighting, and other environmental variables. Furthermore,

exploring more multimodal interaction methods for daily use on mobile devices presents a promising avenue for future research.

8.2 Conclusion

In this paper, we propose GazeSwipe, a novel gaze-and-swipe interaction technique in both smartphones and tablets aimed at addressing reachability issues encountered when using mobile touchscreens with one hand. Users simply need to naturally gaze at the desired target, while controlling the gaze cursor to reach the target through a finger-swipe operation at any position on the screen. To ensure robust gaze estimation results, especially when head movements occur frequently, we introduce a user-unaware auto-calibration strategy. Based on the insight that the interaction process can serve as the calibration process, this method eliminates the need for additional calibration procedures, enabling dynamic and implicit calibration. We also investigated the effectiveness of our auto-calibration strategy and interaction technique. The results indicate that our auto-calibration provides more accurate gaze pointing compared to no calibration and 9-point calibration and that our method achieves high success rates and top preference rankings against Direct Touch, One-handed Mode and Pure Cursor on both smartphones and tablets. Specifically on tablets, our method outperforms others across all subjective and objective metrics, with the exception of a moderately longer completion time.

Acknowledgments

References

[1] Yiwei Bao, Yihua Cheng, Yunfei Liu, and Feng Lu. 2021. Adaptive Feature Fusion Network for Gaze Tracking in Mobile Tablets. http://arxiv.org/abs/2103.11119 arXiv:2103.11119 [cs].

[2] Will Butler. 2024. Apple announces new accessibility features, including Eye Tracking. https://www.apple.com/newsroom/2024/05/apple-announces-new-accessibility-features-including-eye-tracking/

[3] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, Seoul Republic of Korea, 1499–1508. https://doi.org/10.1145/2702123.2702442

[4] Yihua Cheng and Feng Lu. 2022. Gaze Estimation using Transformer. In *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, Montreal, QC, Canada, 3341–3347. https://doi.org/10.1109/ICPR56361.2022.9956687

[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*. ACM, Glasgow Scotland Uk, 1–12. https://doi.org/10.1145/3290605.3300442

[6] Heiko Drewes, Alexander De Luca, and Albrecht Schmidt. 2007. Eye-gaze interaction for mobile phones. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology (Mobility '07)*. Association for Computing Machinery, New York, NY, USA, 364–371. https://doi.org/10.1145/1378063.1378122

[7] Morten Lund Dybdal, Javier San Agustin, and John Paulin Hansen. 2012. Gaze input for mobile devices by dwell and gestures. In *Proceedings of the Symposium on Eye Tracking Research and Applications*. ACM, Santa Barbara California, 225–228. https://doi.org/10.1145/2168556.2168601

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

[9] Augusto Esteves, Elizabeth Bouquet, Ken Pfeuffer, and Florian Alt. 2022. One-handed Input for Mobile Devices via Motion Matching and Orbits Controls.

Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 6, 2 (July 2022), 1–24. https://doi.org/10.1145/3534624

[10] Tobias Fischer, Hyung Jin Chang, and Yiannis Demiris. 2018. RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments. In *Computer Vision – ECCV 2018*, Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Vol. 11214. Springer International Publishing, Cham, 339–357. https://doi.org/10.1007/978-3-030-01249-6_21 Series Title: Lecture Notes in Computer Science.

[11] Kenneth Alberto Funes Mora and Jean-Marc Odobez. 2014. Geometric Generative Gaze Estimation (G ^{3} E) for Remote RGB-D Cameras. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Columbus, OH, 1773–1780. https://doi.org/10.1109/CVPR.2014.229

[12] Huawei. 2019. One-Handed Mode | HUAWEI Support Global. https://consumer.huawei.com/en/support/content/en-us00754292/

[13] Sinh Huynh, Rajesh Krishna Balan, and JeongGil Ko. 2021. iMon: Appearance-based Gaze Tracking System on Mobile Devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 4 (Dec. 2021), 1–26. https://doi.org/10.1145/3494999

[14] Ryo Ikeda, Kyohei Hakka, and Buntarou Shizuki. 2021. Hover-Based Reachability Technique for Executing Single-Touch Gesture on Smartphone. In *Asian CHI Symposium 2021*. ACM, Yokohama Japan, 9–15. https://doi.org/10.1145/3429360.3468171

[15] Kaori Ikematsu, Haruna Oshima, Rachel Eardley, and Itiro Siio. 2020. Investigating How Smartphone Movement is Affected by Lying Down Body Posture. *Proceedings of the ACM on Human-Computer Interaction* 4, ISS (Nov. 2020), 1–17. https://doi.org/10.1145/3427320

[16] Amy K. Karlson and Benjamin B. Bederson. 2007. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. In *Human-Computer Interaction – INTERACT 2007*, Cécilia Baranauskas, Philippe Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa (Eds.). Vol. 4662. Springer Berlin Heidelberg, Berlin, Heidelberg, 324–338. https://doi.org/10.1007/978-3-540-74796-3_30 Series Title: Lecture Notes in Computer Science.

[17] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, and Antonio Torralba. 2019. Gaze360: Physically Unconstrained Gaze Estimation in the Wild. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Seoul, Korea (South), 6911–6920. https://doi.org/10.1109/ICCV.2019.00701

[18] 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*. ACM, Barcelona Spain, 1–17. https://doi.org/10.1145/3229434.3229452

[19] Mohamed Khamis, Regina Hasholzner, Andreas Bulling, and Florian Alt. 2017. GTmoPass: two-factor authentication on public displays using gaze-touch passwords and personal mobile devices. In *Proceedings of the 6th ACM International Symposium on Pervasive Displays*. ACM, Lugano Switzerland, 1–9. https://doi.org/10.1145/3078810.3078815

[20] Mohamed Khamis, Karola Marky, Andreas Bulling, and Florian Alt. 2022. User-centred multimodal authentication: securing handheld mobile devices using gaze and touch input. *Behaviour & Information Technology* 41, 10 (July 2022), 2061–2083. https://doi.org/10.1080/0144929X.2022.2069597

[21] Sunjun Kim, Jihyun Yu, and Geehyuk Lee. 2012. Interaction techniques for unreachable objects on the touchscreen. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*. ACM, Melbourne Australia, 295–298. https://doi.org/10.1145/2414536.2414585

[22] Andy Kong, Karan Ahuja, Mayank Goel, and Chris Harrison. 2021. EyeMU Interactions: Gaze + IMU Gestures on Mobile Devices. In *Proceedings of the 2021 International Conference on Multimodal Interaction*. ACM, Montréal QC Canada, 577–585. https://doi.org/10.1145/3462244.3479938

[23] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Las Vegas, NV, USA, 2176–2184. https://doi.org/10.1109/CVPR.2016.239

[24] 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* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173655

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

[26] Huy Viet Le, Patrick Bader, Thomas Kosch, and Niels Henze. 2016. Investigating Screen Shifting Techniques to Improve One-Handed Smartphone Usage. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*. ACM, Gothenburg Sweden, 1–10. https://doi.org/10.1145/2971485.2971562

[27] Yaxiong Lei, Shijing He, Mohamed Khamis, and Juan Ye. 2024. An End-to-End Review of Gaze Estimation and its Interactive Applications on Handheld Mobile Devices. *Comput. Surveys* 56, 2 (Feb. 2024), 1–38. https://doi.org/10.1145/3606947

[28] Yaxiong Lei, Yuheng Wang, Tyler Caslin, Alexander Wisowaty, Xu Zhu, Mohamed Khamis, and Juan Ye. 2023. DynamicRead: Exploring Robust Gaze Interaction Methods for Reading on Handheld Mobile Devices under Dynamic Conditions. *Proceedings of the ACM on Human-Computer Interaction* 7, ETRA (May 2023), 1–17. https://doi.org/10.1145/3591127

[29] Wing Ho Andy Li and Hongbo Fu. 2013. BezelCursor: bezel-initiated cursor for one-handed target acquisition on mobile touch screens. In *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications (SA '13)*. Association for Computing Machinery, New York, NY, USA, 1. https://doi.org/10.1145/2543651.2543680

[30] Feng Lu, Yue Gao, and Xiaowu Chen. 2016. Estimating 3D Gaze Directions Using Unlabeled Eye Images via Synthetic Iris Appearance Fitting. *IEEE Transactions on Multimedia* 18, 9 (Sept. 2016), 1772–1782. https://doi.org/10.1109/TMM.2016.2576284

[31] Sven Mayer, Gierad Laput, and Chris Harrison. 2020. Enhancing Mobile Voice Assistants with WorldGaze. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–10. https://doi.org/10.1145/3313831.3376479

[32] Takashi Nagamatsu, Michiya Yamamoto, and Hiroshi Sato. 2010. MobiGaze: development of a gaze interface for handheld mobile devices. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*. ACM, Atlanta Georgia USA, 3349–3354. https://doi.org/10.1145/1753846.1753983

[33] Vikrant Nagpure and Kenji Okuma. 2023. Searching Efficient Neural Architecture with Multi-resolution Fusion Transformer for Appearance-based Gaze Estimation. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa, HI, USA, 890–899. https://doi.org/10.1109/WACV56688.2023.00095

[34] Atsushi Nakazawa and Christian Nitschke. 2012. Point of Gaze Estimation through Corneal Surface Reflection in an Active Illumination Environment. In *Computer Vision – ECCV 2012*, Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid (Eds.). Vol. 7573. Springer Berlin Heidelberg, Berlin, Heidelberg, 159–172. https://doi.org/10.1007/978-3-642-33709-3_12 Series Title: Lecture Notes in Computer Science.

[35] Omar Namnakani, Yasmeen Abdrabou, Jonathan Grizou, Augusto Esteves, and Mohamed Khamis. 2023. Comparing Dwell time, Pursuits and Gaze Gestures for Gaze Interaction on Handheld Mobile Devices. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–17. https://doi.org/10.1145/3544548.3580871

[36] Seonwook Park, Emre Aksan, Xucong Zhang, and Otmar Hilliges. 2020. Towards End-to-end Video-based Eye-Tracking. http://arxiv.org/abs/2007.13120 arXiv:2007.13120 [cs].

[37] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. 2015. Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, Charlotte NC USA, 373–383. https://doi.org/10.1145/2807442.2807460

[38] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2021. Multi-user Gaze-based Interaction Techniques on Collaborative Touchscreens. In *ACM Symposium on Eye Tracking Research and Applications*. ACM, Virtual Event Germany, 1–7. https://doi.org/10.1145/3448018.3458016

[39] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, Tokyo Japan, 301–311. https://doi.org/10.1145/2984511.2984514

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

[41] Sheikh Rivu, Yasmeen Abdrabou, Thomas Mayer, Ken Pfeuffer, and Florian Alt. 2019. GazeButton: enhancing buttons with eye gaze interactions. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications - ETRA '19*. ACM Press, Denver, Colorado, 1–7. https://doi.org/10.1145/3314111.3318154

[42] Anne Roudaut, Stéphane Huot, and Eric Lecolinet. 2008. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces*. ACM, Napoli Italy, 146–153. https://doi.org/10.1145/1385569.1385594

[43] Ludwig Sidenmark and Anders Lundström. 2019. Gaze behaviour on interacted objects during hand interaction in virtual reality for eye tracking calibration. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. ACM, Denver Colorado, 1–9. https://doi.org/10.1145/3314111.3319815

[44] Shyamli Sindhwani, Christof Lutteroth, and Gerald Weber. 2019. ReType: Quick Text Editing with Keyboard and Gaze. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–13. https://doi.org/10.1145/3290605.3300433

[45] Sophie Stellmach and Raimund Dachselt. 2012. Look & touch: gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin Texas USA, 2981–2990. https://doi.org/10.1145/2207676.2208709

[46] Yusuke Sugano, Yasuyuki Matsushita, Yoichi Sato, and Hideki Koike. 2008. An Incremental Learning Method for Unconstrained Gaze Estimation. In *Computer Vision – ECCV 2008*, David Forsyth, Philip Torr, and Andrew Zisserman (Eds.). Vol. 5304. Springer Berlin Heidelberg, Berlin, Heidelberg, 656–667. https://doi.org/10.1007/978-3-540-88690-7_49 Series Title: Lecture Notes in Computer Science.

[47] Li Sun, Zicheng Liu, and Ming-Ting Sun. 2015. Real time gaze estimation with a consumer depth camera. *Information Sciences* 320 (Nov. 2015), 346–360. https://doi.org/10.1016/j.ins.2015.02.004

[48] Hsin-Ruey Tsai, Da-Yuan Huang, Chen-Hsin Hsieh, Lee-Ting Huang, and Yi-Ping Hung. 2016. MovingScreen: selecting hard-to-reach targets with automatic comfort zone calibration on mobile devices. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. ACM, Florence Italy, 651–658. https://doi.org/10.1145/2957265.2961835

[49] R. Valenti, N. Sebe, and T. Gevers. 2012. Combining Head Pose and Eye Location Information for Gaze Estimation. *IEEE Transactions on Image Processing* 21, 2 (Feb. 2012), 802–815. https://doi.org/10.1109/TIP.2011.2162740

[50] Evangelos Ververas, Polydefkis Gkagkos, Jiankang Deng, Michail Christos Doukas, Jia Guo, and Stefanos Zafeiriou. 2023. 3DGazeNet: Generalizing Gaze Estimation with Weak-Supervision from Synthetic Views. http://arxiv.org/abs/2212.02997 arXiv:2212.02997 [cs].

[51] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. ACM, Zurich Switzerland, 439–448. https://doi.org/10.1145/2493432.2493477

[52] Simon Voelker, Sebastian Hueber, Christian Corsten, and Christian Remy. 2020. HeadReach: Using Head Tracking to Increase Reachability on Mobile Touch Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–12. https://doi.org/10.1145/3313831.3376868

[53] Simon Voelker, Sebastian Hueber, Christian Holz, Christian Remy, and Nicolai Marquardt. 2020. GazeConduits: Calibration-Free Cross-Device Collaboration through Gaze and Touch. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–10. https://doi.org/10.1145/3313831.3376578

[54] Kang Wang and Qiang Ji. 2017. Real Time Eye Gaze Tracking with 3D Deformable Eye-Face Model. In *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Venice, 1003–1011. https://doi.org/10.1109/ICCV.2017.114

[55] Xuehan Xiong, Zicheng Liu, Qin Cai, and Zhengyou Zhang. 2014. Eye gaze tracking using an RGBD camera: a comparison with a RGB solution. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*. ACM, Seattle Washington, 1113–1121. https://doi.org/10.1145/2638728.2641694

[56] Xing-Dong Yang, Edward Mak, Pourang Irani, and Walter F. Bischof. 2009. Dual-Surface input: augmenting one-handed interaction with coordinated front and behind-the-screen input. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, Bonn Germany, 1–10. https://doi.org/10.1145/1613858.1613865

[57] Neng-Hao Yu, Da-Yuan Huang, Jia-Jyun Hsu, and Yi-Ping Hung. 2013. Rapid selection of hard-to-access targets by thumb on mobile touch-screens. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. ACM, Munich Germany, 400–403. https://doi.org/10.1145/2493190.2493202

[58] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. 2020. ETH-XGaze: A Large Scale Dataset for Gaze Estimation under Extreme Head Pose and Gaze Variation. http://arxiv.org/abs/2007.15837 arXiv:2007.15837 [cs].

[59] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2017. MPIIGaze: Real-World Dataset and Deep Appearance-Based Gaze Estimation. http://arxiv.org/abs/1711.09017 arXiv:1711.09017 [cs].

[60] Maozheng Zhao, Henry Huang, Zhi Li, Rui Liu, Wenzhe Cui, Kajal Toshniwal, Ananya Goel, Andrew Wang, Xia Zhao, Sina Rashidian, Furqan Baig, Khiem Phi, Shumin Zhai, Iv Ramakrishnan, Fusheng Wang, and Xiaojun Bi. 2022. EyeSayCorrect: Eye Gaze and Voice Based Hands-free Text Correction for Mobile Devices. In *27th International Conference on Intelligent User Interfaces*. ACM, Helsinki Finland, 470–482. https://doi.org/10.1145/3490099.3511103