Radi-Eye: Hands-Free Radial Interfaces for 3D Interaction using Gaze-Activated Head-Crossing

Ludwig Sidenmark
l.sidenmark@lancaster.ac.uk
Lancaster University
United Kingdom

Dominic Potts
d.potts2@lancaster.ac.uk
Lancaster University
United Kingdom

Bill Bapisch
bill.bapisch@campus.lmu.de
LMU Munich
Germany

Hans Gellersen
hwg@cs.au.dk
Aarhus University
Denmark

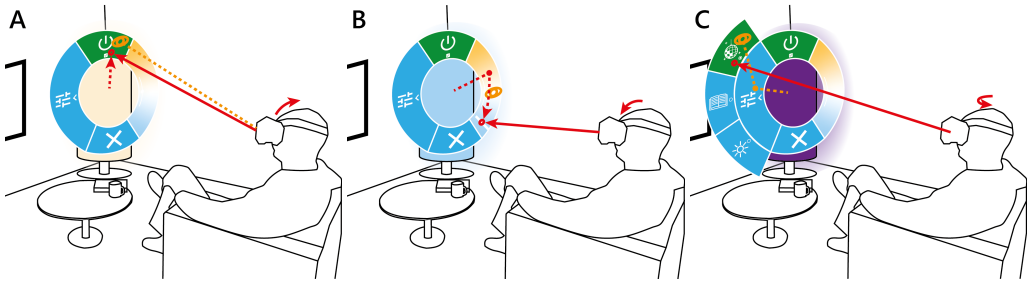

Figure 1: Radi-Eye in a smart home environment for control of appliances. A: The user turns on the lamp via a toggle selection with minimal effort using only gaze (orange) and head (red) movements. B: Selection can be expanded to subsequent head-controlled continuous interaction to adjust the light colour via a slider. C: Gaze-triggered nested levels support a large number of widgets and easy selection of one of the multiple preset lighting modes. The widgets enabled via Radi-Eye allow a high-level of hands-free and at-a-distance control of objects from any position.

ABSTRACT

Eye gaze and head movement are attractive for hands-free 3D interaction in head-mounted displays, but existing interfaces afford only limited control. *Radi-Eye* is a novel pop-up radial interface designed to maximise expressiveness with input from only the eyes and head. Radi-Eye provides widgets for discrete and continuous input and scales to support larger feature sets. Widgets can be selected with *Look & Cross*, using gaze for pre-selection followed by head-crossing as trigger and for manipulation. The technique leverages natural eye-head coordination where eye and head move at an offset unless explicitly brought into alignment, enabling interaction without risk of unintended input. We explore Radi-Eye in three augmented and virtual reality applications, and evaluate the effect of radial interface scale and orientation on performance with Look & Cross. The results show that Radi-Eye provides users with fast and accurate input while opening up a new design space for hands-free fluid interaction.

CCS CONCEPTS

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

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445697

KEYWORDS

Eye tracking; Gaze interaction; Eye-Head Coordination; Radial Interface; Virtual Reality; Augmented Reality

ACM Reference Format:
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 *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3411764.3445697

1 INTRODUCTION

Hands-free interaction is widely studied for 3D interaction in head-mounted displays (HMDs) to support contexts in which the hands are unavailable, occupied, or where space, ability or fatigue constrain their use [27, 29, 37, 46, 48]. Gaze and head are attractive alternatives to manual input, as their movement can be tracked with built-in sensors without needing additional devices. Gaze is fast and effortless in directing attention to objects [42] while head movement is more precise for control [7]. However, both modalities lack an intrinsic mechanism for selection and expose Midas Touch issues as they are "always on", in past work addressed with separate confirmation techniques, such as dwell time [17]. As a result, hands-free interfaces have remained limited in the expressiveness and control they afford.

In prior work, we have shown that input from the eyes and head can be combined for fast and robust pointing and selection, based on the natural coordination of eye and head in directing gaze [37]. In this work, we introduce a holistic interface design for user control and expressive interaction with only eye and head movements. *Radi-Eye* is designed as a pop-up radial interface that provides

widgets for discrete and continuous input, contextual interaction, nested control and toggling of content. When invoked, Radi-Eye pops up in a head-centered position from where radially arranged widgets can be accessed comfortably and efficiently with gaze and head movement. While presented as a radial menu, the interface is scalable to large sets of features, through additional rings, toggling of components displayed on the rings, and adaptation to objects over which Radi-Eye is opened. Fig.1 illustrates some of the affordances.

Look & Cross is a gaze-activated head-crossing technique that complements the design for interaction across the radial interface. The technique employs gaze for hover interaction and pre-selection of widgets, and head-crossing to complete selection of a gaze-activated widget. This is natural and efficient for widget selection as head movement naturally follows eye gaze. It avoids Midas Touch as head orientation normally remains offset from gaze direction unless users explicitly choose to fully align head and gaze [36, 37]. As a result, Look & Cross enables fluid selection of multiple objects across the interface, without risk of unintended activation of objects crossed by either head or gaze alone. The technique design reflects the relative strengths of gaze for visual exploration and fast pointing and head movement for more deliberate and precise control, which can also seamlessly extend from crossing-based selection to manipulation of continuous inputs.

Radi-Eye is designed to maximise user control and expressiveness with only eye and head movement for interaction. The radial interface structure combined with Look & Cross for fluid eye and head control enables a novel style of hands-free HMD interface that we explore through implementation of three applications in virtual and augmented reality. The applications provide insight into the design space of Radi-Eye and design considerations in the interplay of interface layout, eye-head interaction, and visual feedback strategies. In a user study, we then evaluate effect of radial interface parameters on Look & Cross performance, to gain insight into design choices for ring and button sizes.

The contributions of this work comprise: (1) The Radi-Eye concept for hands-free interfaces, for which we discuss the design space, contribute concrete widgets, and present application examples; (2) The Look & Cross technique combining eye and head movement for a fluid style of interaction that supports hover, selection and continuous input; (3) Design insights from applications and study of Radi-Eye, on qualitative and quantitative aspects including impact of radial menu size, widget quantity, and widget positioning on user performance and preference.

2 RELATED WORK

Gaze and head have been studied since the eighties for hands-free interaction, albeit mostly as separate modalities[5, 18]. For the design of Radi-Eye we are building on insight on eye-head coordination and the combined use of gaze and head motion, as well as prior work on radial interfaces and crossing.

2.1 Eye-Head Coordination and Interaction

Gaze is mostly associated with the eyes, but naturally supported by head movement. The eyes have a physical range of about 50 degrees but rarely rotate beyond 30 degrees [40]. Eye movement alone may achieve small gaze shifts, whereas head movement enables larger gaze shifts and maintenance of a comfortable eye-in-head position. During a gaze shift, the eyes are faster and will precede the head. The head moves in the same direction to support gaze to reach further and to maintain a comfortable eye-in-head position [21, 39]. However, the head does not normally move all the way to align with the gaze direction, as head movement requires more energy while a comfortable eye-in-head position is reached sooner [12, 36]. As a result, the eyes and head are generally not aligned as we explore our surroundings [37]. The design of Look & Crossleverages both the natural sequence of gaze preceding the head and the natural offset between the eyes and head for interface control.

A range of works have compared gaze and head movement for interaction. In comparison, eye gaze is faster and more effortless, while the head is more stable and affords better control [5, 7, 20, 31]. Researchers have also developed interaction techniques that use the eyes and head for subsequent usage for refined selection [19, 20, 35, 39], head-turning for target disambiguation of gazed on targets [25], or head movement to move a tool glass over gazed on targets [24]. In recent work, we proposed *Eye&Head* selection techniques that build on the coordinated relationship between the eyes and head [37]. *Eye&Head Convergence* introduced target selection by aligning of gaze and head direction within a small angular threshold, exploiting that gaze and head are normally at an offset but straightforward to align at will. Look & Cross builds on the same principle, but enforces a sequence of interaction. Gaze has to precede the head on the target for a selection to occur, ensuring a controlled three-step interaction process from idle, to hover, to selection.

2.2 Radial User Interfaces

Radial user interfaces place items along the circumference of a circle or ring, and were initially introduced as an alternative desktop interface in the 1980s [8]. Researchers have since then established multiple advantages over traditional linear menus. Radial interfaces afford equivalent distance to all items while also exploiting users' spatial memory by placing items in separate directions, allowing fast selection while reducing the need for precise pointing [4, 8]. Interaction starts from the centre which makes radial interfaces attractive for modalities which have a natural "central state" from which they can move in any direction [9, 16, 27, 46, 47]. However, radial interfaces are limited to a number of items before there is an increase in erroneous selections due to decreased item size [44]. Also, as radial interfaces assume that the cursor is placed in the centre, they cannot be invoked at the edge of a screen without disrupting the interface structure [15]. In Radi-Eye, the pop-up interface combined with HMD-based interaction ensures that interaction can always start from the centre.

In 3D environments, research has shown that radial interfaces are better performing than their linear counterparts [10, 30, 33]. Gebhardt et al. also extended a hand-controlled radial interface to include more advanced widgets such as check-boxes, radio buttons, and sliders [13]. While a hand-controller using ray-casting combined with a button for confirmation is the dominant modality for radial interface input in 3D environments [11, 13, 14, 22, 30, 33], researchers have also explored a wide range of hands-free modalities. Previous work has shown that head [30, 47], feet [27], or body movements [46] can be effectively used for radial interface control.

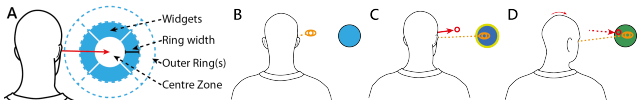

Figure 2: Radi-Eye layout (A) and Look & Cross object selection (B-D). A: Radi-Eye layout and components. B: The object is idle. C: Gaze (orange) moves to the object, triggering hover interaction, enabling the object for selection, and displays a head cursor (red). D: The head moves across the gazed-on object's boundary, selecting the object.

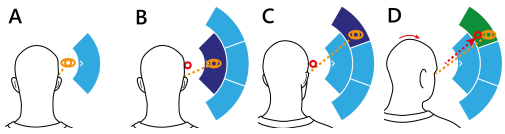

Figure 3: Radi-Eye nested interaction. A: The parent widget is idle. B: Gaze on the parent widget displays the nested widgets on an outer ring. C: Gaze moves to a nested widget, enabling selection. D: The head moves in a direct path to select the nested widget.

We extend the work on hands-free interface input beyond basic selection by combining gaze and head pointing for a variety of widgets that support discrete and continuous user input, interface scaling, and command composition for expressive user control.

2.3 Crossing

Crossing is a selection technique that selects a target by crossing its boundary with a cursor [1]. The technique allows fast and accurate selection of targets [1], and have shown to be as expressive as the common pointing metaphor [2]. Crossing relaxes the constraint on fine-grained pointing within a closed area as the user only has to cross the target boundary. As such, researchers have proposed crossing for users with limited motor capabilities [45], and for a wide array of modalities with limited fine-grained pointing or where the modality lacks an explicit confirmation mechanism [6, 23, 44, 48]. More recently, researchers have shown that crossing can be effectively used in 3D environments by ray-casting with a controller [43], or for hands-free selection via the head [47, 48]. Finally, crossing also allows selection of multiple targets in a single user input [2, 41], allowing fast and expressive user interfaces.

Similar to the Midas Touch problem, a common issue with crossing are distractor targets that are inbetween the cursor and the intended target, forcing the user to interrupt an input [3]. Modalities such as a stylus or mouse solve this issue by lifting the stylus or releasing a mouse button [3, 45]. However, the issue is more problematic for the head and gaze that are "always on" and have no inherent confirmation mechanism. Placing items in different directions makes radial interfaces an ideal interface for crossing interaction as it reduces the risk of distractor targets [1]. Alternatively, researchers have proposed techniques that add additional steps to the interaction, by for example, forcing the user to exit the target in the same direction as they entered [48], or by moving the cursor to a secondary target after crossing [26, 34]. Finally, work has proposed to use gaze to "enable" targets as selectable by a second modality [45]. We build on this notion by using eye gaze to activate targets for crossing selection with the head, which allows users to freely move their head and gaze over distractor targets without triggering accidental selections.

3 RADI-EYE

The core concept of *Radi-Eye* is to use an eye- and head-controlled interface for expressive hands-free control of objects in any 3D environment. To achieve this, Radi-Eye consists of three parts:

(1) A pop-up radial interface for on-demand interaction;
(2) Look & Cross, a gaze-enabled head-crossing selection technique;
(3) Widgets that enable discrete and continuous interaction, and interface scaling.

3.1 Pop-up Radial Interface

Radi-Eye has a radial structure composed of different widgets that are placed along the circumference of one or multiple rings (fig. 2a). The content of the interface (i.e. the widgets displayed) can be fixed or changeable to make additional functionality available. Also, ring widths can be alternated to account for eye tracking accuracy, and information to be displayed on the widgets. The radial structure of Radi-Eye is based on the eyes and head's capability to move in all directions when performing gaze shifts. Radi-Eye interaction is started from the inactive *centre zone* of the radial interface from which the user can move their eyes and head in any direction for interaction. This allows Radi-Eye to exploit proven advantages of the radial layout [8, 15].

The interface supports scaling in three ways to support large interfaces and increased functionality. First, nested interaction is supported by expanding the interface on outer rings (fig. 3). Hidden nested widgets are displayed when gazing on the parent widget, allowing users to search and traverse through a nested interface without committing to a selection. Users can then move their head in a direct path to a nested gazed-on widget for selection. Nested widgets are placed in a fan-like structure to avoid cumbersome "criss-cross" gaze behaviour caused by widgets being on opposite ends of the interface [32]. Second, the interface supports replacing widgets on a ring by toggling a widget. Replacing widgets allows a single ring to store a large number of widgets without relying on multiple layers. Third, Radi-Eye supports the fluid composition of multiple commands (fig. 4), achieved by placing commands and their options on separate rings. The user can then select a command at each ring when traversing through the rings. The user may even pass over a ring without performing a selection if a command is undesirable. Command composition allows efficient input of multiple parameters, and the execution of advanced commands that require heterogeneous input.

The pop-up nature of Radi-Eye allows on-demand control of objects in the environment without causing clutter when not in use. Accounting for the user position and head direction during invocation ensures that the interface can always be positioned in or close to the middle of the screen at the start of an interaction to ensure possible interaction in all directions [15]. Also, HMD-based

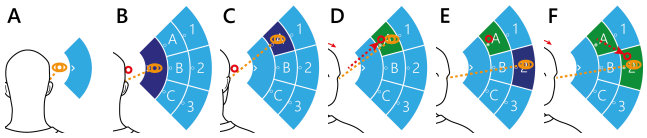

Figure 4: Radi-Eye command composition on multiple rings. A: The parent widget is idle, hiding the available commands. B: Gaze triggers the parent widget showing the nested commands displayed on separate rings. C-D: The user enables (C) and selects (D) the first command. E-F: The user moves on to enable (E) and select (F) the second command.

interaction allows the of space outside the field of view that is revealed during head movement for interface control.

The structure, position, and invocation of a Radi-Eye interface depends on the context of use. In its simplest form, the interface is used to manipulate a specific object in the environment directly. The characteristics of these *Object-dependent* Radi-Eye interfaces are dependent on the interacted object. The invocation method should single out a specific object in the environment (e.g. ray casting), the displayed widgets depend on the object's functionality, and the position and structure of the interface should be linked to the interacted object to allow an efficient feedback-control loop.

In certain situations, a user may want to interact with multiple objects simultaneously, or with objects that are not visible or at an uncomfortable head position. In such cases, an *Object-independent* Radi-Eye interface which is not linked to an object in the environment can be used for interaction. Object-independent invocation is performed independently of any specific object via a generic gesture or command. The interface position is not linked to a specific object but primarily considers user convenience, i.e. centred around the head. The available commands are not dependent on specific objects, and in cases of multiple interacted objects new commands can arise from their common elements or any additional functionally arising from combining them.

As gaze is used for interaction, feedback that triggers visual attention should not disturb the flow of interaction. Feedback can thus be displayed on the centre zone, widgets, or outer rings depending on the context of use to guide the user towards the next steps of the interaction. Also, feedback should not be displayed too far away from the head position (>30°) during interaction to ensure comfortable eye-in-head positions.

3.2 Look & Cross

Look & Cross is designed as a generic interaction technique for Radi-Eye, combining gaze for pre-selection of interface components with head-crossing for their invocation. The technique is based on the natural misalignment between the eyes and head, and that gaze naturally precedes the head during a gaze shift [12, 36, 37]. It uses fast and effortless gaze to explore the interface, trigger hover interaction, and enable widgets for selection. The user then selects a widget by moving the stable head across the gaze-activated widget's boundary. Look & Cross thus supports a controlled three-step interaction process: from idle, to hover, and finally to selection (fig. 2b-d). As such, users can dwell with their gaze on widgets without causing unintended input, which may be useful for cognitively demanding tasks where thorough consideration of choices can induce prolonged fixation.

Building on the natural offset of the eyes and head ensures robust selection as both modalities have to point on the same widget at the same time. The natural offset between the eyes and head allows users to "skip over" idle distractor targets with their head without performing selections. This is useful when performing a series of selections, such as selecting multiple widgets in a list. Also, users can shift their gaze and head outside the interface, allowing free exploration of the interface and surrounding environment without triggering an unwanted interaction.

Look & Cross is inspired by Eye&Head Convergence, a selection technique that uses eye and head alignment for selection [37]. However, the techniques differ in selection condition, and handling of the eyes and head accidentally aligning over a target. Firstly, in Eye&Head Convergence, users have to move the head within an angular distance to the eyes, requiring a gaze cursor to display the selectable area, cluttering the selection space. Look & Cross defines the selection area as the border of the gazed-on widget and therefore require no gaze cursor. Using the widget boundary is also beneficial for defining an area for subsequent continuous head interaction that is not dependent on the gaze position, or when multiple small targets are nearby (e.g. nested interface), as an angular area may overlap multiple targets causing selection ambiguity. Secondly, Eye&Head Convergence starts a timer during which the eyes and head have to remain within the angular threshold to confirm the selection if the head cursor is already within the convergence area when gaze first points at the target. In Look & Cross, we enforce the order of the interaction sequence; gaze has to precede the head on a target. Forcing the movement order ensures hover interaction and the same interaction sequence for all selections. Also, the user does not have to worry about accidental selections caused by dwelling on head-pointed targets for too long.

Table 1: Widgets available with Radi-Eye.

Type	Discrete					Continuous		Scaling	
Widget	Basic selection	Toggle	Checkboxes	Radio button	Step increment	Slider	Hold down	Nested ring	Replace ring
Icon	None	☐☑	☐☑	○◉	− +	⊸	🕒	›	≡×

3.3 Radi-Eye Widgets

Combining the radial pop-up interface and Look & Cross provides the final Radi-Eye component, a broad set of widgets available for object control (table 1).

3.3.1 Discrete selection. The Look & Cross interaction sequence supports a multitude of widgets using discrete selection, from basic selections to toggle widgets (see table 1). Logically connected widgets such as checkboxes and radio buttons can be placed adjacent to each other along the ring to indicate their association with each other. Icons are placed on the widgets to indicate the widget type and required interaction before performing a head movement.

3.3.2 Continuous Interaction. Extending the interaction sequence of Look & Cross provides continuous interaction. To trigger a continuous interaction, the head cursor has to remain within the widget

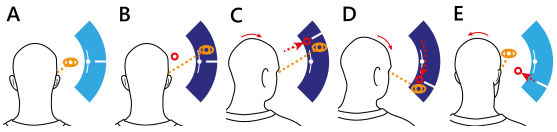

Figure 5: Radi-Eye slider interaction. A: The slider is idle. B: Gaze enables the slider for interaction. C: The head moves to trigger interaction. The slider handle jumps to the cursor position. D: The head moves along the slider's arc, changing the slider value. E: The head exits the slider, the slider handle stays at the last cursor position.

boundary after the initial selection. The continuous interaction is then active until the head cursor leaves the widget boundary, similar to holding down a button. We can also transform the widget into a 1-dimensional slider along the ring arc by adding meaning to the cursor position (fig. 5). During continuous interaction, the user's gaze can move freely outside the widget to ensure no strain caused by gaze being "trapped" within the widget.

4 APPLICATIONS

We developed three applications in both VR and AR environments to showcase the different faucets and highlight design considerations of Radi-Eye: a VR media player, an AR smart home manager, and a VR city builder. We implemented all applications in Unity. We used the HTC Vive with an integrated Tobii eye tracker for the VR applications. For the AR application, we used a Zed mini see-through camera combined with an HTC Vive Pro Eye. Both HTC Vives have a vertical FOV of 110° and a horizontal FOV of 100°. However, the Zed mini only has a vertical FOV of 54° and a horizontal FOV of 85°. Note, the gaze position is not visible to the user in our applications. However, all figures show the gaze position for illustrative purposes.

4.1 Media Player

Our first application is a VR media player designed to demonstrate some of the fundamental types of interaction. The user invokes the media player by dwelling their gaze on a button at the bottom of the television (TV). Once invoked, the interface is positioned centred around the TV and sized to encompass the TV inside the centre zone and support a large number of widgets (40°). Ring width was set to 5-7.5° to make the effect of eye tracking error negligible. Users can play/pause videos using a toggle interaction, toggle subtitles (fig. 6), and seek through the video via the timeline slider for coarse timestamp selection or by reversing/forwarding the video using the "hold-down" reverse and forward buttons for further refinement (fig. 7). To address space limitations, we use the *replace ring* widget to switch from playback control to video browsing via toggling. Users can then select videos to view alphabetically, displaying corresponding videos in an outer layer (fig. 6c-f).

The media player exemplifies an *Object-dependent* interface as it is directly related to the controlled TV. We placed any continuous input in the innermost-ring, the shortest distance to the central feedback (20°), to ensure a comfortable eye-in-head position while users look back at the TV to, for example, inspect the timestamp (fig. 7). Conversely, the nested outer-ring is utilised for its increased

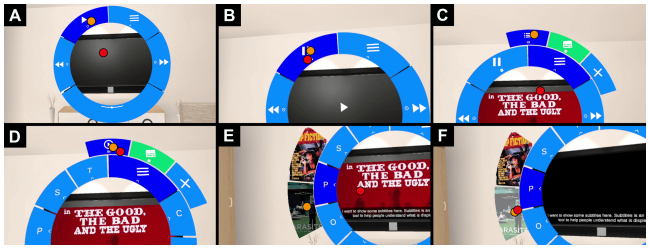

Figure 6: The user plays and changes a video. Gaze (orange) moves to the play button to enable selection (A) while the head (red) crosses the gazed-on button's boundary to start the video (B). The user gaze on the nested select media button (C) and selects it to replace the inner ring's playback controls with media selection controls (E). The user searches through available media (E) and selects a new video (F).

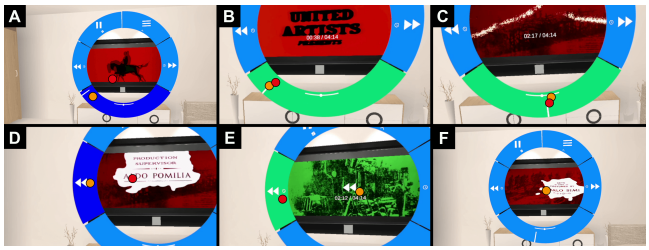

Figure 7: The user changes the video timestamp. Gaze enables the timeline-slider (A) and the head crosses the slider selecting the timestamp at the head position (B). The head moves along the timeline to fast-forward the video (C). The reverse button is then enabled the with gaze (D) and activated with the head (E) for further timestamp refinement. The head exits the reverse button to resume the video (F).

item real estate when selecting different media to display (fig. 6e). The application showcases a number of advantages of Radi-Eye:

- Users can inspect widgets for as long as users need to without risking unintended selection.
- Users can traverse and inspect nested items with gaze without committing a head movement until selection.
- Widgets can complement each other to support different levels of granularity (slider and reverse/forward buttons).
- The interface can be extended to support additional functionality (media selection) via ring replacement and a large number of options via nested interaction.
- Users are free to visually explore the interface or feedback while a continuous interaction is maintained. This is useful when observing feedback that is external to the widget, for example, seeking through a video.

4.2 Smart-Home

Our second application is an AR smart-home manager where users control lighting and home appliances via Radi-Eye. The application supports both contextual, and non-contextual interaction. Radi-Eye interfaces for individual appliance control are *Object-dependent*.

Figure 8: A: The user invokes an Radi-Eye interface for appliance control by gaze dwelling on an appliance. B-C: The object-dependent interface adapts to the selected appliance (B: lamp, C: fan) and is centred around the user's head.

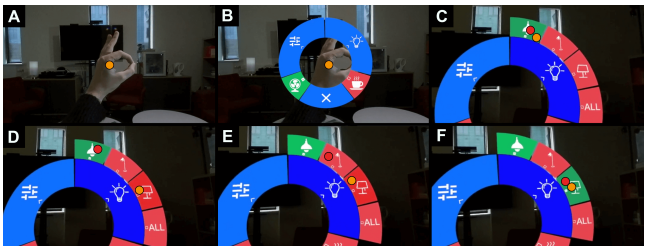

Figure 9: The user invokes and turns on multiple lamps via the master-menu. The user performs a hand gesture (A) to invoke the interface (B). The user then selects and turns on the first lamp (C). Gaze then moves to enable the second button (D). The head moves over a non-enabled button which does not trigger (E) to turn on the second lamp (F).

Similar to the media player, the user performs invocation via gaze dwell on the physical appliance. The interface controls depend on the selected appliance: selecting a kettle shows simple toggle controls for power (20° centre zone size, fig. 8c), while a lamp has more advanced controls for adjusting brightness and colour via sliders thus requiring a slightly larger interface for increased continuous precision (30° centre zone size, fig. 8b). Individual appliance control requires line of sight for invocation which can be problematic depending on a user's position in the room.

We developed an *Object-independent* "master-menu", that can be invoked from any position and enables users to control appliances en masse from one interface. The master-menu has a slightly larger centre zone (40°) to accommodate easy selection of all appliances in the outer ring. In this application, the user invokes the master-menu by performing a simple hand gesture in front of the HMD AR-camera (fig. 9a-b). Voice commands or head gestures can be used as hands-free alternatives. From the master-menu, users can toggle all available appliances, toggle lamps individually and en masse, and toggle lighting presets such as reading, day-light or disco (fig. 9b). The application highlights multiple Radi-Eye advantages:

- Users can invoke an *Object-dependent* interface or an *Object-independent* interface depending on their needs.
- Interfaces can be adaptive to context and interacted objects.
- Through the master-menu users can control out-of-view objects without significant body movements.
- Users can traverse their gaze or head across other options when reaching for a target, affording freedom in the choice of interface layout (fig. 9c-f).
- Radi-Eye can be utilised in both VR and AR and can be adapted to cluttered domestic or workplace settings.

4.3 City Builder

The final application is a VR city builder where users can inspect, build, and edit a city using Radi-Eye. To inspect and edit a building, the user invokes the "Change-menu" by gaze dwelling on a plot with an existing building (fig. 10a-b). The selected building can then be demolished or adjusted by choosing a new rotation or colour.

The user places a new building by dwelling on an empty plot. This invokes a Radi-Eye interface where users can choose between commercial, industrial, residential, or public buildings that are available for the selected plot. The user can also select the rotation and colour of the building via command composition before confirming the building placement (fig. 10c-h). As command composition requires multiple layers depending on the number of performed actions, placing feedback in the centre zone (in this case the placed building) would cause it to be further away from the user the more commands they performed. This feedback placement can lead to issues when users want to observe feedback, having to gaze back to the centre zone disrupting the command composition. Instead, we display feedback on the outermost confirm button, the direction to which the head-gesture and gaze point are moving toward (fig. 10c-h). The centre zone is set to smaller (20°) to accommodate multiple layers, and ring widths are narrower (5°) to allow comfortable selection of the outermost layer (35° from the centre position). The application highlights a number of advantages of Radi-Eye:

- Radi-Eye supports complex object manipulation and rapid multiparameter input via command composition.
- Radi-Eye allows users to skip over commands during command composition to more rapidly perform an action.
- Radi-Eye supports users to change and omit selections during command composition before performing an action via the confirmation button.
- Radi-Eye supports a variety of feedback strategies that can be adopted on or off the interface, depending on the interaction.

5 USER STUDY

In previous work we have compared gaze-head alignment against dwell and found the technique faster for selection of known targets and perceived as more natural and easier to use [37]. This established that users are effective and efficient with the technique. For Radi-Eye we therefore focussed not on comparison against other selection mechanisms, but effect of radial interface parameters on Look & Cross performance. Some of the fundamental design choices of relate to *ring size*, *widget amount*, and *widget direction*. We conducted a user study to investigate how these factors impact user performance and reception in a task where participants had to select a particular widget out of many. For the sake of simplicity, we focused our study on basic selection.

In this study, we refer to ring size as the centre zone size. We decided on a set ring width of 10° visual angle. Ring size impacts the required distance for selection, widget size, and also the occluded areas of the environment during interaction. We investigated ring sizes that cover a range from only requiring small head movements for selection to covering the whole HMD (S: 10, M: 35, L: 60, XL:

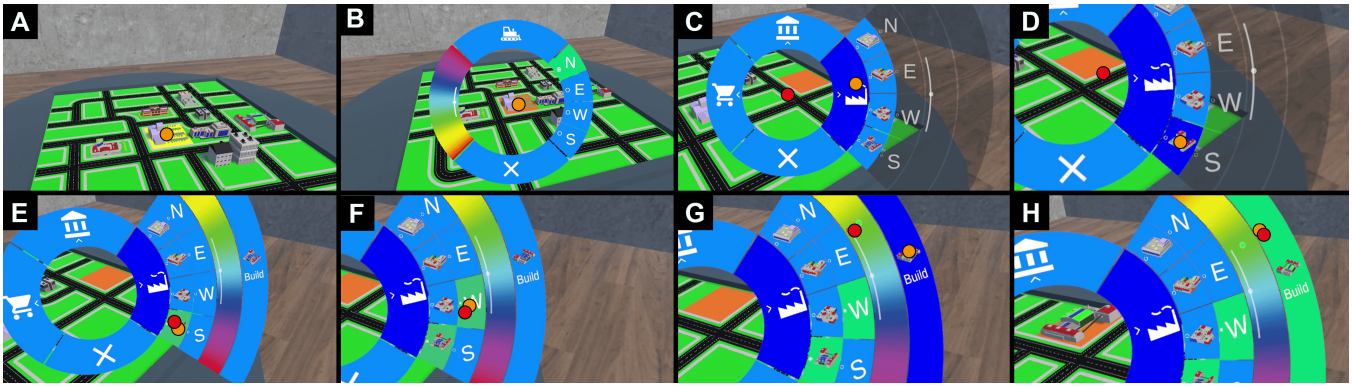

Figure 10: The user adjusts a building (A-B) and places a new building (C-H). Dwelling on a building invokes the change interface (A) which can adjust its rotation or colour, or demolish it (B). The user can invoke the build interface by dwelling on an empty plot. Gazing on a building type shows nested available buildings (C). The user then enables (D) and selects a building (E), enabling further commands. The selected building and its properties are shown on the outermost confirm button. The user selects a rotation (F) and colour (G) which updates the confirm button, and then selects the confirm button to place the building with the selected properties (H). Skipping steps F and G places a building without specifying rotation and colour.

85° visual angle). The widget amount of a ring has an impact on widget size, interface complexity, and search time. We set the range of widget amount to be 4, 8, and 16, limited by widget size. Widgets were equally sized. Finally, we vary the selection direction so that selections were performed in both cardinal and diagonal directions. As such, we used two layouts for the 4-widget condition to support both direction types.

Participants were tasked to select the correct widget out of many as fast and accurate as possible at varying ring sizes, widget amounts and widget directions (fig. 11). To start a trial, participants aligned their gaze and head towards a central target. The trial started after a 300ms of alignment when a radial interface and a single letter in the centre zone appeared at 8 metres distance. Participants were tasked to find and select the widget containing the letter that matches with the centre letter. Gaze feedback was shown by widget colour. A head cursor appeared when participants gazed on a widget. Participants moved the head cursor across the gazed-on widget to perform a selection. A correct or incorrect selection was shown via colour feedback. A trial was finished irrespective of whether the selection was correct or incorrect. Participants would then realign their gaze and head to start the next trial.

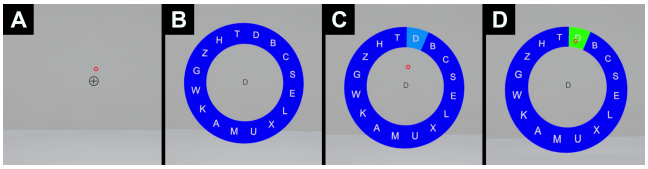

Figure 11: Example study trial. A: The participants start a trial by moving the head cursor and gaze into the central target. B-C: The participant has to find and select the letter "D". D: Colour feedback indicates selection success.

5.1 Apparatus

We used an HTC Vive with an integrated Tobii eye tracker for the user study. The HMD has a FOV of 100° and a 90Hz framerate. Eye tracking data was recorded at 120Hz. The study environment was developed in Unity version 2017.4.3f1.

5.2 Procedure

We recruited 12 participants for the user study (Six male, six female 27.64 ±6.23). For previous VR experience, two participants reported no experience, nine reported occasional, and one reported daily. For previous eye tracker experience, three participants reported no experience, seven reported occasional, one reported weekly, and one reported daily. Participants first signed a consent form and answered a demographic questionnaire. Participants were then seated and put on the HMD. Participants performed an eye tracking calibration and a training session before each test session. Ring size order was counterbalanced with a Latin square. Widget amount and widget direction were randomised. After completing the task with a ring size, participants removed the HMD and answered a NASA TLX Workload questionnaire. A semi-structured interview was conducted at the end to extract preferences and opinions. In total each participant performed 4 ring sizes x 3 widget amounts x 16 repetitions = 192 selections. Note that for the 4-widget condition, half of the selections used the cardinal layout and the other half the diagonal layout. The study took 30-40 minutes to complete.

5.3 Results

Our five dependent variables were search, selection and total time, error rate, and workload. Unless stated otherwise, the analysis was performed via a Ring Size × Widget Amount two-way repeated-measures ANOVA (4 × 3), Greenhouse-Geiser-corrected in cases where Mauchly's test indicated a violation of sphericity, and with Bonferroni-corrected post hoc tests where applicable. Effect sizes are reported as partial eta squared (η_p^2). Time Shapiro-Wilks tests

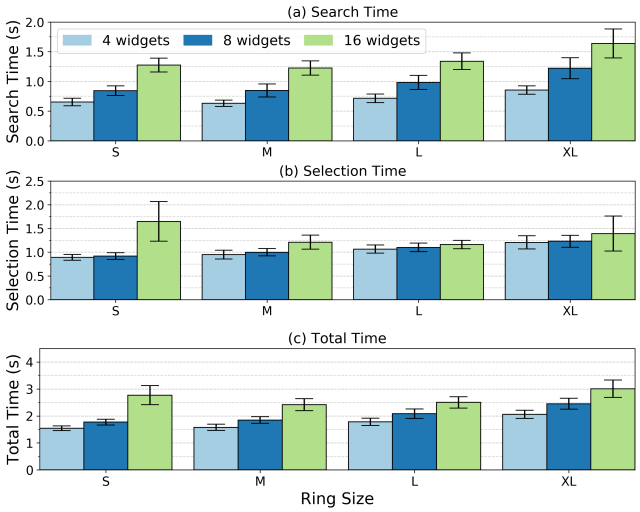

Figure 12: Mean search, selection and total time. Error bars represents mean 95% confidence interval.

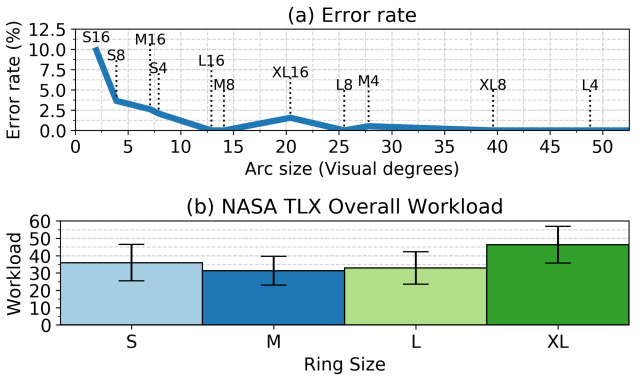

Figure 13: A: Error rate as a function of arc size. Labels indicate study conditions. B: Average overall workload. Error bars represents mean 95% confidence interval.

and Q-Q plots validated the assumption of normality. Only successful trials were used for analysis.

5.3.1 Search time. We defined search time as the time from the interface appears to the participant gaze on the matching widget. No interaction was found between ring size and widget amount ($F_{6,66}$=1.33, p=.257, η_p^2=.11). However, results showed a significant ($F_{2,22}$=279.63, p<.001, η_p^2=.96) increase in search time with increasing widget amount (fig. 12a). Post hoc analysis showed significant differences between all conditions (all $p < .001$). We also found a significant main effect on ring size ($F_{3,33}$=17.82, p<.001, η_p^2=.62). Further tests showed that XL had significantly higher search time than other ring sizes (all $p \leq .008$), indicating that ring size increases search time if users have to rely on head-movement for search. As such, both ring size and widget amount should be considered to reduce search time. Widget layout showed no significance for the 4-widget condition ($F_{1,11}$=1.22, p=.292, η_p^2=.10).

5.3.2 Selection Time. We defined selection time as the time from that the participant gaze on the matching widget until a selection was made. Participants were able to select widgets at 2 seconds or less for all conditions (fig. 12b). We found a significant two-way interaction between ring size and widget amount ($F_{1.70,18.72}$=15.01, p<.001, η_p^2=.58). Further analysis showed that Ring size had a simple main effect at 4 ($F_{3,33}$=15.68, p<.001, η_p^2=.59), 8 ($F_{1.70,18.64}$=17.22, p<.001, η_p^2=.61), and 16 widget amounts ($F_{1.67,18.33}$=5.21, p=.020, η_p^2=.32). Similarly, widget amount had a simple main effect at ring sizes S ($F_{1.09,11.95}$=30.79, p<.001, η_p^2=.74), M ($F_{2,22}$=22.07, p<.001, η_p^2=.67), and L ($F_{2,22}$=8.83, p=.002, η_p^2=.45). However, no significant main effect was found for ring size XL ($F_{2,22}$=1.60, p=.226, η_p^2=.13). A larger ring size led to a higher selection time as larger head-movements are required for selection. Also, as the combination of ring size and widget amount decides the widgets' arc size, a large widget amount combined with a small ring size can thus lead to slower selections as head movements have to be more precise. But the effect of widget amount becomes negligible if the ring size is large enough. Widget layout at the 4-widget condition also showed a significant effect on selection time ($F_{1,11}$=16.50, p=.002, , η_p^2=.60) where the cardinal layout was slightly but significantly faster than the diagonal layout (0.05s).

5.3.3 Total time. We defined total time as the time between the moment a ring is presented until the moment a selection was made which includes both search and selection time. We found a significant interaction between ring size and widget amount (fig. 12c, $F_{1.82,20.26}$=6.32, p=.008, η_p^2=.37). Further investigation into simple main effects showed that ring size had an effect at 4 ($F_{3,33}$=68.85, p<.001, η_p^2=.86), 8 ($F_{1.76,19.40}$=34.00, p<.001, η_p^2=.76), and 16 widget amounts ($F_{1.61,17.70}$=4.61, p=.031, η_p^2=.30). We also found simple main effects for widget amount on S ($F_{1.04,11.42}$=35.57, p<.001, η_p^2=.76), M ($F_{1.38,15.14}$=77.79, p<.001, η_p^2=.88), L ($F_{2,22}$=89.15, p<.001, η_p^2=.89), and XL ($F_{2,22}$=80.77, p<.001, η_p^2=.88). Similar to the selection time results, an increase in widget amount and ring size lead to an increase in total time. A larger ring size lead to larger distances for the head to travel, while a high widget amount combined with a small ring size can lead to slower selections due to an increase in required precision. We also found that widget layout had a small but significant influence on total time ($F_{1,11}$=12.34, p=.005, η_p^2=.53) where the cardinal layout was slightly but significantly faster than the diagonal layout (0.05s).

5.3.4 Error Rate. We define an error as the percentage of erroneous selections among all selections. Participants performed a low amount of erroneous selections. In total, only 39 out of 2304 (1.7%) selections were erroneous. Figure 13a highlights how the error rate is affected by a combination of ring size and widget amount, which decides the widgets' visual angle arc size. The results showed an increase in error rate for widgets with an arc size smaller than 2°.

5.3.5 Workload. Friedman test on the overall workload from the NASA TLX questionnaire (fig. 13b) showed that ring size had a significant effect (χ^2(3)=10.66, p=.014). Further Bonferroni corrected

Wilcoxon comparisons showed that the ring size XL had significantly more workload than M (p=.034) and L (p=.027). Friedman tests on the weighted averages of each sub-scale showed significant differences in Physical Demand ($\chi^2(3)$=12.69, p=.005). Bonferroni corrected Wilcoxon analysis showed that ring size S had significantly lower Physical Demand than ring size XL (p=.012).

5.3.6 Preference. Look & Cross was considered to be "easy to use" (P7) and "intuitive" (P8). Participants also mentioned that it was "natural to use" (P2) and the use of both modalities "helps preventing false selections" (P9). When asked about their preferred ring size, 8 participants stated the ring size M as their most preferred. Three participants stated that sizes M and L were equally preferred. Lastly, one participant preferred the smallest size S due to its "quickness" (P9). Ring sizes S and XL were disfavoured for various reasons. Participants had trouble with size S due to "being unable to control the eyes enough to stay on the buttons" (P12) or because it "restricted head movement" (P7). The ring size XL was disliked due to "too much head movement" (P5) and "long search time" (P4). Finally, all participants mentioned either horizontal or vertical directions as their most preferred selection direction.

6 DISCUSSION

Radi-Eye is effective for hands-free selection and control of objects and affords a wide variety of widgets for discrete and continuous input in any 3d environment. The pop-up nature of Radi-Eye allows convenient and on-demand control of the surrounding environment (both visible and occluded) from any user position. The ability to control objects outside the FOV and the reliance on only eye and head movements is significant and highly relevant for contexts with constrained user motion input (e.g. seated usage).

Radi-Eye is versatile and lends itself to implementation in different configurations of invocation method, widgets, and interface structure as showcased by our three applications. Radi-Eye can control single or multiple objects to the users' discretion as demonstrated in the smart home application. Widgets can complement each other for verbose and refined interactions as highlighted in the media player which uses a slider for initial selection and the hold-down widget for further refinement of the timestamp (fig. 7). We can also combine widgets into a fluid series of commands via command composition for efficient and expressive interaction, as shown when placing a building (fig. 10). Both the applications and user study points toward design aspects to consider when designing a Radi-Eye interface.

The key Radi-Eye design consideration that encompasses all design aspects of the interface are the natural behaviours of the eyes, head, and their coordination. Feedback supports and guides the user through the interaction rather than disturbing the interaction flow. For example, feedback is always shown on the outer rings guiding the user towards the next steps of the interaction when placing a building in the city builder (fig. 10). Also, the interface design should keep users within a comfortable eye-in-head position. In the media player, feedback can be safely displayed in the centre zone as the distance between the centre and the widget positions are small (fig. 7), while feedback displayed in the centre zone for the build menu could potentially force users to move outside their comfortable eye-in-head range (fig. 10). All our application interfaces were positioned centrally or approximately central in front of the user's head when invoked to allow gaze exploration in all directions.

In addition to the eye-in-head position, neck ergonomics is an important factor in the Radi-Eye interface design. For example, the number of layers in the interface should be considered to avoid neck strain. A small centre zone may be appropriate for large interfaces (city builder) to limit the use of head movement while a shallow menu can use a larger centre zone without causing discomfort. Furthermore, object-independent interfaces can be used if the interacted object is at an uncomfortable neck position (e.g. a roof lamp). Future work could investigate interfaces that adapts to the current neck position and adjusts to ensure neck comfort regardless of the performed interaction. Examples include oval interfaces that are closer to the starting position in an uncomfortable direction, or interfaces that adapt the widgets so that multi-layered widgets are placed along the direction with the widest range.

The study results showed that users can quickly and accurately select Radi-Eye widgets. However, the combination of ring size and the number of widgets on a ring has to be taken into account as they define widgets' arc sizes. The combination of a small ring size and a high number of widgets thus increases the reliance on refined movement and the effect of eye tracking error. As such, fewer widgets on a single ring allow small ring sizes which reduces selection time and extraneous head movement, while also reducing search time and the need for precise head movements for selection. However, if a large number of widgets is necessary for object control, offloading widgets to outer rings via nested interaction or via the replace ring feature as shown in the media player allows easy selection (fig. 6). Finally, placing frequently used widgets along the cardinal axes allows faster and more comfortable selection, as shown in the study results.

Look & Cross builds on fundamental eye-head coordination insights by utilising a head-crossing metaphor that caters to a user's natural sequence of gaze shifts. Our results show that the technique is intuitive and easy to learn as users only have to add a small extra head movement for a natural gaze shift to turn into a selection. However, the offset between the eyes and head is large enough during exploration so that users can safely explore the interface or environment without triggering accidental selections. This capability allows users to easily avoid selection of unwanted targets which is useful for the composition of commands (fig. 10) or selection of multiple widgets in a list (fig. 9c-f).

At the core of Look & Cross lies the distribution of different parts of the interaction to the eyes and head. This allows the user to utilise the relative strengths of the modalities. The eyes enable fast and effortless search and hover interaction without risking accidental selections. The stable head can then be used for precise selection confirmation. In our work, Look & Cross was combined with a radial interface to extend the available interactions as highlighted by the diverse set of widgets used in our applications. However, Look & Cross is not limited to the usage in radial interfaces and can be extended as a general technique for object selection and manipulation in 3D or 2D environments.

While eye-head alignment was accurate for selection in our user study and previous work [37], it has not been evaluated for continuous or nested interactions. Similarly, our user study primarily focussed on fundamental menu properties and their impact on

selection performance. As such, future work could evaluate the use of gaze and head for continuous or nested interaction to discover limitations. In the media player, we mitigated the effect of limited head precision by using widgets to complement each other (fig. 7). Alternatively, adjusting the control-display ratio has been effective for head-based selection refinement and could increase precision during slider interaction [20]. In nested interaction and command composition, we introduced a delay (0.4s) before hiding nested items to avoid interaction interruption caused by eye tracker jitter, or unexpected saccades moving outside the widgets. Also, the head was used as an "anchor" to keep nested items open if hovering over an item in the hierarchy while the eyes move away. Finally, we carefully placed feedback to ensure that the gaze would not wander outside the widget hierarchy and hide the nested widgets.

We can also extend the Look & Cross concept to other modalities. In this work, we utilised the synergetic relationship of the eyes and head for interaction, and similar relationships exist between other modalities. For example, the eyes and hands are highly connected as we use our eyes to guide our hands, and previous research has leveraged this coordinated relationship for interaction in 3D environments [28, 38]. Therefore, we could imagine a crossing-based technique that uses gaze for object activation and the hands for crossing. The combination of different modalities offers an exciting future research direction, where the choice of modalities will have a significant effect on the technique's capabilities. Finally, future work can also further extend Radi-Eye by developing new widgets to increase interface expressiveness or by extending existing widgets via for example having nested widgets expand inwards to minimise head movement or adjusting the widget functionality to be dependent on the crossing direction.

7 CONCLUSION

We introduced Radi-Eye, a novel radial interface for on-demand and hands-free object control via gaze and head input that we validated in application prototypes and a user study. The reliance on only gaze- and head-movements for input is useful in situations where the hands are unavailable, and highly relevant for contexts where the hands and body is limited in movement. Furthermore, combining gaze and head for interaction extends their capabilities to allow effortless interface exploration and hover interaction, and provides users with more stable selection, feedback, and alleviates Midas Touch issues to support freedom to roam the interface and the surrounding environment with gaze and head movements without compromising interaction efficiency.

The radial interface is efficient in supporting gaze- and head-based interaction, and together with Look & Cross provide users with a wide variety of widgets for discrete and continuous interaction, and heterogeneous input via command composition for expressive hands-free control. However, the choice of interface layout, dimensions, and feedback can have significant impact on the user performance and experience of the interface. As such, careful consideration of these factors have to be made to support easy interaction that does not disturb natural gaze or head behaviours. Furthermore, the Radi-Eye pop-up design supports control of both visible and occluded objects, and supporting users with both object-dependent and object-independent interfaces is important to allow comfortable object control in any interaction context without having to perform significant body movements.

REFERENCES

[1] Johnny Accot and Shumin Zhai. 2002. More than Dotting the i's — Foundations for Crossing-Based Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) *(CHI '02)*. Association for Computing Machinery, New York, NY, USA, 73–80. https://doi.org/10.1145/503376.503390

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

[3] Georg Apitz, François Guimbretière, and Shumin Zhai. 2008. Foundations for Designing and Evaluating User Interfaces Based on the Crossing Paradigm. *ACM Trans. Comput.-Hum. Interact.* 17, 2, Article 9 (May 2008), 42 pages. https://doi.org/10.1145/1746259.1746263

[4] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *ACM Comput. Surv.* 49, 4, Article 60 (Dec. 2016), 41 pages. https://doi.org/10.1145/3002171

[5] Richard Bates and Howell Istance. 2003. Why are Eye Mice Unpopular? A Detailed Comparison of Head and Eye Controlled Assistive Technology Pointing Devices. *Universal Access in the Information Society* 2, 3 (Oct. 2003), 280–290. https://doi.org/10.1007/s10209-003-0053-y

[6] Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. 2006. Precise Selection Techniques for Multi-Touch Screens. 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, 1263–1272. https://doi.org/10.1145/1124772.1124963

[7] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of Eye-gaze over Head-gaze-based Selection in Virtual and Augmented Reality Under Varying Field of Views. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. ACM, New York, NY, USA, Article 1, 9 pages. https://doi.org/10.1145/3206343.3206349

[8] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. 1988. An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Washington, D.C., USA) *(CHI '88)*. ACM, New York, NY, USA, 95–100. https://doi.org/10.1145/57167.57182

[9] Debaleena Chattopadhyay and Davide Bolchini. 2014. Touchless Circular Menus: Toward an Intuitive UI for Touchless Interactions with Large Displays. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (Como, Italy) *(AVI '14)*. Association for Computing Machinery, New York, NY, USA, 33–40. https://doi.org/10.1145/2598153.2598181

[10] K. Das and C. W. Borst. 2010. An evaluation of menu properties and pointing techniques in a projection-based VR environment. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 47–50. https://doi.org/10.1109/3DUI.2010.5444721

[11] Matthew M. Davis, Joseph L. Gabbard, Doug A. Bowman, and Dennis Gracanin. 2016. Depth-based 3D gesture multi-level radial menu for virtual object manipulation. In *2016 IEEE Virtual Reality (VR)*. IEEE, 169–170. https://doi.org/10.1109/VR.2016.7504707

[12] Edward G. Freedman. 2008. Coordination of the Eyes and Head During Visual Orienting. *Experimental Brain Research* 190, 4 (Oct. 2008), 369–387. https://doi.org/10.1007/s00221-008-1504-8

[13] S. Gebhardt, S. Pick, F. Leithold, B. Hentschel, and T. Kuhlen. 2013. Extended Pie Menus for Immersive Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 19, 4 (April 2013), 644–651. https://doi.org/10.1109/TVCG.2013.31

[14] D. Gerber and D. Bechmann. 2005. The spin menu: a menu system for virtual environments. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005.* IEEE, 271–272. https://doi.org/10.1109/VR.2005.1492790

[15] Don Hopkins. 1991. The Design and Implementation of Pie Menus. *Dr. Dobb's J.* 16, 12 (Dec. 1991), 16–26.

[16] Anke Huckauf and Mario H. Urbina. 2008. Gazing with pEYEs: Towards a Universal Input for Various Applications. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (Savannah, Georgia) *(ETRA '08)*. ACM, New York, NY, USA, 51–54. https://doi.org/10.1145/1344471.1344483

[17] Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. 2012. Gaze Gestures or Dwell-based Interaction?. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. ACM, New York, NY, USA, 229–232. https://doi.org/10.1145/2168556.2168602

[18] Robert J. K. Jacob. 1990. What You Look at is What You Get: Eye Movement-based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90)*. ACM, New York, NY, USA, 11–18. https://doi.org/10.1145/97243.97246

[19] Shahram Jalaliniya, Diako Mardanbegi, and Thomas Pederson. 2015. MAGIC Pointing for Eyewear Computers. In *Proceedings of the 2015 ACM International*

Symposium on Wearable Computers (Osaka, Japan) *(ISWC '15)*. Association for Computing Machinery, New York, NY, USA, 155–158. https://doi.org/10.1145/2802083.2802094

[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, Article 81, 14 pages. https://doi.org/10.1145/3173574.3173655

[21] Michael Land and Benjamin Tatler. 2012. *Looking and Acting: Vision and Eye Movements in Natural Behaviour*. Oxford University Press, United Kingdom. https://doi.org/10.1093/acprof:oso/9780198570943.001.0001

[22] Zhen Li, Michelle Annett, Ken Hinckley, Karan Singh, and Daniel Wigdor. 2019. HoloDoc: Enabling Mixed Reality Workspaces That Harness Physical and Digital Content. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 687, 14 pages. https://doi.org/10.1145/3290605.3300917

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

[24] Diako Mardanbegi, , Benedikt Mayer, Ken Pfeuffer, Shahram Jalaliniya, Hans Gellersen, and Alexander Perzl. 2019. EyeSeeThrough: Unifying Tool Selection and Application in Virtual Environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 474–483. https://doi.org/10.1109/VR.2019.8797988

[25] Diako Mardanbegi, Tobias Langlotz, and Hans Gellersen. 2019. Resolving Target Ambiguity in 3D Gaze Interaction through VOR Depth Estimation. 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, Article 612, 12 pages. https://doi.org/10.1145/3290605.3300842

[26] Pallavi Mohan, Wooi Boon Goh, Chi-Wing Fu, and Sai-Kit Yeung. 2018. DualGaze: Addressing the Midas Touch Problem in Gaze Mediated VR Interaction. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 79–84. https://doi.org/10.1109/ISMAR-Adjunct.2018.00039

[27] Florian Müller, Joshua McManus, Sebastian Günther, Martin Schmitz, Max Mühlhäuser, and Markus Funk. 2019. Mind the Tap: Assessing Foot-Taps for Interacting with Head-Mounted Displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 477, 13 pages. https://doi.org/10.1145/3290605.3300707

[28] Ken Pfeuffer, Matthias J. Geiger, Sarah Prange, Lukas Mecke, Daniel Buschek, and Florian Alt. 2019. Behavioural Biometrics in VR: Identifying People from Body Motion and Relations 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, 1–12. https://doi.org/10.1145/3290605.3300340

[29] T. Piumsomboon, G. Lee, R. W. Lindeman, and M. Billinghurst. 2017. Exploring natural eye-gaze-based interaction for immersive virtual reality. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 36–39. https://doi.org/10.1109/3DUI.2017.7893315

[30] Majid Pourmemar and Charalambos Poullis. 2019. Visualizing and Interacting with Hierarchical Menus in Immersive Augmented Reality. In *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry* (Brisbane, QLD, Australia) *(VRCAI '19)*. Association for Computing Machinery, New York, NY, USA, Article 30, 9 pages. https://doi.org/10.1145/3359997.3365693

[31] Yuan Yuan Qian and Robert J. Teather. 2017. The Eyes Don't Have It: An Empirical Comparison of Head-Based and Eye-Based Selection 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, 91–98. https://doi.org/10.1145/3131277.3132182

[32] Krystian Samp and Stefan Decker. 2010. Supporting Menu Design with Radial Layouts. In *Proceedings of the International Conference on Advanced Visual Interfaces* (Roma, Italy) *(AVI '10)*. Association for Computing Machinery, New York, NY, USA, 155–162. https://doi.org/10.1145/1842993.1843021

[33] A. Santos, T. Zarraonandia, P. Díaz, and I. Aedo. 2017. A Comparative Study of Menus in Virtual Reality Environments. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. Association for Computing Machinery, New York, NY, USA, 294–299. https://doi.org/10.1145/3132272.3132277

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

[35] Ludwig Sidenmark, Christopher Clarke, Xuesong Zhang, Jenny Phu, and Hans Gellersen. 2020. Outline Pursuits: Gaze-Assisted Selection of Occluded Objects in Virtual Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376438

[36] Ludwig Sidenmark and Hans Gellersen. 2019. Eye, Head and Torso Coordination During Gaze Shifts in Virtual Reality. *ACM Trans. Comput.-Hum. Interact.* 27, 1, Article 4 (Dec. 2019), 40 pages. https://doi.org/10.1145/3361218

[37] Ludwig Sidenmark and Hans Gellersen. 2019. Eye&Head: Synergetic Eye and Head Movement for Gaze Pointing and Selection. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. ACM, New York, NY, USA, 1161–1174. https://doi.org/10.1145/3332165.3347921

[38] 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* (Denver, Colorado) *(ETRA '19)*. ACM, New York, NY, USA, Article 6, 9 pages. https://doi.org/10.1145/3314111.3319815

[39] Ludwig Sidenmark, Diako Mardanbegi, Argenis Ramirez Gomez, Christopher Clarke, and Hans Gellersen. 2020. BimodalGaze: Seamlessly Refined Pointing with Gaze and Filtered Gestural Head Movement. In *Proceedings of the 12th ACM Symposium on Eye Tracking Research & Applications (ETRA '20)*. ACM, New York, NY, USA, 9. https://doi.org/10.1145/3379155.3391312

[40] John S. Stahl. 1999. Amplitude of human head movements associated with horizontal saccades. *Experimental Brain Research* 126, 1 (apr 1999), 41–54. https://doi.org/10.1007/s002210050715

[41] Ahmed N. Sulaiman and Patrick Olivier. 2008. Attribute Gates. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (Monterey, CA, USA) *(UIST '08)*. Association for Computing Machinery, New York, NY, USA, 57–66. https://doi.org/10.1145/1449715.1449726

[42] Vildan Tanriverdi and Robert J. K. Jacob. 2000. Interacting with Eye Movements in Virtual Environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (The Hague, The Netherlands) *(CHI '00)*. ACM, New York, NY, USA, 265–272. https://doi.org/10.1145/332040.332443

[43] Huawei Tu, Susu Huang, Jiabin Yuan, Xiangshi Ren, and Feng Tian. 2019. Crossing-Based Selection with Virtual Reality Head-Mounted Displays. 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, Article 618, 14 pages. https://doi.org/10.1145/3290605.3300848

[44] Mario H. Urbina, Maike Lorenz, and Anke Huckauf. 2010. Pies with EYEs: The Limits of Hierarchical Pie Menus in Gaze Control. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (Austin, Texas) *(ETRA '10)*. ACM, New York, NY, USA, 93–96. https://doi.org/10.1145/1743666.1743689

[45] Jacob O. Wobbrock and Krzysztof Z. Gajos. 2008. Goal Crossing with Mice and Trackballs for People with Motor Impairments: Performance, Submovements, and Design Directions. *ACM Trans. Access. Comput.* 1, 1, Article 4 (May 2008), 37 pages. https://doi.org/10.1145/1361203.1361207

[46] Wenge Xu, Hai-Ning Liang, Yuxuan Zhao, Difeng Yu, and Diego Monteiro. 2019. DMove: Directional Motion-based Interaction for Augmented Reality Head-Mounted Displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 444, 14 pages. https://doi.org/10.1145/3290605.3300674

[47] Wenge Xu, Hai-Ning Liang, Yuxuan Zhao, Tianyu Zhang, Difeng Yu, and Diego Monteiro. 2019. RingText: Dwell-free and hands-free Text Entry for Mobile Head-Mounted Displays using Head Motions. *IEEE Transactions on Visualization and Computer Graphics* 25, 5 (May 2019), 1991–2001. https://doi.org/10.1109/TVCG.2019.2898736

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