PalmGazer: Unimanual Eye-hand Menus in Augmented Reality

Ken Pfeuffer
Aarhus University
Aarhus, Denmark
ken@cs.au.dk

Jan Obernolte
LMU
Munich, Germany
jan.obernolte@campus.lmu.de

Felix Dietz
Bundeswehr University Munich
Munich, Germany
felix.dietz@unibw.de

Ville Mäkelä
University of Waterloo
Waterloo, Canada
ville.makela@uwaterloo.ca

Ludwig Sidenmark
University of Toronto
Toronto, Canada
lsidenmark@dgp.toronto.edu

Pavel Manakhov
Aarhus University
Aarhus, Denmark
pmanakhov@cs.au.dk

Minna Pakanen
Aarhus University
Aarhus, Denmark
mpakanen@cc.au.dk

Florian Alt
Bundeswehr University Munich
Munich, Germany
florian.alt@unibw.de

ABSTRACT

How can we design the user interfaces for augmented reality (AR) so that we can interact as simple, flexible and expressive as we can with smartphones in one hand? To explore this question, we propose PalmGazer as an interaction concept integrating eye-hand interaction to establish a singlehandedly operable menu system. In particular, PalmGazer is designed to support quick and spontaneous digital commands– such as to play a music track, check notifications or browse visual media – through our devised three-way interaction model: hand opening to summon the menu UI, eye-hand input for selection of items, and dragging gesture for navigation. A key aspect is that it remains always-accessible and movable to the user, as the menu supports meaningful hand and head based reference frames. We demonstrate the concept in practice through a prototypical mobile UI with application probes, and describe technique designs specifically-tailored to the application UI. A qualitative evaluation highlights the system's interaction benefits and drawbacks, e.g., that common 2D scroll and selection tasks are simple to operate, but higher degrees of freedom may be reserved for two hands. Our work contributes interaction techniques and design insights to expand AR's uni-manual capabilities.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**; **Mixed / Augmented reality**.

KEYWORDS

augmented reality, menu, gaze, gestures, eye-hand interaction

SUI '23, October 13–15, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0281-5/23/10…$15.00
https://doi.org/10.1145/3607822.3614523

ACM Reference Format:
Ken Pfeuffer, Jan Obernolte, Felix Dietz, Ville Mäkelä, Ludwig Sidenmark, Pavel Manakhov, Minna Pakanen, and Florian Alt. 2023. PalmGazer: Unimanual Eye-hand Menus in Augmented Reality. In *The 2023 ACM Symposium on Spatial User Interaction (SUI '23), October 13–15, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3607822.3614523

1 INTRODUCTION

Whereas the user interface (UI) for augmented reality (AR) head-mounted devices (HMD) is maturing for stationary settings like at home or at work, less has been focused on AR UIs tailored for aiding people in their daily life like smartphones do presently. Everyday environments are highly dynamic and confront users with a variety of information which can lead to situations where users have limited physical and attentional resources available. One of the most common types of scenarios is when just one hand is available. Consider the case with smartphones. They allow to readily grip the device to bring it into view and support a variety of actions from spontaneously playing a song to extended sessions browsing photographs, reading messages, and navigating a map –all unimanually. A phenomenon that enables this is the duality of two aspects: i) the device is *handheld* which allows to bring the content into the ideal viewing place, and ii) the device is *interactive* through few, simple gestures that cover an expressive range of UI controls.

Such basic yet expressive one-handed interactions are desirable but only partially supported in gestural AR UIs (Figure 1). The landscape for one-hand interaction is two-fold. First, handheld UIs such as the Hand Menu [30] support one hand performing all UI activities. They also use the non-dominant hand as reference frame for a UI. This enables users to use their human sense of proprioception to stay aware of where the UI is in relation to the body [32]. Taken together, this would be typically classified as a case of asymmetric bi-manual interaction according to Giuard [12]. Second, users may easily interact uni-manually when the UI is detached from the user's hand and instead is for instance fixed to any location in the physical world. This ties the user to a stationary place, which is useful in situations where you remain for an extended period of time. However, if a user's position in

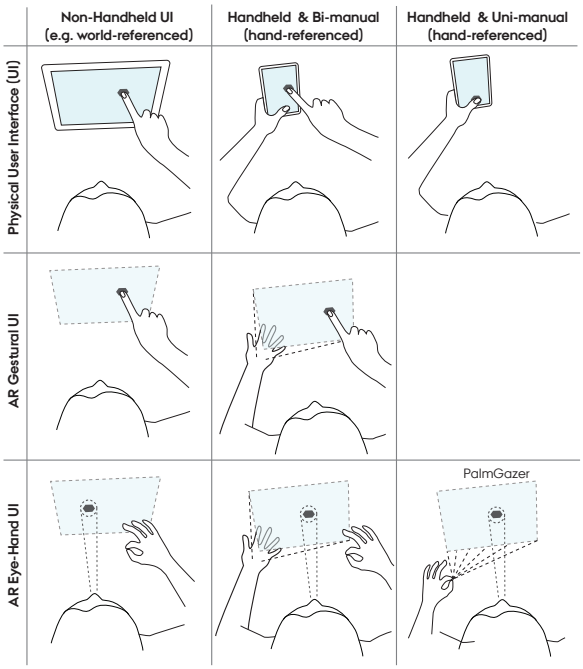

Figure 1: Physical UIs support the primary handheld and one-handed interaction classes (top row). Gestural and eye-hand UIs partially cover one-handed interaction. PalmGazer is an eye-hand UI for handheld, uni-manual interaction in AR.

the world space changes, he or she must manually relocate and re-instantiate the UI, which may be time-consuming [24].

A 3D interaction medium based on the eyes and hands in combination has lately received rising attention in the Human-Computer Interaction (HCI) literature [8, 26, 33, 51]. AR HMDs (e.g., Microsoft HoloLens, Meta Quest, Magic Leap, and Apple Vision Pro) integrate variations of this medium. Eye-hand interaction has so far been proposed for the same stationary cases as for gestural UIs: world-referenced UIs for one-handed tasks [21] and handheld UIs for two-handed tasks [35, 39]. In this work, we take a distinct direction by exploring how the two modalities can facilitate, and if yes to what extend, handheld unimanual interaction.

PalmGazer is a novel UI system for AR HMDs that support eye and hand tracking. PalmGazer combines eye and hand gestures to establish a simple but sufficiently-rich interaction model to afford expressive one-handed menu operations. At its core, it covers three fundamental interaction tasks through three mutually-compatible interaction techniques:

- **UI Activation (Figure 2-left)**: It starts with the user activating the UI by a palm-open gesture that summons a hand-attached home menu. Hereafter they are free to adjust the UI position by hand to place it mid-air as desired, as the UI remains active as long as the palm is opened. The user closes their hand to disengage the UI at any moment of time, facilitating the notion that it is readily available and easily dismissable at will.
- **Selection (Figure 2-center)**: Instead of then engaging a second hand, the interaction design is fully tailored to the same hand – via pinch gestures while the palm is opened. To select an object in the menu, eye-hand input is employed in form of gaze for target acquisition and a quick-release pinch gesture for confirmation. This allows for compound interactions where users can rapidly dovetail palm-open, look, and pinch for a rapid one-off action.
- **Navigation (Figure 2-right)**: Navigation commands, e.g., scrolling, extend the UI beyond a page. Here performed by a pinch dragging gesture in the respective direction, it involves a design conflict. Moving the hand for scrolling inevitably means the handheld UI is moved in its place. We address this by a peephole-inspired UI behaviour that retains spatial relationships of content in space.

All three techniques may be effortlessly integrated into a compound system operation. From an input-theoretic standpoint, a first key benefit is that the three techniques establish a coherent and simple input command language for one hand only– that lends itself to be as expressive as two-handed or non-hand-referenced UIs. A second key benefit is that the menu is always available and easily dismissable at will, allowing for spontaneous actions on the go. In light of this, our main objective is to understand what kind of features and applications are suitable for such a one-handed AR UI concept as formulated in the following research question:

How expressive can PalmGazer be, how many of the basic mobile interactions can be covered by one hand, and at which point is a task more suitable for a two-handed approach?

Our work takes a systems-oriented approach, where we design and implement a holistic UI that includes interaction techniques and AR application probes, forming a particular balance between expressive power, ease of use, and multimodal fusion. The holistic approach allows to gain insights into the low-level task- and application-specific design issues and trade-offs, and the high-level integration and compatibility as a whole. This is followed by an informal study, where users experience the system with simple *vs.* complex interactions and hand *vs.* head based reference frames.

We find that the holistic UI concept was, after a brief training phase, easy to use for the study participants. Interestingly, the ability to move the UI farther or closer to the eyes facilitates eye-tracking interactions as users can dynamically resize the visual target size. With regards to expressiveness, we find that all basic actions for selection and navigation are suitable, while higher degrees-of-freedom tasks become too challenging with one hand and gaze only. These findings contribute to the prior knowledge by proposing a novel approach to the class of fully one-handed interaction, and provision of a better understanding of the merits and limitations across a variety of usability factors.

Our contributions include four points. First, our design is grounded in an analysis of relevant eye-hand AR UI systems that expands the understanding of the fragmented nature of UI activation and interaction techniques in research and industry. Second, we present new interaction concepts as basis for PalmGazer, including a consistent set of selection and navigation techniques for 1D/2D/3D transformations, and the reference frames of *Above-Hand* and *On-Hand* with distinct ergonomics/visibility trade-offs offered as choices for users. Third, a prototype system showcased by 6 applications to demonstrate how expressive the input concept is, and to selectively

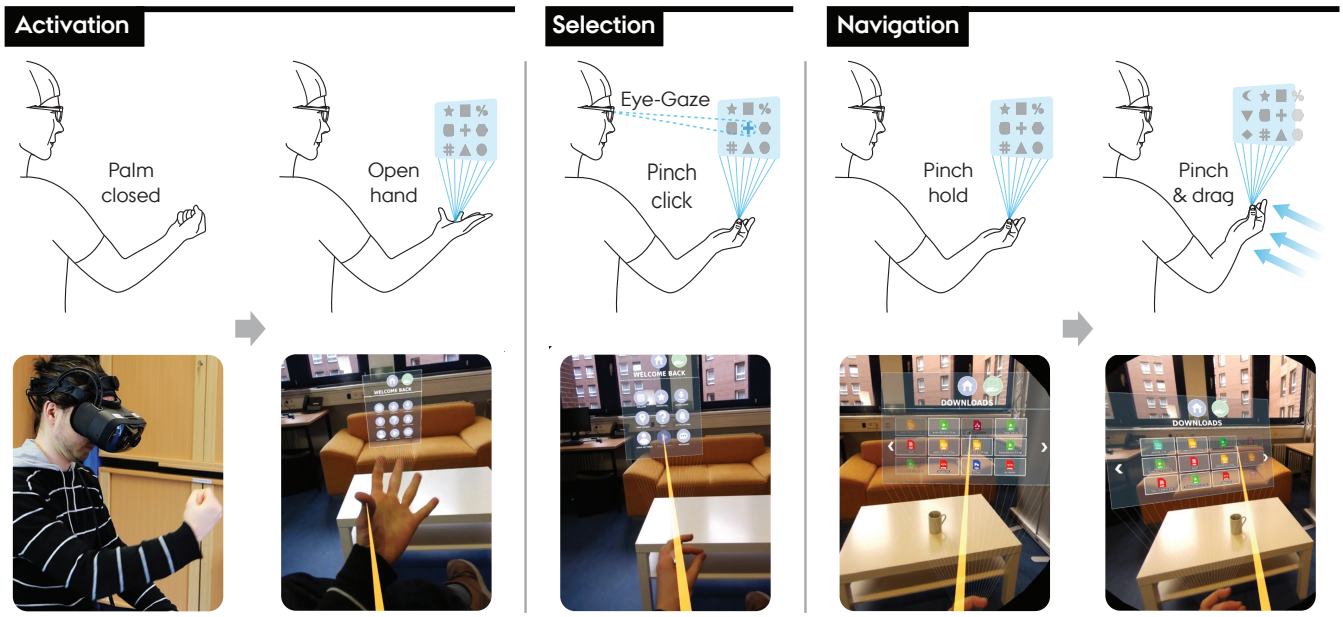

Figure 2: PalmGazer integrates three techniques to form a unimanual, on-demand UI: (a) UI activation can be performed by a palm-open gesture that summons the UI above the hand, allowing on-demand access by one hand when needed. (b) Selection is accomplished through gazing at the item of interest and a pinch gesture. (c) Navigation commands via pinch-dragging involve both content transformation and UI movement.

highlight interesting design decisions to take for system-wide and application-specific actions. Lastly, we present insights on concept and implementation from an informal evaluation of the system.

2 RELATED WORK

AR UIs can be put in 3D space along various frames of reference such as body-attached, hand-referenced, or references relative to the world. In this context, major challenges involve (1) UIs in the physical world can be easily left unnoticed and need time-consuming and physical effort to move them around, (2) menus staying continuously in the field of view obstruct the reality of the user, and (3) a button for activation can be distracting, difficult to hit, and the number of buttons on input devices is limited as well as adds an unneeded step to user interaction [24, 32].

A related interaction concept are adaptive UIs that exploit contextual information to automatically place and present the digital information to the user [2, 5]. Eye-tracking allows to infer attention to a particular widget in AR, based on a concept where virtual information of the widget reacts to the user's attention, and gradually expands the passive perception of information [6, 22, 23, 35, 41]. Such glanceable widgets are appropriate for simple notifications and information retrieval [41], but are less suitable for expressive menu interaction. Lu et al. highlight the need for specific activation techniques to render the UI more expressive and avoid cluttering the UI [22, 23], for which we provide a solution. In sum, context-aware and glanceable UIs can improve the user's efficiency in accessing information, pointing to the potential of AR to advance mobile interaction [6]. These efforts aim to provide implicit, hands-free interaction, while we focus on explicit one-handed interaction.

Eye-tracking as an input device is becoming established, and research efforts are increasingly exploring integration into menu UIs. For example, Ahn et al. have recently investigated *StickyPie* [1], where gaze enables scale-independent marking menus that overcome overshoot errors of regular gaze-based pie menus. Yi et al.'s *GazeDock* [53] employs gaze for activation of and item selection in a view-fixed peripheral menu, finding that 4-8 items leads to the highest throughput and that the method was preferred by users over the Dwell-time and Pursuits techniques. *Radi-Eye* by Sidenmark et al. uses gaze- and head-interaction in a world-fixed pop-up radial UI [46], where each circular level provides a submenu. In their studies, they find it affords fast and error-free interaction by nurturing on natural eye-head coordination. The concept of Gaze-Hand Alignment has been introduced by Lystbæk et al., where the mere spatial coordination between hand and gaze ray provides a fast selection mechanism for world-fixed menus in the environment [26]. These works show important milestones for gaze-enabled menu systems, which we extend by study of unimanual menus.

Eye-hand interaction can advance the manual input capabilities of a computer user, e.g. when interacting with the mouse and multi-touch displays [36, 54]. In 3D virtual environments, eye-hand interaction such as Gaze + Pinch [39], where the eyes select targets and the hands perform manipulations, has been studied in a few controlled settings with virtual and augmented reality devices for interaction with the presented 3D content. From that, a set of empirical evaluation papers have been published by researchers. These papers indicate that this type of input medium has advantageous qualities with regards to task completion times and physical effort

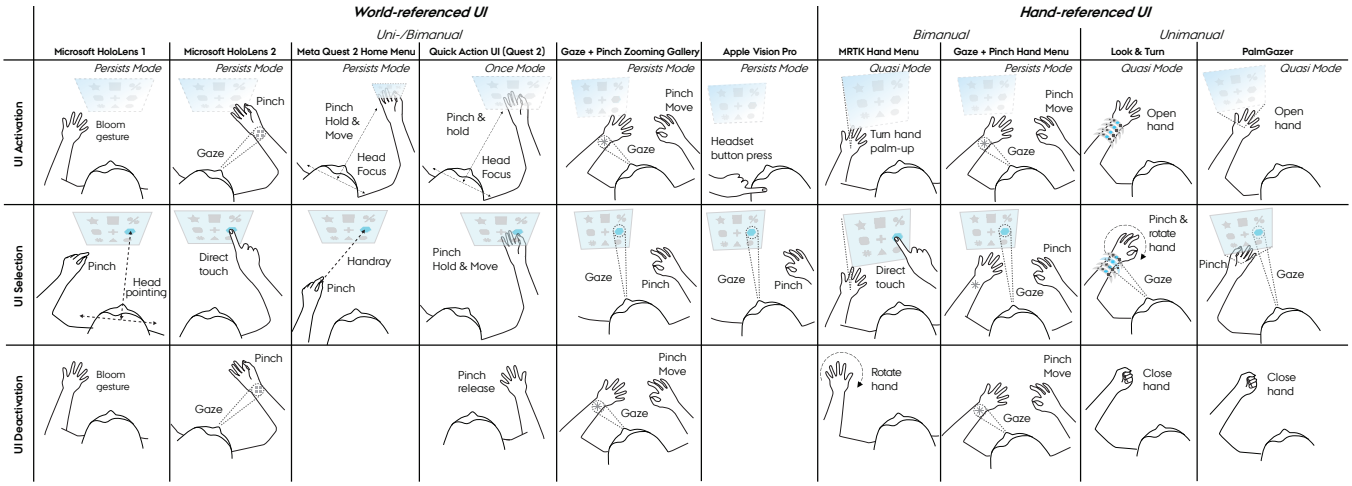

Figure 3: A comparison of AR menu UIs that involve hand and eye interactions to activate and interact with virtual content.

compared to hands-only or eyes-only UIs [20, 25, 26, 33, 51]. Pfeuffer et al. have proposed a set of concepts how eye and hand inputs can be used in combination in a 'gaze selects, hand manipulates' division of labour [39]. Two-handed VR menus [39] and one-handed menus for computer aided design have been studied [43]. We extend the prior art through an exploration of how eye-hand inputs can advance one-handed menu interaction.

In current AR HMDs (E.g., Meta Quest and Microsoft HoloLens series), developers have access to gaze and hand tracking and resulting interaction techniques (e.g., the *Gaze and Commit* of the Mixed Reality Tool Kit (MRTK) [31]). Apple's Vision Pro is using a multimodal eye, hand, and voice UI for control of their spatial computing operating system VisionOS, showing clear potential as a new input paradigm for AR. In PalmGazer, several interaction concepts overlap with Apple's UI, such as the selection and navigation actions. We note that our work is based on a master thesis submitted before the release of the Vision Pro [34], and our work focuses on handheld UIs as outlined next.

3 CHARACTERISATION OF AR UIS

UIs must support at least a way to activate the UI as well as a way to interact with its content. A set of modalities and techniques have been employed for those tasks in prior work. To shed light into this, we selected a set of closely-related systems, which means that they support hand and/or eye tracking, use primarily pinching as an intuitive and established hand tracking input [44] and offer 2D spatial windows for pointing. Given this, we categorise and discuss the approaches across the factors of *Mode Management*, *Reference Frames*, and *Handedness*, to clarify similarities and differences and lay out the design opportunity that PalmGazer tackles.

3.1 UI Mode Management

UI activation (and deactivation) techniques can be categorised into three types of UI mode management models [16]. Current XR systems employ a controller button press and hand gesture using a **Persists** model, where the UI persists at a specific place after activation. E.g., the Meta Quest 2 offers the Quick Action menu. Here the user holds a pinch gesture, which activates a small menu with a button to open the main menu. The HoloLens 1 uses a dedicated 'bloom' hand gesture (flexing all fingers out), to be performed each time for activation/deactivation. VEIA and the Microsoft HoloLens 2 operating system offer a persistent button at the hand's forearm for activation. The user can either directly pinch at it with the other hand, or employ Gaze & Pinch. The Apple Vision Pro, in contrast, uses a physical button on the headset. Explicit activation lends itself to suit longer sessions at a stationary place with world-referenced UIs. The Quick Action menu itself represents a **Once** model, as the UI opens once when holding a pinch for a set time, and closes when the pinch gesture is released. To avoid conflicting with default pinch gestures, the system employs a dwell-time during which the user holds the gesture at the centre of the FoV. This is useful for infrequent, one-off actions. In a **Quasimode** [42] model, the UI mode is active as long as the user maintains a constant kinesthetic action, that can reduce mode errors [45]. A commonly employed method that can considered a Quasimode is UI activation when the hand's palm is facing forward and the fingers are spread out [3, 14, 15, 39]. While palm-up works well for UI display, the same hand can still be used for object interaction when not held palm-up [3], representing a plausible implicit mode-switch for users.

3.2 Reference Frames

Many use cases include the stationary (e.g., at home, workplace, or at an exhibition), where digital content defaults in a fixed position (**world-referenced**) until people explicitly relocate or re-instantiate it. As exception, MRTK's tag-along behaviour also adapts the UI to the headset's position when it falls outside the FoV, and follows the user around. **Handheld** (or hand-referenced) UIs leverage the users' sense of proprioception to enable menus hiding at a place on the user's body that can be pulled out on demand and to allow users to deliberately hold and reposition the UI on-demand [11, 32, 52]. This enables having the UI flexibly 'at hand' and avoids potential occlusion of objects or persons in the vicinity and inconvenient automatic placements. Most closely to a mobile phone is a placement directly inside the palm and using the other hand for

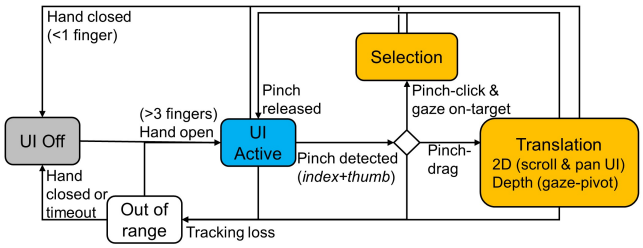

Figure 4: Input state model of PalmGazer, based on the user's hand posture and gestures sensed by the system.

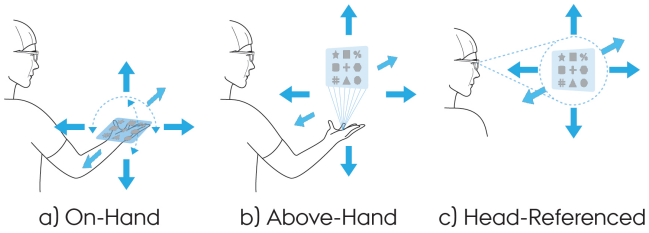

Figure 5: PalmGazer supports two hand-attached UI placement concepts, one that resembles smartphone use in the hand (a) and another where a UI 'hovers' well above the hand (b), compared to a typical head/world-referenced UI (c).

direct touch. However, with inside-out hand tracking, two hands can block each other in the camera's vision. As workaround, hand tracking SDKs [10, 30] offer UI panels next to the hand.

3.3 Handedness

World-fixed UIs are commonly used with for **one-handed interaction**. But hand-attached UIs seldom enable one-handed operation, as moving the UI and pointing at it clash when done with the same hand. To approach this, the interaction can be temporally or spatially multiplexed. With Meta Quest's Quick Access menu, users can summon the UI to their hand's location where it snaps to world space, and then use the same hand to point at a menu item. A long-pinch motion is employed as a mode switch to summon the UI, which needs to be performed each time a menu selection is desired. To afford parallel use, both activities can be spatially multiplexed. In the same hand, specific hand postures and fine-grained finger motions can be employed to increase expressiveness [13, 18, 52], but it departs from the simplicity of pinch [44, 49]. Instead, typically a second spatial pointer is enabled via the second hand. **Two-handed interaction**, aligned with Guiard's Kinematic Chain Model through an asymmetric division of labour [4, 12], has often been employed for 3D interaction [10, 15, 27, 30, 48, 55]. Considering gaze input, bimanual interaction has also shown compatibility with gaze pointing for symmetric and asymmetric constellations [37, 39, 40]. In only few cases, Gaze + Pinch has been employed for UIs on the same hand, i.e.in Look & Turn or in the HoloLens 2's activation method. The UI pointing can be offloaded to gaze, offering a new option to enable simultaneous usage of holding (by hand) and interacting (by gaze and pinch), as a core concept guiding PalmGazer's UI.

3.4 PalmGazer

Based on this characterisation of 3D UI systems, PalmGazer is a unique system that encapsulates (1) *quasimode UI mode management*, as users can rapidly activate or deactivate the UI by a hand-open or hand-close gesture; (2) *handheld reference frames*, to afford that users can place and reposition the UI at will as it is hand-attached; and (3) *uni-manual interaction*, where users can simultaneously hold and interact with the UI using the same hand.

4 DESIGN OF PALMGAZER

An overview of the input states and transitions is shown in Figure 4. The UI system integrates those tasks into a state model that as receives the user's gaze and hand-tracking event information and reacts accordingly. By default, the UI is in the *UI Off* state, but opening the hand will summon it. The most recently active application is shown within the UI, and the user can interact with it. Performing a *pinch-drag* gesture, e.g., for scrolling a list with overflowing elements, sets the system into the *Translation* state. Issuing a *pinch-click* command invokes the *Selection* state action, combined with the gaze directed at a target item. The user can then simply close the hand, which returns the system to the *UI Off* state.

4.1 Reference Frames: Hand, Above Hand, Head

PalmGazer can utilise three reference frames for eye-hand menus. Close to a smartphone is the **On-Hand** placement (Figure 5a). As UIs on the hand render overlapping hands difficult to track, they are typically not recommended [30]. Yet, via our uni-manual concept, we avoid this issue altogether. The On-Hand placement locates the UI slightly above the hand's palm centre and orients the UI according to the hand's palm. The user can shift the UI to a comfortable viewing position by moving their hand. The major benefit of its similarity to a smartphone, where the UI is always available in hand, is trading with the potential of ergonomic viewing and interaction posture. If the hand is held at the height at the field of view, prolonged use can lead to arm fatigue; if the hand is held lower (e.g., at waist level), arm fatigue is reduced, but a less ergonomic neck posture may be the case [7, 50].

As an alternative, we propose the **Above-Hand** placement (Figure 5b). The 'summon' type of hand-attached UI that floats at a distance above the hand is inspired by science fiction movies where it has been intuitively leveraged for the presentation of holographic data (e.g., Iron Man, Loki, or Star Wars). We apply a similar adaptation policy to On-Hand, except that the UI is raised 30 cm above the hand, with an additional offset of 15 cm away from the user's hand in direction away from the user's head. This enables users to retain their hand in a natural posture close to their waist while maintaining a good view of the UI within their field of view. Lastly, we also note that while we focus on hand-attached UIs, the PalmGazer interaction technique can be employed for **Head-Referenced** (Figure 5c) or world relative UIs, for a different usage scenario where it strikes a different trade-off between UI and real-world visibility, as it remains in place when moving.

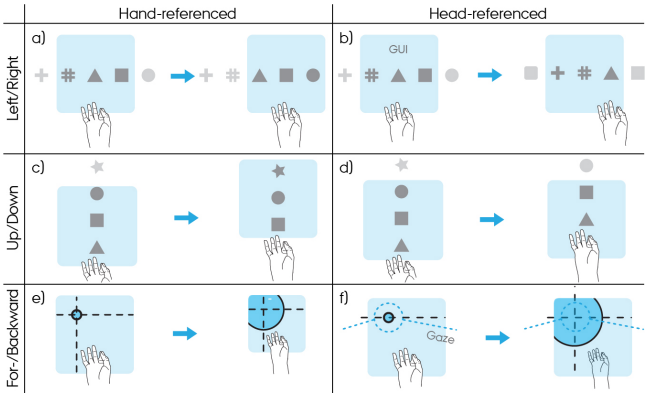

Figure 6: UI navigation integrates dynamic peephole metaphors to address the conflict of UI frame vs. content panning with the same hand (left), using a different mapping than existing head/world-reference frame mapping (right).

4.2 2D Navigation: Static and Dynamic Peephole

Navigating the UI should be as simple as on a smartphone where a swipe allows the user to see the next page or pan an image. We employ an analogous principle: when holding a pinch gesture, horizontal or vertical hand movement translates directly to a scrollable or draggable UI (Figure 6a-d). An interesting aspect is how directional hand motion is mapped to manipulation parameters of the UI depending on the UI placement. We consider a metaphor that the UI represents a peephole view into the virtual world allowing to retain the spatial relationships of the virtual content [9, 28]. A head or world-referenced UI is more suitable for a metaphor of a **static peephole UI**, as the UI position is independent of hand motion (b, d). A hand-referenced UI is different, as when conducting a dragging gesture, the UI would move with it to sustain the spatial anchor to the hand. Here we reverse the translation direction, rendering the interaction a **dynamic peephole window**. This means the UI moves with the hand, and acts like a window to a larger virtual space behind it (Figure 6a, c).

4.3 Depth Navigation: Discrete vs. Continuous

Depth navigation is a feature in many applications of **discrete** (e.g., a hierarchical tree structure) and **continuous translation** (e.g., zooming a map). Much like with 2D navigation described above, users can hold the pinch gesture and move their hand in the forward and backward direction from the perspective of the user. The head-referenced variant is akin to MRTK's Gaze-supported pan and zoom, where a single hand pinch gesture allows zooming a map window at the gaze pivot [21, 29]. We extend this by a more detailed assessment of hand versus head/world-referenced UIs. There is essentially a trade-off: As apparent in Figure 6e, performing the zooming operation will also visibly reduce the size of the UI as it gets farther away.

In contrast, in a head/world referenced UI (Figure 6f), the hand operation occurs in an independent space to the UI, and can move in front or behind the UI. In both ways, the user operation includes clutching in case of multiple zoom operations; and integration of a control-display ratio is relevant. However, this can highly depend on the application use case. For continuous zooming operations, more fine-grained translation functions must be implemented; for discrete operations (e.g., a folder structure with 3-5 levels), short hand motions can be sufficient.

Table 1: PalmGazer applications use gaze to point, as it is translated into (X,Y) UI coordinates, accompanied by a pinch-click gesture to select. Use of pinch-drag in one or more directions (X-horizontal, Y-vertical, Z-depth) is application-specific.

Application	Gaze			Pinch-click	Pinch-drag		
	X	Y	Z		X	Y	Z
Main menu	×	×		×			
Favorites Folder	×	×		×			
Notifications	×	×		×			
Music Player	×	×		×			
Downloads Folder	×	×		×	×		
Image Gallery	×	×					×
Map Viewer	×	×		×	×	×	×

5 APPLICATIONS

We now explore design issues and opportunities for (1) system-wide application of the PalmGazer concept in a concrete system implementation, and (2) specific applications by way of technology probes [17]. An overview of the supported applications, and how each modality is utilised is presented in Table 1.

We used the Varjo XR-3[1] head-worn display with its integrated capabilities for pass-through AR, eye tracking (200 Hz, 1° reported accuracy) and Ultraleap hand tracking.The software was implemented in Unity. The *On-Hand* UI variant fixes the orientation in relation to the hand, and the UI is positioned just above the palm position of the monitored hand (by 4.5 cm). The UI may pick up on our natural hand motion and hand-tracking jitter. Smoothing the position helps, and through testing, we found that a parameter of 100 ms at a frame rate of 90Hz provided a good balance between UI stability and the delay in UI movement. For the *Above-Hand* variant, given the hand-tracking quality, it is more susceptible to hand jitter as brief hand motions can cause large UI displacements. To counteract this effect, the UI ignores the hand rotation and always faces the user regardless of its position. This allowed to flexibly move the UI in or out of the user's view through hand movement along all three axes while remaining relatively stable to be used regardless of the hand rotation. Lastly, a baseline *Head-referenced* is included where the UI adapts its position to 55 cm in front of the forward vector of the headset user, oriented towards the user. Item selection by gaze and pinch includes a default, hover (visual highlighting of icon background), and selected state. When buttons are close to each other, a single gaze point can theoretically fall into the empty space in between and void a selection intended by the user. To approach this, we assume if a gaze is within the UI bounds, a user is focusing on an interactive element, and hover/selection

[1]Varjo XR-3, https://varjo.com/products/xr-3/, accessed 04/02/2023

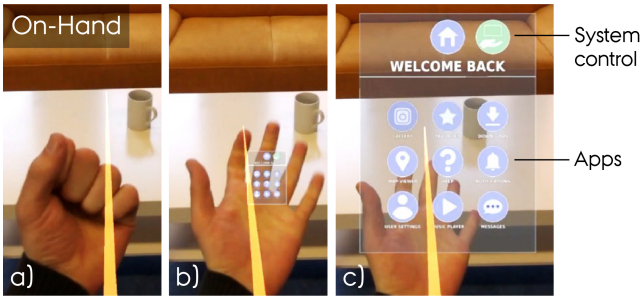

Figure 7: UI Activation by a gradual summoning animation (a-c) opens the home menu. Note: The orange ray indicates the user's gaze across all figures (not shown in real use).

automatically snap to the closest target (not for non-interactive targets, e.g., when viewing images in the gallery).

5.1 Home Menu

The home menu allows to choose applications, which are then opened in individual UI windows. It is opened by a gradual scale-up animation to provide a smooth transition (Figure 7a-c). It offers a set of applications in the main UI area, and a fixed menu area at the top. The system remembers the last used application and will resume the state when the UI reactivates.

5.1.1 Implementational Details. The item size for each app icon is 4.5 cm. An additional top bar menu has been designed with a few buttons of size 3.8 x 3.8 cm. The menu is active across applications, allowing the user to return to the home menu on demand. For this, a dedicated button is reserved in the menu. A second button allows toggling the reference frame between *On-Hand*, *Above-Hand*, and *Head-Referenced*, to offer users the choice (Figure 7). Setting this option will adopt the techniques for navigation to the respective reference frame as elaborated in section 4.2.

5.2 Music Player App

The music player supports standard control buttons such as play, pause, and next, as well as a window of songs at the lower part of the UI window. Song item button sizes were 2.1 x 3.5 cm. It demonstrates the quickest to perform action in PalmGazer. To play a song, only a 'palm-open, look at song, and pinch' input sequence (assuming the music player is still open from last time of use) – which, over time, appears like one cognitively unified interaction chunk, rather than a tedious sequence of operations.

5.3 Notifications App

This application provides an overview of notifications in a vertical list that users can quickly browse and respond to. It demonstrates in-page context menus that are integrated within each notification element (Figure 8a). By default, a notification element displays the information on application, title, and description. At selection, each notification element transforms into a context menu with three buttons to react to it (check, postpone, delete) (Figure 8b-c).

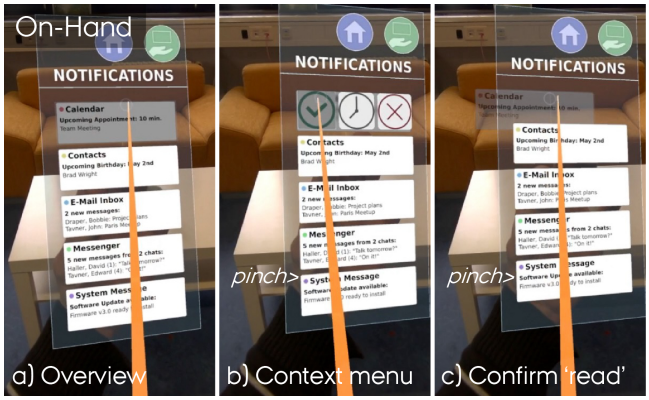

Figure 8: Notifications allow opening a context menu after selection (a-b), and a 'swipe' for a quick response (c).

5.3.1 Implementational Details. We tested persistent buttons next to the notifications as well, but found these to be needing too large spaces for robust selection and also they reduced the amount of information possible to be read on the notification UI element. Instead, we use the aforementioned two-selection model where the notification is opened first and then processed. An alternative to the default selection technique of the response to the notification would be to adopt shortcuts similar to touchscreen UIs. In the information view, the user can look at a notification and perform a pinch-drag-left gesture to immediately perform the 'check' reply (or a pinch-drag-right gesture for 'delete'). This variant provides extended expressive power, as multiple UI commands can be mapped to each logical dragging direction, similar to gaze and pinch based pie and marking menus [39].

5.4 Downloads, Favorites

Both Downloads and Favorites applications are standard 2D grid UIs that we started with as baselines. These demonstrate that the many potential 2D grid based applications can be easily supported. As an example, 'Downloads' includes a large number of files to demonstrate a horizontal scrollable grid view. An example of an Above-Hand scroll technique is shown in Figure 9. Here, a user can browse through the downloaded files only by a pinch-drag gesture using a dynamic peephole view. To see more content on the right side of the grid, the UI is held at the left side of the field of view. Then, a pinch gesture is initiated from left (Figure 9a) to the right (Figure 9b-c). Afterwards, the user can select a file (Figure 9d) and see details of it in the detail view. A back button, that appears in the detail view, is implemented to return to the grid (Figure 9e).

5.4.1 Implementational Details. We considered alternative ways of interaction. The user can scroll without moving the hand, but by holding a pinch gesture and moving the own field of view (i.e., turning the head). Then, the UI scrolls equally as by hand. In principle, both ways can be provided to users as they are complementary, and users get the choice. Item sizes were set to 6.2 x 3.5 cm.

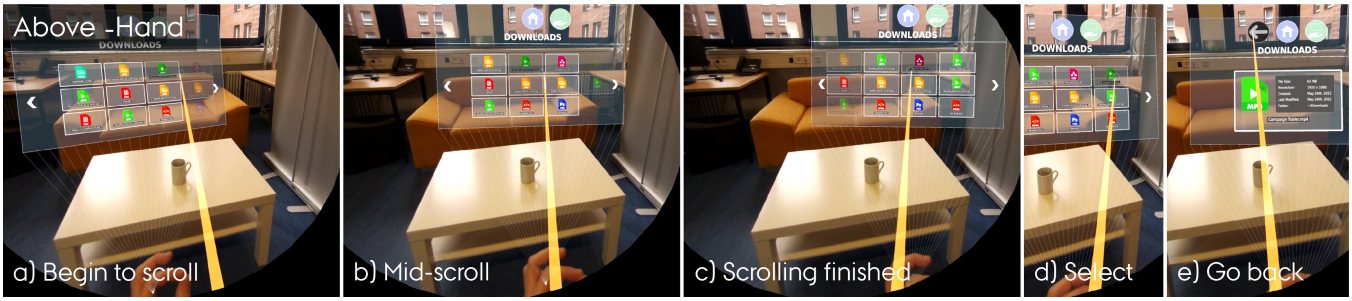

Figure 9: The downloads app integrates a grid menu. Pinch-drag gestures scroll the content (a-c), parallel to moving the UI with the hand position. The user can immediately select (d) and view (e) files, and return via the clicking back (e-top).

5.5 Image Gallery Application

This is a grid UI where users can scroll in vertical/horizontal directions (more pictures and folders), but also in-depth (overview, folder, picture) that we highlight in this probe. It offers the user a method to rapidly view and traverse a large set of images in multiple hierarchy levels, with currently a depth or 3 layers supported.

5.5.1 Deep-Gallery Interaction Technique. Navigating into depth can be performed in two ways. First, like the Downloads app, the user can select an item to traverse to the next depth level. The user performs a selection command each time. Second, a pinch-drag gesture can be performed in the for-/backward direction. We virtually place the hierarchical layers 5 cm apart from each other and map the hand movement linearly to traverse these layers. This has an interesting appeal as it speaks to a metaphor of traversing depth levels, mapped to depth motion of the hand. The user can initiate a pinch gesture and hold it in the folder overview (Figure 10a), then look at the folder and move the hand forward to open the image overview (Figure 10b), and then look at an image and move the hand further forward to open the image (Figure 10c).

5.5.2 Implementational Details. We tested thresholds for the distance between 2 layers, and found that 5 cm provide a robust experience to hold the hand in a layer, but also easily shift to the next while focusing gaze on the appropriate targets. This highlights a trade-off between manual effort and hierarchical data traversal. For basic applications (e.g., Downloads), a single pinch-select gesture is sufficient and avoids UI repositioning. For multiple hierarchy levels, using a single pinch-drag-depth technique can become beneficial.

5.6 Navigation App for Map Interfaces

Enlarging the area of visual attention can be a natural and accurate method for zooming in map navigation [19, 36, 37, 47]. *Map Viewer* supports spontaneous access to geographical information and rapid traversing of the map. The prototype defaults to a full world map, at which the users can begin to navigate. It supports sequential as well as simultaneous pan and zoom operations, providing the necessary set for navigation like on multi-touch based systems.

5.6.1 Eye-Hand Pan-Zoom Interaction Technique. Users can pan with a pinch-drag gesture without eye-tracking (Figure 11a-b). To zoom, users gaze at an area of interest and and can zoom into it by performing a pinch-drag toward from the UI, respectively (Figure 11c). Similar to the *Downloads Folder*, the direction of panning is adjusted based on the placement of the menu: In a *Head-referenced* mode, the content moves in the same direction as the hand, with the direction being reversed in the hand-referenced modes. To zoom in or out, depth navigation in form of continuous translation is used concurrently to 2D navigation via a dynamic peephole view.

5.6.2 Implementational Details. Panning uses a control-display gain of 1:1, which is intuitive as if dragging the content (in the On Hand reference frame). Zooming transforms hand motion to zoom in/out of the map with factor 2, i.e., more physical motion is applied to a zoom level of the image. We find that this allows to subsequently perform pinch dragging gestures of a length of approximately 5-20 cm to operate all typical pan and zoom operations. Zoom-out happens at pinch-drag backward, however instead of using the gaze pivot, the center of the map is used as the eyes are likely not indicative of the exact desired zoom-out position.

6 INFORMAL USER EVALUATION

A qualitative evaluation explores the usability of the PalmGazer prototype with a focus on the core concepts: reference frames and interaction techniques. Every user experienced the *On-Hand*, *Above-Hand*, and *Head-referenced* placements. Each UI involved interactions with the Favorites Folder, Music Player, Image Gallery, Map Viewer applications. In each application, users received tasks to support a goal-directed experience, displayed on a virtual board. In the *Favorites Folder* app, participants were given a filename and a property of that file (e.g., "Flyer.pdf", File Size property), and were instructed to find, view, and read out a file detail (file randomised, as in all tasks). In the *Music Player*, participants were given a track title to play. For the *Gallery*, one image was overlaid with a three-digit number. Participants were given the album name and instructed to find the image. In the *Map Viewer*, the map was modified to contain three red circles, each marking a certain area on the map. Inside one of them, a three-digit number, displayed small enough to be only visible when zoomed in, was the search goal.

Users were first briefed, filled out consent and demographic forms, and then began the training. After showing and explaining the interactions, participants trained until they felt confident and familiar enough to perform the tasks on-demand. Then, the tasks began. The active application and UI placement were set beforehand

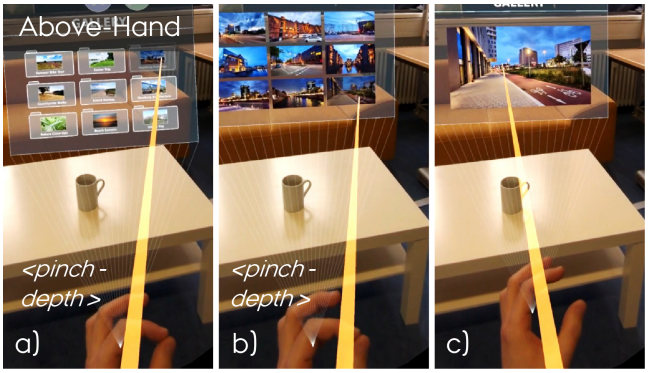

Figure 10: The Gallery involves a 3-level hierarchy traversed by gaze pointing and pinch for-/backward motion.

to ensure correct procedure. Users were free to take short breaks between apps. After each round, participants provided written and/or verbal feedback. Users interacted with each application on average for 32.5 seconds (SD=7.4). Then, users were presented with the option to freely explore the system. Eight participants did so, spending between 3 and 28 minutes (M=14, Mdn=9). Lastly, an open-ended interview was conducted, with questions about the overall usability, the UI activation, gaze-pinch clicks, gaze-pinch dragging actions. The study lasted around 90 minutes on average.

18 paid persons (10 male, 8 female) took part via a university mailing list, messaging platforms, and personal contacts. All were right-handed and 20–40 years old (mean = 26.6, SD = 4.5). One wore glasses underneath the headset and three wore contact lenses. Participants rated their experience on scales from 0 (no experience) to 4 (expert), as low to moderate with VR/AR (M=1.8, SD=1.1), eye tracking UIs (M=1.3, SD=1.3), and mid-air gestures (M=0.6, SD=0.8).

6.1 Results

PalmGazer was overall well received, even though we encountered issues with more complex operations and hand-tracking constraints. P15 stated "*It is very cool!*" and P5 stated similarities to smartphones: "*I can see how it can be used in the future instead of smartphones or in addition to them.*" UI activation and selection was received well by all users (described as '*intuitive*', '*easy*', '*fun*', or '*magical*'). Users found specific functions useful, such as changing the gaze selection during a longer pinch-drag interaction, either for quickly and continuously going through different images or for zooming in and out of different portions of the map by holding the pinch gesture while moving the hand back and forth.

Pinch-drag interactions were called user-friendly and related to traditional touchpad gestures, e.g. P7: "*A nice tool, and with a bit of adapting to it, it works quite well.*" On the other hand, P10 stated: "*I like the idea, but I find it hard to control it how far you have to move the hand*", while P7 pointed out that it needs getting used to and P16 called it "*very extraordinary in the beginning*". This aspect was also mentioned by P8, who stated: "[I]*never thought about an action like that, but once you know it, it works great*". The combined use of UI positioning and content alteration needs more getting used to than the selection principle, before it worked well.

Figure 11: The Map Viewer has continuous pan & zoom, via a pinch-dragging into the gaze-based zooming pivot.

Gaze and pinch-drag-based depth navigation was more positively received for the Gallery, which used only movement along one axis than for the Map Viewer. As stated by P1: "*The dragging back and forth was very smooth and exact. Mixed with the dragging to the left and right* [to see maps], *it was hard to handle.*". In addition, P15 noted: "*I like it, particularly for the gallery. For the map app it was a bit more demanding to figure out how to do it properly.*"

Participants liked the idea of "*holding*" the contents of the UI on their hand in the On-Hand condition. For example, P5 compared this to holding a smartphone and stated that they liked this placement concept the best, although they found it more difficult to use as of issues with tracking. Specifically for the Gallery app, the idea of holding the UI with an image On-Hand and being able to show it to others in a shared AR scenario was also mentioned by two participants as a positive, although they both added that it needs a more stable hand tracking. P1 and P7 reported for hand-attached UIs that small hand movements such as executing a pinch can shift the UI, possibly interfering with the user's gaze selection.

The eyes can naturally follow the hand, which was different across placements. With the Head-Referenced and Above-Hand UIs, the eyes can lead to a hand-following behaviour which detracts from the main UI. E.g., *P1* stated: "*When I move my hand, my first intuition is to look at my moving hand, but then I lose track of the menu and I have to re-orientate.*". For the On-Hand UI, P5 mentioned that the UI is positioned where their eyes wanted to look naturally and P14 pointed out that this also fits with panning the map by moving the hand with the UI on it, where the eyes can follow the hands. A potential reason is user's familiarity from smartphones.

7 DISCUSSION

The PalmGazer UI brings together many different design dimensions, interaction trade-offs, and usability aspects. Here we discuss our insights gained from the design, implementation, and evaluation of the system.

Simultaneous and Interchangeable Eye-Hand Interaction. Users have two options for pointing at a button for hand-referenced UI versions. They can either look at the UI button and pinch with their hand. Conversely, they can glance at a location in the UI (which is not the desired button) and move their hand, which moves the UI, to bring the desired button into the gaze region. This interchangeability of modalities is useful to provide more alternatives, balancing the interaction burden between modalities, accommodating possible gaze error, and supporting apps with small targets.

Eye tracking accuracy needs careful target size considerations and may be user-specific. A portable UI, on the other hand, allows the user to achieve the appropriate visual target size because

bringing the target closer increases size implicitly. Adapting UI orientation to the hand, similar to smartphones, appears natural at first glance. However, because rotating the UI targets reduces visual target size, it can also have a negative influence on ocular input quality. Auto-rotation towards the user is a practical solution to this problem. Overall, by making accuracy an implicit user-orchestrated approach, it simplifies interacting with the eyes and hands.

Division of Labour Between Modalities and Tasks. Tracking hand gestures beyond a standard pinch gesture becomes increasingly difficult for the hand tracking sensor. The combination of *palm open* to activate and *eye-hand* input to interact while keeping the hand open provided a practical trade-off between interaction expressiveness and tracking accuracy. While we first assumed it would be utilised mostly when the palm is looking up or towards the user, it was frequently used when the palm was facing away from the user. This enabled superior pinch tracking because the two fingers were more visible, and also afforded a more ergonomic interaction.

Supporting fundamental 2D interactions (selection, scrolling) allows for rapid activities when mobile (changing songs, checking notifications, etc.). The 3D interaction tasks presented us with a different image. One-directional and basic actions, such as discrete navigation in depth through a *pinch dragging forward or backward*, were easy to use, such as in our Gallery app, where users leap to the next hierarchical level. More degrees of freedom, such as integral, continuous 3D pan and zoom in map navigation, were challenging and clear pointers for future improvement, as coordination of the degrees of freedom while holding the UI overwhelmed the users.

Uni- vs. Bimanual Control. What are the limits to one hand only? It was harder to control our most complex applications such as map navigation, where the two modalities are mixed for pan & zoom. Such issues are reminiscent of interaction constrains of one-handed interaction with smartphones: performing an advanced operation such as zooming with the gripping hand is sub-optimal. But, this issue must be viewed in context. One-handed input is a common technique to engage with many smartphone activities. Similarly, our research provides clear indicators that PalmGazer is expressive enough to invoke a wide range of common input commands.

UI Navigation. Our UI methods have shown to be effective in providing expressive power comparable to one-handed interaction with a smartphone. A home button in the header of the menu lets users to utilise *gaze and pinch* to swiftly transition from one app to the home menu to a new app, making OS-level navigation effortless. In order to navigate within the app, *pinching and dragging* (without eye-tracking) is appropriate for scrolling, panning, and swiping. A key interaction challenge is that users may use hand movement to both position the UI as well as manipulate UI information. Standard world-referenced input mappings are not suitable by design - for which we implemented a peephole metaphor [9, 28]. We did not find any issues with its usability and users were generally positive about the various scrolling and panning tasks, thus we recommend its use to establish an intuitive navigation experience.

Reference Frame Trade-Offs. The On-hand frame necessitates the user either moving the UI into view or moving the FoV to the hand, resulting in physical weariness. However, because the UI is directly associated with the hand in space, it has a stabilising effect to the UI in a position. In contrast, the Above-Hand UI, in theory, addresses the previous issue since users may drop their hand to a convenient position while the UI is in a comfortable view location. We found, however, that it is less stable since little hand jitter translates to large UI motion effects. Adding system choices that allow users to switch between reference frames to obtain the best of all worlds depending on the scenario can be suitable, e.g., a quick 'lift-up' gesture to 'throw' the UI from On-Hand to Above-Hand, or even employing adaptation algorithms that takes the user's reachability and visibility constraints into account [2].

8 LIMITATIONS AND FUTURE WORK

We did not conduct a formal evaluation of the user performance, useful to better understand the strengths and weaknesses in contrast to existing AR UIs. Future studies can include 1) measuring the time it takes to discover and open an app while on the go, 2) potential trade-offs between interaction complexity and mental load across the different included techniques, and 3) the effects of switching between eye-hand and hands-only control. Notably, our system's concept of integrated selection and navigation has been recently introduced in the context of the Apple Vision Pro UI. The forthcoming release of this product might shed more light on how consumers perceive and embrace this part of the UI, and uncover new challenges to tackle in future.We focused on smartphone-like one-handed interaction mainly for tasks of pointing and navigation. Other tasks such as text entry [25] can be just as relevant and it is open how to offer uni-manual control. Furthermore, our research holds applicability beyond smartphone apps, such as computer-aided design to provide users quick, on-demand access to all those menus and tool palettes. PalmGazer can extend to VR and 3D UIs, for example as a convenient avenue for system menu access.

9 CONCLUSION

We designed, implemented, and evaluated PalmGazer, a new one-handed hand-attached UI for spontaneous interaction in everyday AR. We demonstrated this through a system prototype that offers input capabilities similar to modern smartphones – to select, scroll, navigate, and interact with mobile apps fluidly with one hand. The limits in the user's expressiveness are indicated in more complex operations, where users must coordinate multiple dimensions in parallel, indicating room for improvements or potentially the border to employ hands-only. But the fundamental interaction principle is promising as a diverse, easy-to-use interaction style, facilitating quick actions through a spontaneously activated UI.

In contrast to hand-based HCI, the work for eye-hand UIs is at its infancy. With current industrial devices becoming just ready for a novel eye-hand interaction paradigm for consumers, it becomes ever more important to explore the vast design space of eye-tracking advancements in all those contexts that have been reserved for hands only. Our work expands the research line for gaze and hand based UI systems [36, 38, 39], and informs the design of multi-modal, flexibly-placed, on-demand AR UIs for a future where our mobile life spans far beyond holding a mobile device every time we want to interact, toward non-invasive but always-available UIs to seamlessly engage with digital content on the go.

REFERENCES

[1] Sunggeun Ahn, Stephanie Santosa, Mark Parent, Daniel Wigdor, Tovi Grossman, and Marcello Giordano. 2021. StickyPie: A Gaze-Based, Scale-Invariant Marking Menu Optimized for AR/VR. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 739, 16 pages. https://doi.org/10.1145/3411764.3445297

[2] Joao Belo, Mathias N. Lystbæk, Anna Maria Feit, Ken Pfeuffer, Peter Kán, Antti Oulasvirta, and Kaj Grønbæk. 2022. AUIT - the Adaptive User Interfaces Toolkit for Designing XR Applications. In *Proceedings of the 35nd Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3526113.3545651

[3] D.A. Bowman and C.A. Wingrave. 2001. Design and evaluation of menu systems for immersive virtual environments. In *Proceedings IEEE Virtual Reality 2001*. IEEE, 149–156. https://doi.org/10.1109/VR.2001.913781

[4] Bill Buxton, Abigail Sellen, and Yves Guiard. 2008. Two-handed input in human-computer interaction. *Haptic Input* 11 (2008).

[5] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. SemanticAdapt: Optimization-Based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 282–297. https://doi.org/10.1145/3472749.3474750

[6] Shakiba Davari, Feiyu Lu, and Doug A. Bowman. 2022. Validating the Benefits of Glanceable and Context-Aware Augmented Reality for Everyday Information Access Tasks. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 436–444. https://doi.org/10.1109/VR51125.2022.00063

[7] Jack T. Dennerlein. 2015. The state of ergonomics for mobile computing technology. *Work* 52 (2015), 269–277. https://doi.org/10.3233/WOR-152159 2.

[8] Ajoy S. Fernandes, T. Scott Murdison, and Michael J. Proulx. 2023. Leveling the Playing Field: A Comparative Reevaluation of Unmodified Eye Tracking as an Input and Interaction Modality for VR. *IEEE Transactions on Visualization and Computer Graphics* 29, 5 (2023), 2269–2279. https://doi.org/10.1109/TVCG.2023.3247058

[9] George W. Fitzmaurice, Shumin Zhai, and Mark H. Chignell. 1993. Virtual Reality for Palmtop Computers. *ACM Trans. Inf. Syst.* 11, 3 (jul 1993), 197–218. https://doi.org/10.1145/159161.159160

[10] Ultraleap for developers. 2021. UI Panels. Website. Retrieved April 2, 2023 from https://docs.ultraleap.com/xr-guidelines/Components/ui-panels.html

[11] Chelsey Nicole Gobeli. 2018. *The Effects of Reference Frames on 3D Menus in Virtual Environments*. Ph. D. Dissertation.

[12] Yves Guiard. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action. *Journal of Motor Behavior* 19, 4, 486–517. https://doi.org/10.1080/00222895.1987.10735426

[13] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwype: Word-Gesture Typing Using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300244

[14] Chris Harrison, Shilpa Ramamurthy, and Scott E. Hudson. 2012. On-Body Interaction: Armed and Dangerous. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction* (Kingston, Ontario, Canada) *(TEI '12)*. Association for Computing Machinery, New York, NY, USA, 69–76. https://doi.org/10.1145/2148131.2148148

[15] Zhenyi He and Xubo Yang. 2014. Hand-Based Interaction for Object Manipulation with Augmented Reality Glasses. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Shenzhen, China) *(VRCAI '14)*. Association for Computing Machinery, New York, NY, USA, 227–230. https://doi.org/10.1145/2670473.2670505

[16] Ken Hinckley, Francois Guimbretiere, Patrick Baudisch, Raman Sarin, Maneesh Agrawala, and Ed Cutrell. 2006. The Springboard: Multiple Modes in One Spring-Loaded Control. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/1124772.1124801

[17] Hilary Hutchinson, Wendy Mackay, Bo Westerlund, Benjamin B. Bederson, Allison Druin, Catherine Plaisant, Michel Beaudouin-Lafon, Stéphane Conversy, Helen Evans, Heiko Hansen, Nicolas Roussel, and Björn Eiderbäck. 2003. Technology Probes: Inspiring Design for and with Families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. Association for Computing Machinery, New York, NY, USA, 17–24. https://doi.org/10.1145/642611.642616

[18] Haiyan Jiang, Dongdong Weng, Xiaonuo Dongye, and Yue Liu. 2022. PinchText: One-Handed Text Entry Technique Combining Pinch Gestures and Hand Positions for Head-Mounted Displays. *International Journal of Human–Computer Interaction* 0, 0 (2022), 1–17. https://doi.org/10.1080/10447318.2022.2115333

[19] Konstantin Klamka, Andreas Siegel, Stefan Vogt, Fabian Göbel, Sophie Stellmach, and Raimund Dachselt. 2015. Look & Pedal: Hands-Free Navigation in Zoomable Information Spaces through Gaze-Supported Foot Input. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction* (Seattle, Washington, USA) *(ICMI '15)*. Association for Computing Machinery, New York, NY, USA, 123–130. https://doi.org/10.1145/2818346.2820751

[20] 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

[21] HiMER Lab. 2020. Demo: Eye Tracking Interaction Example - MRTK, Hololens 2. Video. Retrieved April 2, 2023 from https://youtu.be/UPH4lk1jAWs?t=21

[22] Feiyu Lu, Shakiba Davari, and Doug Bowman. 2021. Exploration of Techniques for Rapid Activation of Glanceable Information in Head-Worn Augmented Reality. In *Symposium on Spatial User Interaction* (Virtual Event, USA) *(SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 14, 11 pages. https://doi.org/10.1145/3485279.3485286

[23] Feiyu Lu, Shakiba Davari, Lee Lisle, Yuan Li, and Doug A. Bowman. 2020. Glanceable AR: Evaluating Information Access Methods for Head-Worn Augmented Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 930–939. https://doi.org/10.1109/VR46266.2020.00113

[24] Feiyu Lu and Yan Xu. 2022. Exploring Spatial UI Transition Mechanisms with Head-Worn Augmented Reality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 550, 16 pages. https://doi.org/10.1145/3491102.3517723

[25] Mathias N. Lystbæk, Ken Pfeuffer, Jens Emil Sloth Grønbæk, and Hans Gellersen. 2022. Exploring Gaze for Assisting Freehand Selection-Based Text Entry in AR. *Proc. ACM Hum.-Comput. Interact.* 6, ETRA, Article 141 (may 2022), 16 pages. https://doi.org/10.1145/3530882

[26] Mathias N Lystbæk, Peter Rosenberg, Ken Pfeuffer, Jens Emil Grønbæk, and Hans Gellersen. 2022. Gaze-Hand Alignment: Combining Eye Gaze and Mid-Air Pointing for Interacting with Menus in Augmented Reality. *Proceedings of the ACM on Human-Computer Interaction* 6, ETRA (2022), 1–18.

[27] Daniel P. Mapes and J. Michael Moshell. 1995. A Two-Handed Interface for Object Manipulation in Virtual Environments. *Presence: Teleoperators and Virtual Environments* 4, 4 (11 1995), 403–416. https://doi.org/10.1162/pres.1995.4.4.403

[28] Sumit Mehra, Peter Werkhoven, and Marcel Worring. 2006. Navigating on Handheld Displays: Dynamic versus Static Peephole Navigation. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (dec 2006), 448–457. https://doi.org/10.1145/1188816.1188818

[29] Microsoft. 2022. Eye-supported navigation in MRTK. Website. Retrieved April 2, 2023 from https://learn.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/mrtk2/features/input/eye-tracking/eye-tracking-navigation?view=mrtkunity-2022-05#gaze-supported-pan-and-zoom

[30] Microsoft. 2022. Hand menu. Webpage. Retrieved September 7, 2022 from https://docs.microsoft.com/en-us/windows/mixed-reality/design/hand-menu

[31] Microsoft. 2023. Gaze and commit. Website. Retrieved April 2, 2023 from https://learn.microsoft.com/en-us/windows/mixed-reality/design/gaze-and-commit

[32] Mark R. Mine, Frederick P. Brooks, and Carlo H. Sequin. 1997. Moving Objects in Space: Exploiting Proprioception in Virtual-Environment Interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., USA, 19–26. https://doi.org/10.1145/258734.258747

[33] Aunnoy K Mutasim, Anil Ufuk Batmaz, and Wolfgang Stuerzlinger. 2021. *Pinch, Click, or Dwell: Comparing Different Selection Techniques for Eye-Gaze-Based Pointing in Virtual Reality*. Association for Computing Machinery, New York, NY, USA, Chapter 15, 7. https://doi.org/10.1145/3448018.3457998

[34] Jan Obernolte. Master Thesis, Submitted 30-09-2022. *Exploring Gaze and Gesture Interactions in Augmented Reality*. Ludwig-Maximilians-Universität München, Munich, Germany.

[35] Ken Pfeuffer, Yasmeen Abdrabou, Augusto Esteves, Radiah Rivu, Yomna Abdelrahman, Stefanie Meitner, Amr Saadi, and Florian Alt. 2021. ARtention: A design space for gaze-adaptive user interfaces in augmented reality. *Computers & Graphics* 95 (2021), 1–12. https://doi.org/10.1016/j.cag.2021.01.001

[36] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-Touch: Combining Gaze with Multi-Touch for Interaction on the Same Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 509–518. https://doi.org/10.1145/2642918.2647397

[37] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2016. Partially-Indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2845–2856. https://doi.org/10.1145/2858036.2858201

[38] 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 (Tokyo, Japan) *(UIST '16)*. Association for Computing Machinery, New York, NY, USA, 301–311. https://doi.org/10.1145/2984511.2984514

[39] Ken Pfeuffer, Benedikt Mayer, Diako Mardanbegi, and Hans Gellersen. 2017. Gaze + Pinch Interaction in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction* (Brighton, United Kingdom) *(SUI '17)*. Association for Computing Machinery, New York, NY, USA, 99–108. https://doi.org/10.1145/3131277.3132180

[40] Ken Pfeuffer, Lukas Mecke, Sarah Delgado Rodriguez, Mariam Hassib, Hannah Maier, and Florian Alt. 2020. Empirical Evaluation of Gaze-Enhanced Menus in Virtual Reality. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology* (Virtual Event, Canada) *(VRST '20)*. Association for Computing Machinery, New York, NY, USA, Article 20, 11 pages. https://doi.org/10.1145/3385956.3418962

[41] Robin Piening, Ken Pfeuffer, Augusto Esteves, Tim Mittermeier, Sarah Prange, Philippe Schröder, and Florian Alt. 2021. Looking for info: Evaluation of gaze based information retrieval in augmented reality. In *Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part I 18*. Springer, 544–565. https://doi.org/10.1007/978-3-030-85623-6_32

[42] Jef Raskin. 2000. *The humane interface: new directions for designing interactive systems*. Addison-Wesley Professional.

[43] Katharina Reiter, Ken Pfeuffer, Augusto Esteves, Tim Mittermeier, and Florian Alt. 2022. Look & Turn: One-Handed and Expressive Menu Interaction by Gaze and Arm Turns in VR. In *2022 Symposium on Eye Tracking Research and Applications* (Seattle, WA, USA) *(ETRA '22)*. Association for Computing Machinery, New York, NY, USA, Article 66, 7 pages. https://doi.org/10.1145/3517031.3529233

[44] Martin Schmitz, Sebastian Günther, Dominik Schön, and Florian Müller. 2022. Squeezy-Feely: Investigating Lateral Thumb-Index Pinching as an Input Modality. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 61, 15 pages. https://doi.org/10.1145/3491102.3501981

[45] Abigail J. Sellen, Gordon P. Kurtenbach, and William A.S. Buxton. 1992. The Prevention of Mode Errors Through Sensory Feedback. *Human–Computer Interaction* 7, 2 (1992), 141–164. https://doi.org/10.1207/s15327051hci0702_1

[46] Ludwig Sidenmark, Dominic Potts, Bill Bapisch, and Hans Gellersen. 2021. Radi-Eye: Hands-Free Radial Interfaces for 3D Interaction Using Gaze-Activated Head-Crossing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 740, 11 pages. https://doi.org/10.1145/3411764.3445697

[47] Sophie Stellmach and Raimund Dachselt. 2012. Investigating Gaze-Supported Multimodal Pan and Zoom. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 357–360. https://doi.org/10.1145/2168556.2168636

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

[49] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2019. Experimental Analysis of Barehand Mid-Air Mode-Switching Techniques in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300426

[50] Patricia Tegtmeier. 2018. A scoping review on smart mobile devices and physical strain. *Work* 59 (2018), 273–283. https://doi.org/10.3233/WOR-172678 2.

[51] Uta Wagner, Mathias N. Lystbæk, Pavel Manakhov, Jens Emil Grønbæk, Ken Pfeuffer, and Hans Gellersen. 2023. A Fitts' Law Study of Gaze-Hand Alignment for Selection in 3D User Interfaces. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, 15 pages. https://doi.org/10.1145/3544548.3581423

[52] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 113 (Sept. 2017), 21 pages. https://doi.org/10.1145/3130978

[53] Xin Yi, Yiqin Lu, Ziyin Cai, Zihan Wu, Yuntao Wang, and Yuanchun Shi. 2022. GazeDock: Gaze-Only Menu Selection in Virtual Reality using Auto-Triggering Peripheral Menu. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 832–842. https://doi.org/10.1109/VR51125.2022.00105

[54] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 246–253. https://doi.org/10.1145/302979.303053

[55] Yang Zhang, Wolf Kienzle, Yanjun Ma, Shiu S. Ng, Hrvoje Benko, and Chris Harrison. 2019. ActiTouch: Robust Touch Detection for On-Skin AR/VR Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1151–1159. https://doi.org/10.1145/3332165.3347869