UI Mobility Control in XR: Switching UI Positionings between Static, Dynamic, and Self Entities

Siyou Pei*
University of California, Los Angeles
Los Angeles, California, USA
sypei@ucla.edu

David Kim
Google Research
Zurich, Switzerland
kidavid@google.com

Alex Olwal
Google Research
Mountain View, California, USA
olwal@acm.org

Yang Zhang
University of California, Los Angeles
Los Angeles, California, USA
yangzhang@ucla.edu

Ruofei Du†
Google Research
San Francisco, California, USA
me@duruofei.com

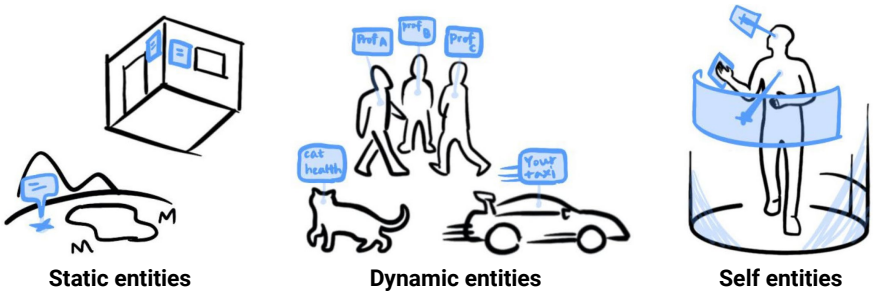

Figure 1: UI mobility refers to the UI transitions between different entities. Static Entity: fixed objects or surfaces, *e.g.,* walls, tables, or floor. Dynamic Entity: living creatures or movable objects, *e.g.,* animals or vehicles. Self Entity: attachable surfaces/positions in one's personal space, *e.g.,* head-anchored, body-anchored, or attached to hands. From left to right, the three distinct UI placements: UIs affixed to static entities, dynamic entities, or self entities. With UI mobility control, users can switch UI placement between these entities.

ABSTRACT

Extended reality (XR) has the potential for seamless user interface (UI) transitions across people, objects, and environments. However, the design space, applications, and common practices of 3D UI transitions remain underexplored. To address this gap, we conducted a need-finding study with 11 participants, identifying and distilling a taxonomy based on three types of UI placements — affixed to static, dynamic, or self entities. We further surveyed 113 commercial applications to understand the common practices of 3D UI mobility control, where only 6.2% of these applications allowed users to transition UI between entities. In response, we built interaction prototypes to facilitate UI transitions between entities. We report on results from a qualitative user study (N=14) on 3D UI mobility control using our FingerSwitches technique, which suggests that perceived usefulness is affected by types of entities and environments. We aspire to tackle a vital need in UI mobility within XR.

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**; **Virtual reality**; **Graphical user interfaces**; *Gestural input*.

KEYWORDS

UI Mobility, Extended Reality, Virtual Reality, User Interface Behaviors, Hand Gestures, Mode Switching, Embodied Interactions

ACM Reference Format:
Siyou Pei, David Kim, Alex Olwal, Yang Zhang, and Ruofei Du. 2024. UI Mobility Control in XR: Switching UI Positionings between Static, Dynamic, and Self Entities. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24), May 11–16, 2024, Honolulu, HI, USA*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3613904.3642220

1 INTRODUCTION

The prevalence of mobile devices has made UI transitions desirable — it has become increasingly common for UIs (user interfaces) to traverse various screen devices, such as phones, smartwatches, external monitors, laptops, and projectors. Consider the following scenario: *A tourist is watching an online video of "top-rated local restaurants" on her computer. Before she can finish the video, her*

*Project conducted while the first author interned at Google Labs.
†Corresponding author.

CHI '24, May 11–16, 2024, Honolulu, HI, USA

ACM ISBN 979-8-4007-0330-0/24/05
https://doi.org/10.1145/3613904.3642220

taxi arrives. She rushes downstairs and gets into the taxi, where she takes out her phone, opens the online video app, and searches for the video that she wants to resume watching. From this example, we see how user increasinhly expect UIs to travel across digital screens, based on users' in-situ intent. Consequently, seamless changing UI's positioning across screens of devices [5, 8, 20, 38, 40, 48] has been extensively researched to meet the growing demands for a more flexible and seamless user experience. Compared with cross-device configuration, the realization of Extended Reality (XR) has brought about new affordances and applications [18, 29], leading to unique opportunities and challenges for UI transition [36]. On the one hand, with spatial tracking displays [28], UIs are no longer bounded by screens on physical devices. On the other hand, immersive XR anchors a larger design space for UI interaction and spatial manipulation, which creates new research questions on *the design space, applications, common practices, and user perception of 3D UI positioning transition in XR.*

In particular, we aim to tackle four key Research Questions (RQs):

- RQ1: What is the design space of controlling 3D UI placement?
- RQ2: What are the *scenarios* and *applications* of 3D UI mobility?
- RQ3: What are the *common practices* on 3D UI mobility control in commercial products?
- RQ4: How do people perceive 3D UI mobility control in different scenarios?

In this paper, we conducted a need-finding study to answer RQ1 and RQ2. The need-finding study distills three types of entities that a 3D UI can be attached to. As shown in Figure 1, these host entities span both **static entities**, such as fixed objects or surfaces, and **dynamic entities**, including living creatures or moving objects. Finally, **self entities** are attachable surfaces/positions in one's personal space, including head zone, torso surround, and hand. We introduce a term, "**3D UI mobility**", to concisely describe the transition of a UI between different entities in 3D. Additionally, "**3D UI mobility control**" refers to the control of 3D UI mobility. Based on the old and new entity of a transition, we categorize rich scenarios and applications of 3D UI mobility into a taxonomy for future reference.

Furthermore, we performed a commercial survey of 113 XR applications to summarize the common practice on 3D UI mobility control and highlight the gap between existing work and user expectation (RQ3). To collect user perception of 3D UI mobility control, we devised an example interaction technique, FingerSwitches, as a probe for an on-the-spot user study with 14 participants (RQ4), and present findings for XR designers. We aim for our work to address an important gap in UI mobility within XR by exploring these research questions.

2 RELATED WORK

Our work is motivated by prior art in design frameworks of spatial UI positioning (§ 2.1) and UI mobility (§ 2.2). We also briefly discuss prior research about mode-switching techniques in XR (§ 2.3) for a comprehensive survey.

2.1 Reference Frames of 3D UIs

Research in spatial UI positioning has explored various design frameworks with rich dimensions, among which "reference frame" is the most convenient dimension to describe the position of a movable UI. Feiner *et al.* [17] present three types of UIs in Virtual Reality (VR): display-fixed, surround-fixed, and world-fixed windows, which can be affixed to a static location or a moving object. In Air Pointing [10], Cockburn *et al.* classify spatial UI locations into absolute, relative-to-object, relative-to-body, and relative-to-device locations. LaViola Jr *et al.* [31] introduce one more category, head-referenced UI. In comparison, Ethereal planes [16] present a two-class taxonomy, egocentric and exocentric UIs, and has brought up UI's movability, *i.e.,* the ability to move a UI around related to a reference, as a fundamental property. The introduction of UI movability reflects the growing momentum in XR interaction research. UI mobility, which involves movement of a UI across different reference frames, goes further than movability.

As mentioned in these works, the positioning of a spatial UI can be described by the reference frame of the specific entity it is bound to, assuming the UI is in proximity to the entity. In our work, we generalize the entity classification to include all aforementioned UI positionings, and enable transitions between them.

2.2 UI Mobility

Lu *et al.* [36] pioneered the use of the wording "UI mobility" to describe moving UIs in Augmented Reality (AR). In our paper, we define 3D UI mobility as the movement of a UI in XR between different entities (static, dynamic, or self entities). For consistency, we also define 2D UI mobility — the movement of a UI across screens — with 2D being a subset of 3D context.

For 2D UI mobility, research on cross-device interaction [8] has laid the foundation by highlighting interface transition, which may happen between a user and a public display [6, 22, 44], between collaborators' devices or multiple personal screens [21, 27, 39, 48]. As XR gains popularity and UI positions become more diverse, researchers have explored interaction techniques to reposition UIs in 3D. Prior work investigated ways to change 3D UI positions and orientation with absolute coordinate adjustment [11, 13, 23], surface-based alignment [14, 15, 34, 43], or automatic layout arrangement [2, 37]. Furthermore, Lu *et al.* [36] explored automated dynamic UI placements in 3D which can follow users.

Closer to our paper, Lages and Bowman [30] propose an adaptive workspace that allows display windows to automatically transition from static to following the user in 3D when they walk around. Lu et al. [35] approach 3D UI mobility from the viewpoint of an "information access method" and enable switching 3D UI positioning from residing in eye periphery to staying in FoV with a gaze-summon technique in AR. Lee et al. [32] regard 3D UI mobility control as "3D window management", conveniently adjusting the scale and static position of a 3D UI. Apple Vision Pro [1] also demonstrates how users can drag a UI by pinching at its bottom bar. These works have drawn attention to user control of 3D UI mobility in real-world contexts. However, these approaches do not fully cover the design space of 3D UI mobility, as shown in Table 1, whose dimensions are explained in § 3. In the rest of the paper, UI mobility stands for 3D UI mobility unless explicitly clarified.

Project	Static →			Dynamic →			Self →		
	Static	**Dynamic**	**Self**	**Static**	**Dynamic**	**Self**	**Static**	**Dynamic**	**Self**
Pick-and-Drop [48]	✓	✓		✓	✓				
View Management [2]	✓				✓				
Shoot & Copy [6]		✓							
Marquardt *et al.* [39]				✓	✓				
Conductor [21]					✓				
Memory Stones [27]				✓	✓				
SnapToReality [43]	✓								
Paay *et al.* [44]				✓					
Projective Windows [32]	✓								
Lages *et al.* [30]			✓				✓		
Plane, Ray, and Point [23]	✓								
Hartmann *et al.* [22]	✓	✓		✓					
Glanceable AR [35]									✓
Embodied Axes [11]					✓				
Armstrong [34]									✓
Luo *et al.* [37]	✓								
Lu *et al.* [36]	✓		✓				✓		✓
FingerSwitches	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: The design space of related work on UI mobility shows the possible UI transitions between entities. The first and second row denote the kind of entity a UI transitions starts, and ends at. For example, the third column's "Static →" and "Dynamic" refers to a UI that moves from a static entity to a dynamic entity. The design space highlights the novelty of FingerSwitches in enabling UI mobility across static, dynamic, and self entities.

2.3 Mode Switching Techniques in XR

Mode switching allows a limited number of input methods to map to a broader set of commands for computers. Common mode-switching techniques involve hardware buttons [9, 52, 55], speech [4, 52, 57], gaze [25], head motions [42, 51] or hand gestures [4, 22, 42, 45, 52, 53, 56, 57], and have been adopted for headset-based XR and phone-based XR. Our research also applied mode-switching techniques but investigated it in a specific domain, UI mobility control, *i.e.,* using mode-switching techniques to change 3D UI positioning.

3 NEED-FINDING STUDY

In this section, we present a need-finding study to understand the use cases of UI mobility control. From the collected data, we recognized three kinds of host entities to which a UI could be attached: **static entities**, **dynamic entities**, and **self entities**. Using this classification, we defined UI placement and subsequently categorized the transitions based on the type of entity a UI was moving between. The various combinations of host entities lead to a taxonomy of "UI mobility" that refers to the transitions of UI positioning between different entities.

3.1 Methods

Participants. We invited 11 participants (3F, 8M) to our need-finding study. Table 2 shows the demographic and background information of these 11 participants. Their ages ranged from 25 to 40 years (M=29, SD=4.65). Their VR experience ranged from novices to experts. The study was approved by our local Institutional Review Board.

Procedure. The need-finding study consisted of (1) introduction and demographics (10 min), (2) perception of UI positioning (15 min), and (3) usefulness of UI repositioning (20 min). At the beginning of the study, we collected demographic information about their varied VR experience and diverse backgrounds. We also introduced the study's goal: to understand users' perceptions of UI repositioning in XR, assuming a future with ubiquitous head-worn displays. Next, we presented visuals of scattered UI elements in 3D space through screenshots from XR apps and relevant videos. Participants participated the study online, independently, using their own slides within a shared slide deck. To understand their perceptions of UI positioning, we posed open-ended questions regarding how different placements affected their feelings and usage of a UI. Later on, we introduced the concept of UI repositioning by inviting participants to envision the possibility of freely relocating a UI in XR. To investigate the usefulness, we prompted participants to describe scenarios in which they would find UI repositioning beneficial. They were asked to elaborate on their motivation with example applications in detail within 20 minutes.

Data Analysis. We collected the demographic information, responses about perception of UI positioning, and answers about the usefulness of UI repositioning from our 11 participants. Two researchers conducted a coding and thematic analysis on the collected data. For question (2), our initial coding started with participants' feelings, labeling the types of UI placement they perceived as "personal", "public" *etc.* For question (3), our initial codes were derived from environments (*e.g.,* "classroom", "zoo", "conference" *etc.*), or from user scenarios (*e.g.,* 'information sharing", "facilitate interaction" *etc.*). Meanwhile, we observed the pattern that they frequently

ID	Age	Gender	Education Level	Occupation/Major	Have you used VR	Type of devices	First time of use	How often do you use VR devices	Duration of each use	Handedness
U1	31	M	M.S.	Software Engineer	Yes, a lot	Quest 2	5 years ago	400 times in life	30-120 min	Right
U2	36	M	Ph.D.	Research Scientist	Yes, a little	Google cardboard, Quest Pro	5 years ago	Once a few months	30–60 min	Right
U3	29	M	Ph.D.	Research Scientist	Yes, a little	Quest 2, Quest Pro	1 year ago	5–10 times in life	20–30 min	Left
U4	32	F	M.S.	Designer	Yes, some	Quest 1, Quest 2, Quest Pro	2 years ago	monthly	20–30 min	Right
U5	27	M	M.S.	Computer Science	Yes, a little	Quest 2	5 years ago	Few times a year	<30 min	Right
U6	40	M	Ph.D.	Software Engineer	Yes, some	Quest Pro	recently	Every weekday	5–10 min	Left
U7	29	M	Ph.D.	Hardware Engineer	Yes, a lot	Vive, ViveProEye, Quest 2, Quest Pro	7 years ago	Weekly	20 min	Right
U8	35	M	M.S.	Art Director	Yes, a lot	Rift/Quest, Vive, PSVR, Google Cardboard/Daydream, Disney Quest, Varjo	9 years ago	Weekly	20–30 min	Right
U9	26	F	B.S.	Electrical Engineering	Yes, a lot	Quest 2	1 year ago	Weekly	30–120 min	Right
U10	25	F	M.S.	Computer Science	Yes, a lot	Quest 2	1 year ago	Weekly	60 min	Right
U11	28	M	M.S.	Mechanical Engineering	Yes, a little	Quest 2	2 years ago	Less than once a year	20 min	Right

Table 2: Demographic and background information of participants (U1-U13) in need-finding study.

described UI positioning in relation to a host entity, *i.e.,* an entity that the UI is attached to/placed around. Mentioned host entities included, but were not limited to, "house", "kid", "professor", "fridge", "the user themself", "car", "monitor", and "book". We categorized host entities into three types: **static entities**, **dynamic entities**, and **self entities**. After discussion, we decided to continue coding the entire dataset with this entity classification as themes, since a host entity implied both spatial positioning and the function of a UI. In this manner, we generated findings on how user perception of a UI was influenced by its host entity. We also created an application taxonomy of UI repositoning based on combinations of old and new entities, *i.e.,* the entity a UI was moved from and the one it was moved to. We introduce the term, "UI mobility", to describe the UI repositioning between entities.

3.2 Findings

In this section, we summarize findings on entity classificaiton and entity-based UI positionings, user perception/understanding of UI positioning, and the usefulness of UI repositioning.

Entity Classification and UI Positionings. As shown in Figure 1, the three types of host entites are

- **Static Entity**: fixed objects or surfaces (*e.g.,* walls, tables, floor)
- **Dynamic Entity**: living creatures or moving objects (*e.g.,* walking people, cats, vehicles)
- **Self Entity**: attachable surfaces/positions in a user's personal space (*e.g.,* head zone, around torso, hand)

Typically, we position the UI near a host entity of interest. This approach has led to three distinct types of UI positionings corresponding to the three types of entities: UIs attached to static entities, dynamic entities, or self entities, as illustrated by blue widgets in Figure 1. For example, it is logical to place a road sign UI at a road crossing, whereas a namecard UI might be designed to follow a person. A static entity is paired with a static UI, while the UI of a dynamic entity moves along with its moving host. In essence, a UI is often aligned with the reference frame of its host entity and remains in close proximity to that entity.

However, there are exceptions to these typical scenarios. One such case occurs when users wish to use the in-situ coordinates of a dynamic entity to anchor a UI, but do not want the UI to follow the entity. In this scenario, the UI is static. Movable objects or individuals, at that moment, act as an anchor point to fix the UI in place. Consequently, the UI remains in its original position, even if the object or person moves later.

Another case is when users want to attach a movable UI to a static entity, *e.g.,* floor, statue, or wall. The UI will move together with a statue if the statue leaves its place. However, static entities will hardly move in a human's lifespan, so the entity-tracking behavior will never take effect. Therefore, it is equivalent to assigning a static UI to a static entity.

In this paper, we focus on common cases instead of corner cases unless explicitly stated otherwise. In the following sections, "static UI" represents a UI attached to a static entity, anchored in the world coordinate system; "dynamic UI" denotes a UI that is attached to and moves with a dynamic entity. Similarly, "self UI" means a UI attached to and moving with a self entity. In essence, we categorize UIs based on the types of entities that they are associated with.

Perception of UI Positioning. UI positioning can be described by the types of host entities to which a UI is attached. Analyzing participants' responses revealed distinct perceptions and associations with the three types of UI positionings. Participants perceived static UIs as public (6/11), and geo-related (8/11), making them suitable for public displays and geo-based reminders/navigation. In contrast, self UIs were seen as more personal and private (5/11) while no one regarded static/dynamic UIs as personal and private, hence more appropriate for sensitive activities like checking bank accounts, messages, or reading presentation hints. Additionally, participants reported self UIs were also more readily available and convenient to access (9 out of 11), ideal for calendars, control panels, and utility tools. At the same time, all participants considered dynamic UIs closely tied to their host entities (people, animals, handheld objects or vehicles). Their perferred usage of dynamic UIs was to display closely related information, such as nametags, dialogue captions of

a. Static to Static	b. Dynamic to Static	c. Self to Static
➪ Reposition a router UI for Wi-Fi configuration (information access) ➪ Move an agenda UI from a large display to the tabletop to give room for more workspace area (information organization) ➪ Rearrange UI panels around to customize my virtual workspace (information organization) ➪ Place a plant UI from the pot to above its leaves (information access) ➪ Stick photo gallery UI on the wall and adjust their sizes (decoration)	➪ Pin an info UI from cat food to the wall (information organization) ➪ Share the map UI on a coffee table to plan a hiking trip with friends (information sharing) ➪ Collect idea bubbles from students to a whiteboard for better analysis (information organization) ➪ Put family name tags onto a Christmas tree (decoration) ➪ Move a Nest speaker's UI to the wall to share song details (information sharing)	➪ Pin my reminder "drink water" to the lab entrance (in-situ reminder) ➪ Share my screen during meeting or presentation (information sharing) ➪ Edit and place instructions beside a 3D printer in the real world (in-situ reminder) ➪ Pin my access UI to games in the living room so I can control gaming time in bedroom (limit interaction) ➪ Project email UIs from glasses to a larger display to show details (interaction access)
d. Static to Dynamic	**e. Dynamic to Dynamic**	**f. Self to Dynamic**
➪ Pull a fancy graffiti pattern from the wall to my wallet (decoration) ➪ Pin a static agenda UI to team members during a conference (task designation) ➪ Share a UI of art work to a friend (information sharing) ➪ Found a useful finance UI and attach it to my credit card (information access) ➪ Grab an outfit UI in a store and apply it to my avatar (decoration)	➪ Transfer a task UI from one robot to another (task designation) ➪ Drag a lovely outfit on a puppy and drag its digital copy to mine (decoration) ➪ Copy group labels from one person to another in a party game (information organization) ➪ Archive a information UI of an animal in a zoo to my personal notebook (decoration) ➪ Move a Nest speaker's UI to the wall to share song details (information sharing)	➪ Pass digital messages to friends (information sharing) ➪ Edit a meeting agenda and share it with the other meeting host (information sharing) ➪ Create notes and attach them to related objects (in-situ reminder) ➪ Drag health statistics from screen space to the kitchen coutertop after exercises (information organization) ➪ Write a review in my screen space and then pin it to a travel spot (information sharing)
g. Static to Self	**h. Dynamic to Self**	**i. Self to Self**
➪ Drag a thermostat UI on the wall to my screen space so I can touch it without walking to it (facilitate interaction) ➪ Pull information from a bus stop to me when it is too far or too crowded ➪ Found a good recipe on YouTube and bring it with me to the kitchen (information access) ➪ Drag a sale promotion UI to me when it interests me (information access) ➪ Grab camera feeds to my eyes from a security camera at the roof top (information access)	➪ Pull product information UIs to me to compare them (information access) ➪ Grab dashboard UI of a taxi to my screen and give ratings (facilitate interaction) ➪ Drag people's digital name cards to me to read and click "collect" (information access, facilitate interaction) ➪ Grab the introduction about specimens in a virtual science museum (information access) ➪ Pull patient case to screen space for diagnosis if I were a doctor (information access)	➪ Move a globe UI from head screen to my hand so it will rotate with hand (facilitate interaction) ➪ Project my virtual watch plate from head space (information organization) ➪ Transfer running statistics UI from head space to side of body (information organization) ➪ Move widgets from head space to wrist interface to save space (information organization) ➪ Reposition a UI from the side to head screen (information access)

Table 3: The need-finding study generated the taxonomy of UI mobility applications. These applications can also be clustered into eight user scenarios (in parentheses) — "information access", "information organization", "information sharing", "facilitate interaction", "task designation", "in-situ reminder", "decoration", and "limit interaction".

people/animals, or properties/specs of items. In summary, we found that UI placement affected user perception of their utility. In particular, static UIs were more likely associated with public/geo-related settings, self UIs resonated more with personal/private settings, and dynamic UIs tied more closely to their host entities.

Usefulness of UI Repositioning. During the need-finding study, participants generated 52 applications of UI repositioning that they considered useful. After merging the repeating ideas, 45 distinct ideas remained (Table 3). Based on their explanation, we coded the applications with eight user scenarios — "information access", "information organization", "information sharing", "facilitate interaction", "task designation", "in-situ reminder", "decoration", and "limit interaction" (The user scenario of each application can be found in parentheses in Table 3). "Information access" refers to gaining access to information on a UI. "Information organization" means adjusting the layout of UIs. "Information sharing" indicates the intent to share information on a UI with other people. "Facilitate interaction" refers to retrieving a UI for more convenient interaction, while "facilitate interaction" means to make the interaction less convenient (*e.g.,* restraint the use to control screen time). "Task designation" means assigning tasks in the form of UI to others. "In-situ reminder" is creating geo-related reminders while "decoration" means changing appearance of something by attaching UIs to it. The wide variety of user scenarios unveils that UIs play a more important role than simply display static information as computing is increasingly integrated into users' physical environment with recent advances in XR that enhance the concept of "spatial computing".

In addition to these user scenarios, we also coded the example applications given what entity a UI was moved between, *i.e.,*, the old and new host entity. **We defined the positional characteristics of a UI based on their host entities as "UI Mobility"**. Furthermore, we refer to the transition of UIs between entities as "UI mobility control". With this definition, we created an application

taxonomy of UI mobility (Table 3). Each block represents one possible combination of old and new host entities. For example, Table 3g is "Static to Self", which means "a UI was attached to static entity and is now attached to a self entity". The block collects scenarios where users move a UI from a static entity to a self entity. In each block, there are several example applications. The first example application of "Static to Self", is "Drag a thermostat UI on the wall to my screen space so I can touch it without walking to it (facilitate interaction)". In this example, the thermostat UI is originally pinned to the wall (static entity). The user then pulls the UI from the wall to her screen space (self entity), placing the UI within her arm's reach to facilitate interaction. The nine-block taxonomy depicts concrete examples of such applications unlocked by UI mobility control.

4 COMMERCIAL SURVEY OF XR APPLICATIONS

In this section, we present a survey of commercial XR apps to understand common practice of UI mobility control and find the gap between existing approaches and our goal — identifying interaction techniques that can enable transitions between all types of UI mobilities.

4.1 Methods

We performed a thorough three-pass review of 113 HoloLens 2 applications[1] in 2023. We chose HoloLens 2 as it is the latest XR headset that supports augmented reality with the largest user base. Following Hruschka *et al.*'s iterative coding methodology [24], two researchers independently reviewer each app, and labeled the UI positionings and mobility control featured in them, if any. After each pass, they revised the codebook together to address all corner cases, and assessed the inter-rater reliability of the labels using Krippendorff's Alpha (α) statistical measure. They ended up using "static", "dynamic", "self" as codes for UI positionings, and combinations (*e.g.,* "static to self") as codes for UI mobility control.

4.2 Results and Findings

Using static, dynamic, and self entities as our final coding rules, the inter-rater reliability of the labels in three passes are presented in Table 4, with $\alpha > 0.963$ in the last around, which means researcher had agreed on the labeling. In this section, we present the state of UI positionings in existing apps, and showcase the common practice of UI mobility control.

Labeling task	1st pass (α)	2nd pass (α)	3rd pass (α)
Types of UI positionings in each app	0.279	0.481	0.963
Types of UI mobility control in each app	0.955	0.997	1.000

Table 4: Krippendorff's alpha inter-rater agreement scores on labeling UI mobility on 113 commercial HoloLens apps.

[1]The applications were obtained from the Microsoft Store collection for HoloLens 2 immersive apps available at: https://microsoft.com/store/collections/HL2ImmersiveApps/hololens, where similar apps of the same company were merged.

Distribution of UI Positionings. Results show that 22/113 (19.5%) apps included all three UI positionings, *i.e.,* they had static UI, dynamic UI and self UI. 46/113 (40.7%) apps had two types of UI (static and dynamic 13.3%, dynamic and self 5.3%, static and self 22.1%). Meanwhile, 35/113 (31.0%) apps had only one type of UI, which consisted of static only (18.6%), dynamic only (6.2%), and self only (6.2%). We visualize the distribution of UI positionings in Figure 2 (a). In the app store, some apps were designed for a specific application or for demo purposes. Therefore, not all of them provided users with rich user interfaces. As we see, only 19.5% of existing apps covered all three kinds of UIs that we came across. There were even 10 (8.8%) apps that had no UIs at all. The design of UIs depends on the goal of each app, *e.g.,* compared with a mature app, a demo app may have fewer UIs or limited interaction. We envision a richer set of UIs for future XR ecosystem in everyday life.

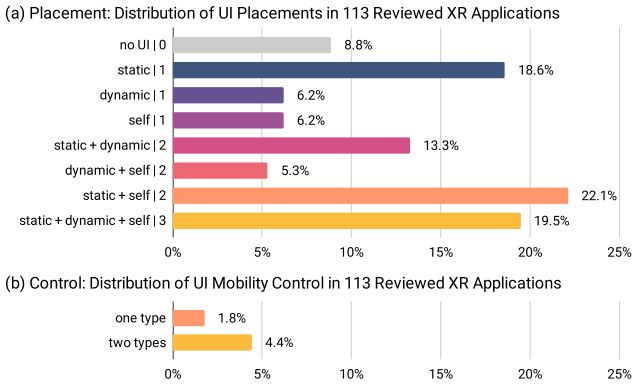

Figure 2: Distribution of UI placements and UI mobility control in 113 reviewed, commercial XR applications. (a) More than half of the apps supported at least two types of UI positionings while only 19.5% supported all three. (b) Only 6.2% of the apps allowed users to control UI mobility.

Common Practice on UI Mobility Control. While 93.8% apps maintained the same host entity of UI as always, 7/113 (6.2%) apps allowed users to switch a UI between different types of entities. The seven apps with UI mobility control are *Altoura*, *Hololight Space*, *ARWing*, *HoloLens 2 On-Stage Live Demo*, *HoloLens Playground*, *Mirage: Virtual Monitors*, and *Power BI*. Among the seven apps, five of them enabled two types of transitions across entities, and two of them allowed one type of transition across entities. The specific type(s) of UI mobility control and their interaction techniques are shown in Figure 3 with screenshots. *HoloLens 2 On-Stage Live Demo* and *HoloLens Playground* are merged as Figure 3D. *Hololens 2 Demo* for the sake of brevity. The interaction modalities chosen by developers — mid-air hand gestures, user proximity, speech — are commonly used in 3D interactions.

Now we explain applications in Figure 3 one by one. The adjacent text in Figure 3 describes the app name, interaction techniques, and the included UI mobility (italicized). *Altoura* is an immerive platform for collaboration. In Figure 3A, the user is working on assembly with a remote collegue. A blue UI is originally a self UI in the user's personal space. The UI displays a tutorial on how

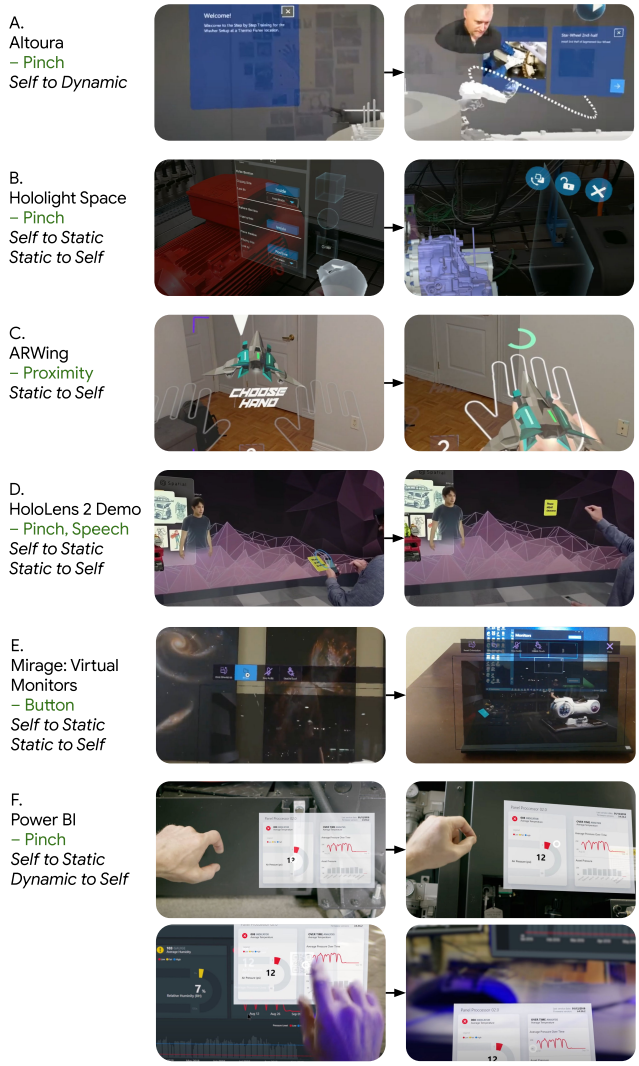

Figure 3: Interaction techniques and types of UI mobility control in commercial XR applications. None of them covers the full taxonomy of UI mobility. For example, the single pinch gesture in *Power BI for HoloLens* cannot distinguish user intent for static/dynamic entities.

to assemble a machine. By pinching, the user is able to connect the UI to the corresponding mechanical component with a dotted line. A self UI is turned into a dynamic UI. Figure 3B is captured in *Hololight Space*, an app for visualization of complex 3D CAD models. It shows a user pinching and dragging a body-anchored UI to the environment. That UI is originally attched to the user, but later stays static in the environment after the gesture. The transition is self to static. Another type of mobility, static to self, is not pictured here. Figure 3C is *ARWing*, a game to control a plane with a hand. The plane UI originally floats in the air (static). When a user's hand approaches the static UI, the UI automatically snaps to the back of the hand. This depicts UI mobility control from static to self, using physical proximity. Figure 3D show UI mobility control in *HoloLens 2 On-Stage Live Demo* and *HoloLens Playground*. On the stage, a local user moves a UI from their hand (self) to a transparent whiteboard (static). A speech command "come here" is also available (for static to self), but not shown in this Figure. Figure 3E is a monitor UI in *Mirage: Virtual Monitors*. A person toggles a "follow" button on that UI to switch between following them or staying static. This application allows UI mobility between static and self states. Figure 3F are screenshots from *Power BI*, a tool to view reports and dashboards in AR. In the first row, a UI is originally head-anchored, attached to the user (self). After a pinch-and-drag, the user moves the UI to the surface of immovable equipment (static). The mobility is from self to static. The second row is a user dragging a UI from a movable dashboard (dynamic) to their head display (self) with a pinch. The mobility is from dynamic to self.

We noted that none of them supported the full taxonomy of UI mobility control in Table 3. For example, the single pinch gesture in *Power BI for HoloLens* cannot distinguish if a user wants to assign a UI to a static or dynamic entity. Similarly, *Mirage: Virtual Monitor* cannot attach a UI to a moving object. The other apps also did not enable UI transitions between all kinds of entities. UI mobility control was still underexplored in commercial XR applications. The lack of references in common practices unveiled the lack of exploration in delivering full UI mobility.

5 UI MOBILITY WITH FINGERSWITCHES

In § 3 and § 4, we identify the limited use of UI mobility control in existing XR apps. To fill the gap, we developed a prototype of fully facilitated UI mobility control with a probing interaction technique. Specifically, we utilized a combination of gaze and pinch, given its popularity as a multimodal pointing-and-selecting interaction technique in XR. We named this technique FingerSwitches. We demonstrate UI mobility control with FingerSwitches in three example scenes in XR — a classroom, a conference room, and a national park. We hope that our probing technique and findings from using it could inspire future designs of generic or app-specific interactions for controlling UI mobility across a wider array of applications.

5.1 Probing Technique: FingerSwitches

Considering the popularity of pinching gestures in existing products and research [3, 26, 33, 46, 49, 50, 54, 60], we decided to adopt pinch-based hand interaction to facilitate three types of UI positionings. As shown in Figure 4, users can selectively rest their thumb tip on a large portion of their index finger — embodying a three-mode switch for the thumb to interact with — to preemptively determine the candidate entity — static, dynamic, or self entity.

For demonstrations of FingerSwitches in action, please refer to the supplementary video, or refer to the the diagram in Figure 5 and screenshots in Figure 4. Our technique leverages both gaze direction and pinch-based gesture, akin to ideas of prior works – M[eye]cro and Gaze + Pinch [47, 58]. As shown in Figure 5, a user first selects a UI by ① looking at it and ② pinching thumb and index finger. The user then actively maintains the pinch while deciding where to place the UI, by ③ choosing a host entity. In this step, one is free to change the thumb's contact position on the index finger, before releasing the pinch. A semi-transparent copy

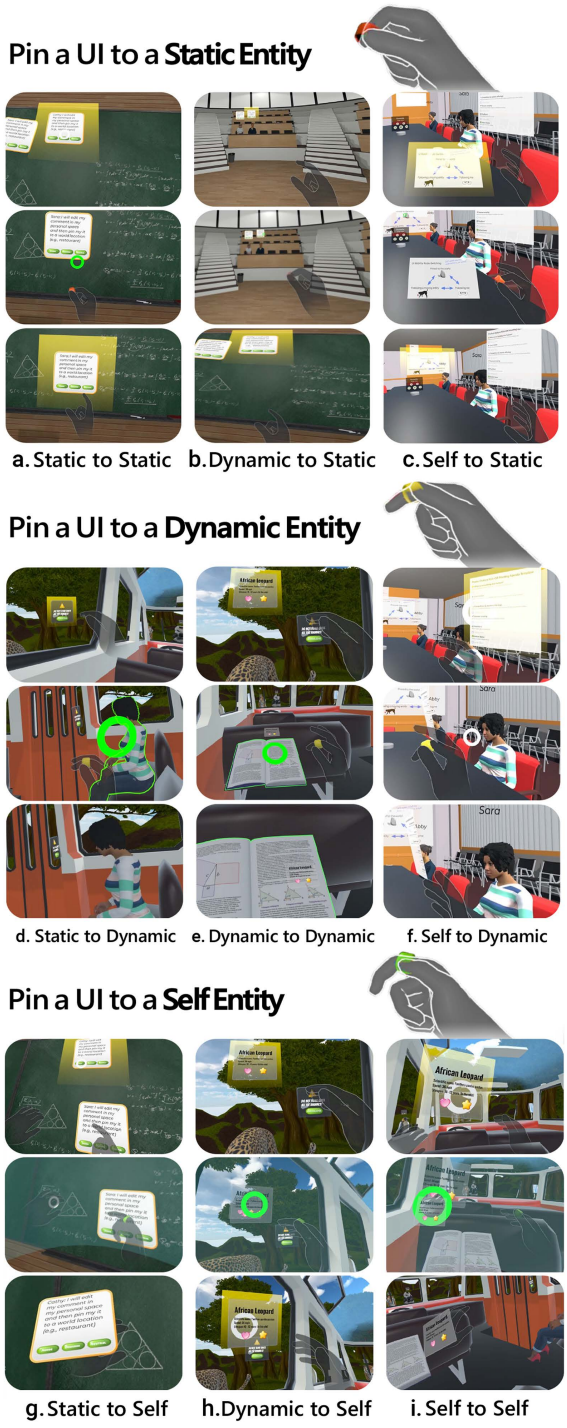

Figure 4: FingerSwitches leverages pinch gestures to pin a UI to a *static entity*, a *dynamic entity*, or a *self entity* in XR. Users first *look at and pinch* a UI, then *look at* the target entity, and finally *release the pinch* to confirm UI transition. Each column (a-i) of screenshots depicts an example application of UI mobility, with detailed descriptions in § 5.2. The green circle indicates the user's gaze.

Figure 5: Interaction state transition diagram for FingerSwitches. ① Look at a UI to move. ② Pinch to confirm UI selection. ③ Look at a candidate target entity. ④ Release the pinch. The location on the index finger where the thumb is released determines the mode ("Static, Dynamic or Self"), and the UI will therefore have corresponding positioning and behavior. The interaction is complete and goes back to idle state.

Figure 6: UI mobility prototype experience in three daily environments: (A) teaching in a classroom, (B) meeting in a conference room, and (C) go sightseeing on a tour bus in a national park.

of the UI will follow the user's gaze, previewing the position where the UI will be anchored. When the user ④ releases pinching, the interaction completes. Specifically, when the user releases at the index finger distal/intermediate/proximal phalange, the target UI will be respectively anchored to a static, dynamic, or self entity.

5.2 Example Applications

From the need-finding study in § 3, we selected three example scenes to implement in XR — a classroom, a conference room, and a national park (Figure 6). Each of them includes a set of example applications that cover the full taxonomy of UI mobility control. We aim to offer users a comprehensive experience of UI mobility in both indoor/outdoor, academic/commercial, and professional/recreational settings. We used the Meta Quest Pro to test both AR and VR applications by toggling the passthrough capability, but to maintain consistency in a controlled lab setting, we opted to simulate AR using VR with realistic environments.

Classroom. In the classroom application (Figure 6A), the user acts as an instructor in a lecture hall, conducting a discussion session with students. Students walk around in the room, and share their opinions and statements in the form of thought bubbles, where these bubble UIs are bound to the students (dynamic). The user can manipulate a UI on a blackboard (Figure 4a. static to static), drag the UI from the blackboard (static) to the back of the hand (self) (Figure 4g. static to self), or pull a UI from a student (dynamic) to the blackboard (static) (Figure 4b. dynamic to static).

Conference Room. The user acts as a meeting host in a conference room with colleagues (Figure 6B). In the room, there is a long table, a smart speaker on it, and a projector screen. The user can share the meeting agenda with the co-host (dynamic) (Figure 4f. self to dynamic), or interact with the smart speaker UI by pulling it (dynamic) to their head zone (self). Additionally, the user can share their screen from head zone (self) to the projector screen (static) (Figure 4c. self to static).

National Park. On a touring bus in a national park (Figure 6C), the user can spot various wild animals and view the animal UIs, which includes their names, habits, and buttons for "like" and "collect". The UIs follow the moving animals. There is also a warning sign in the forest. The user can drag the animal UIs to themselves (self) (Figure 4h. dynamic to self), pin it to their notebook (dynamic) (Figure 4 dynamic to dynamic), or share a warning sign with their nearby friend (dynamic) (Figure 4d. static to dynamic). The user can also move a UI from the head zone (self) to the back of the hand (self) (Figure 4i. self to self).

6 USER STUDY

Using FingerSwitches as a probing interaction technique, we conducted a user study to investigate user perception of UI mobility control in different scenarios.

6.1 Participants

We recruited 14 participants (6 female and 8 male) from our institution's email list and internal communication channel, labeled as P1 to P14. Participants were 23–36 years old (avg=29.5, std=9.38), with varying backgrounds. All of them were right-handed and 9/14 participants had little to none experience with VR devices.

6.2 Procedure

The qualitative study lasted 60 minutes in a controlled lab setting. For those without VR or AR experience, a tutorial on using Quest Pro headset was included. Users were asked to enter the three environments: classroom, conference room, and national park. Although simulated in VR, users were encouraged to imagine it as an in-person AR experience. We informed them of tasks to do as mentioned in § 5.2, *e.g.,* sharing the screen on a projector, assigning tasks to teammates, and retrieve information from moving animals. The order that they experienced the applications in was counterbalanced by Latin square design. The study concluded with an open-ended interview on their feelings and suggestions about the experience. Audio recordings were captured to transcribe quotes with participants' consent. Each user was compensated with a $25 gift card for their participation.

6.3 Data analysis

To identify patterns and consensus among participants' perceptions of UI mobility control, we conducted a thematic analysis of their quotes about perceived usefulness with themes of "entities" and "environments". The qualitative analysis identified the following key findings.

6.4 Findings

This subsection present two findings for RQ4 on perception of UI mobility control.

Perceived usefulness varies by the target entity. People explained why they perceived UI mobility control as useful/unhelpful with the target entity being a main factor.

X to Self: 13/14 users perceived **X to Self** as useful. They appreciated the ability to transition a distant UI to self entites for better visibility, for example, P13 said, "*I couldn't really read what it was up there because it was a bit further away but it's very easy to just drag it and see.*"

Meanwhile, P9 raised up the concern on distraction, "*I could see how it'd be useful, I just find it more distracting and I'd rather have my little laptop and its contained little screen.*"

X to Dynamic: Regarding **X to Dynamic** transitions, 10/14 users recognized it as being helpful. They expressed how these transitions matched their needs to share digital content on-the-spot. P3 commented, "*It's very useful if I want to share something. I think this may happen often in daily life.*"

Even P9, who disliked **X to Self**, was a big fan of **X to Dynamic**, saying "*I would use that nonstop all the time because that'd be really fun.*" On the contrary, the three participants were hesitated about this mode. For example, P7 preferred walking to the dynamic entity directly.

X to Static: As for **X to Static** mode, 13/14 users had no questions about it. The only exception is P7 who disliked the mode and preceived it as being distracting.

Perceived usefulness varies across environments. Nine users explicitly mentioned that they perceived the usefulness in professional or recreational environments differently.

Six users (P5, P8, P11, P12, P13, P14) had a preference for using UI mobility control in professional/formal environments, such as a meeting room, a classroom, or a design workshop. Their strongest motivation to control UI mobility was to improve productivity in daily life, as P5 said, "*I think all three applications are very useful, especially the following teaching, training, and meeting* e.g., *screen sharing.*":

In contrast, three users (P1, P9, P10) strongly preferred recreational/informal applications. Their motivation was for aesthetics and entertainment. P1 would like to use it for stylish decoration, saying "*decorate your own mini office with a lot of live information... maybe a three-wall of code.*" P10 gave an example of dining in a restaurant, "*I think it would be good if you could bring up the menu. If you're by a restaurant and then you can check*"

In an exception, P7, perceived all applications consistently as being unattractive. She attributed that to her philosophy to minimize distraction from technologies, and commented "*I try to keep all notifications on my phone like off and like my phone doesn't make any noises.*" However, she preferred our technique to existing methods due to its intuitiveness when she had to use UI mobility control.

7 DISCUSSION

In this section, we first discuss design challenges (§ 7.1-§ 7.2), and then suggest guidelines based on the findings in § 6 to customize UI mobility control for a specific application.

7.1 Adjustment of UI orientation

UI orientation is important for visibility and readability. In our prototype for the user study, we adopted an automatic, rule-based adaptation to free users from fine-tuning UI orientation. Static UIs were aligned with the surfaces of their static host entites. For example, the UI on a wall was positioned on that wall like a poster. Dynamic UIs were designed case by case, depending on user scenarios. For instance, UIs for "information access" faced towards the user , but UIs for "information sharing" were orientated towards the person to share. Besides, self UIs always faced the user to ensure readability. Apart from our approach, researchers have investigated both automatic adaptation [30, 32, 35, 36] and manual spatial manipulation of UI orientation [7, 31, 41]. We envision an adoption of both manners with a balance between convenience and accuracy in predicting user intent.

7.2 Privacy and Security in UI Mobility

In our user study, participants naturally raised concerns when sharing things with people nearby, and commented *"Can others see the UI in my personal space?"* and *"Would anyone have the power to throw information at me?"*. In our current settings, self UIs are only visible to the owner, while sharing permission are granted to colleagues in the same meeting or friends with consent. UI mobility control does introduce challenges in privacy and secruity. We expect more investigation in interaction protection for user privacy and data security. Practitioners in security and privacy have discussed potential related crises in XR [12, 19, 59], which recommend mechanisms that guard interactions with permission protection.

7.3 Design Implications for Utility

In the future, we will have XR applications in all industries and sectors. Designers are tasked with the responsibility to maximize the usefulness of their interface, where UI mobility control can be an essential part. According to our user study, perceived usefulness varies across different types of UI mobility, and across different applications. Therefore, it is necessary to ask the following questions when designing for a specific application: Q1) Among the three modes "static/dynamic/self", which will be used in the context of this application, and whether the intent can be automatically classified; Q2) Are users comfortable with using UI mobility in that application?

The answer to Q1 determines how designers should balance the frontend interaction and backend automation. For example, a student is organizing digital notes (UIs) in their dorm. They want to keep all notes static for readability. They also try to avoid accidentally sending a random note to their roommate with dynamic mode. In this case, it is not necessary for them to choose from "static" or "dynamic". They only want to differentiate "Self" and "Static". In this case, the backend algorithm should take the workload of classification — it can reliably classify people nearby as "dynamic" (disabled) and other objects as "static" (enabled). The cognitive load of users is transferred to algorithm computing load, which will make the technique even easier.

The answer to Q2 will lead to a personalized user experience. We learned from the study that some users strongly preferred having UI mobility in professional occasions, while some others only wanted to have UI mobility for recreational scenarios. Therefore, designers should decide whether to have UI mobility based on application context. Specifically, designers should disable the mobility of most UIs in a non-favorable application to avoid boredom. For example, if a person is unwilling to use UI mobility in XR for a meeting, designers should offer alternative/traditional options for information organization and sharing.

8 LIMITATIONS AND FUTURE WORK

We describe the limitations of our need-finding study and our interaction prototype's implementation, and discuss directions for future improvement.

8.1 More Diverse Participants and Applications

While our need-finding study invited participants with various backgrounds, it is important to acknowledge that all participants had used VR at least once, which did not fully represent the average users. Their prior VR experience and habits might affect their perception of different UI positionings, and influence their imagination of useful applications. We also acknowledge that the exploration of applications was not exhaustive. We view this as an opportunity for future research, where we can delve deeper into specific areas (*e.g.,* medical training, immersive analytics) or engage with a wider range of users with special needs to further enhance our understanding and address potential gaps.

8.2 Improved Hand Tracking and Occlusion Management

We prototyped our system in Unity with Meta Quest Pro connected to a laptop. The limited computing power affected tracking accuracy. We tried to compensate that by asking users to avoid hand occlusion. Another limitation was occasional false activations (false positives) when users inadvertently rested their thumb on the index finger. In response, we added a guardian for allowing interaction only when the hands were in a safe zone (customized by users). Though effectively mitigating false activations, this constraint might limit user interactions in natural settings, keeping users from using our techniques at postures they feel most comfortable with.

In addition, selecting an overlapping UI remained a challenge. Our prototype required users to slightly alter their perspective until the desired UI was not entirely occluded. We could further improve our interaction technique by incoporating the start position of the pinch as part of the interaction sequence. Specifically, we could start the pinch at the proximal phalange to specify a self UI, or start at the distal phalange (fingertip) to indicate a static UI on the wall, even when they overlap along the gaze direction.

Nonetheless, our proposed interactions will be more accurate and thus practical with the foreseeable increase in computing power of XR devices and hand tracking accuracy due to the continuing advances in XR processor and sensor technologies.

9 CONCLUSION

In this paper, we first define 3D UI mobility as the transition of a 3D UI between different entities. To answer research questions in design space, applications, common practice, and user perception, we adopted a multi-faceted approach. It starts with a need-finding

study, where we distill three types of entities that a 3D UI can attach to: **static**, **dynamic**, and **self** entities. The need-finding study is followed by a review of 113 XR apps, and a user study with application prototypes and a probing interaction technique, FingerSwitches. In summary, we present the classification of entities, a taxonomy of UI mobility applications, a report of commercial XR applications, and a set of key findings on user perception. These endeavors collectively pave the way for researchers and designers to further explore UI mobility control and its interaction.

ACKNOWLEDGMENTS

We would like to extend our thanks to Xingyu "Bruce" Liu, Sai Yuan, Eric Turner, Federico Tombari for providing assistance and/or reviews for the manuscript. We would also like to thank our anonymous reviewers for their insightful feedback.

REFERENCES

[1] Apple. 2023. Introducing Apple Vision Pro: Apple's first spatial computer. https://www.apple.com/newsroom/2023/06/introducing-apple-vision-pro/

[2] Blaine Bell, Steven Feiner, and Tobias Höllerer. 2001. View Management for Virtual and Augmented Reality. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology - UIST '01*. ACM Press. https://doi.org/10.1145/502348.502363

[3] Roger Boldu, Alexandru Dancu, Denys J.C. Matthies, Pablo Gallego Cascón, Shanaka Ransir, and Suranga Nanayakkara. 2018. Thumb-in-Motion: Evaluating Thumb-to-Ring Microgestures for Athletic Activity. In *Proceedings of the Symposium on Spatial User Interaction*. ACM. https://doi.org/10.1145/3267782.3267796

[4] Richard A. Bolt. 1980. "Put-That-There": Voice and Gesture at the Graphics Interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '80*. ACM Press. https://doi.org/10.1145/800250.807503

[5] Sebastian Boring, Manuela Altendorfer, Gregor Broll, Otmar Hilliges, and Andreas Butz. 2007. Shoot & Copy: Phonecam-Based Information Transfer From Public Displays Onto Mobile Phones. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology*. 24–31. https://doi.org/10.1109/CISDA.2014.7035639

[6] Sebastian Boring, Manuela Altendorfer, Gregor Broll, Otmar Hilliges, and Andreas Butz. 2007. Shoot & Copy: Phonecam-Based Information Transfer From Public Displays Onto Mobile Phones. In *Proceedings of the 4th International Conference on Mobile Technology, Applications, and Systems and the 1st International Symposium on Computer Human Interaction in Mobile Technology (Mobility '07)*. ACM, New York, NY, USA, 24–31. https://doi.org/10.1145/1378063.1378068

[7] Doug A Bowman, Ernst Kruijff, Joseph J LaViola, and Ivan Poupyrev. 2001. An introduction to 3-D user interface design. *Presence* 10, 1 (2001), 96–108. https://doi.org/10.1162/105474601750182342

[8] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Nylandsted Klokmose, and Nicolai Marquardt. 2019. Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. In *Proceedings of the 2019 Chi Conference on Human Factors in Computing Systems*. 1–28. https://doi.org/10.1145/3290605.3300792

[9] Sebastian Cmentowski, Andrey Krekhov, and Jens Krueger. 2019. Outstanding: A Perspective-Switching Technique for Covering Large Distances in VR Games. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3290607.3312783

[10] Andy Cockburn, Philip Quinn, Carl Gutwin, Gonzalo Ramos, and Julian Looser. 2011. Air Pointing: Design and Evaluation of Spatial Target Acquisition With and Without Visual Feedback. *International Journal of Human-Computer Studies* 69, 6 (2011), 401–414. https://doi.org/10.1016/j.ijhcs.2011.02.005

[11] Maxime Cordeil, Benjamin Bach, Andrew Cunningham, Bastian Montoya, Ross T. Smith, Bruce H. Thomas, and Tim Dwyer. 2020. Embodied Axes: Tangible, Actuated Interaction for 3D Augmented Reality Data Spaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu HI USA, 2020-04-21). ACM, 1–12. https://doi.org/10.1145/3313831.3376613

[12] Jaybie A De Guzman, Kanchana Thilakarathna, and Aruna Seneviratne. 2019. Security and privacy approaches in mixed reality: A literature survey. *ACM Computing Surveys (CSUR)* 52, 6 (2019), 1–37. https://doi.org/10.1145/3359626

[13] Ruofei Du, David Li, and Amitabh Varshney. 2019. Geollery: A Mixed Reality Social Media Platform. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI, 685)*. ACM, 13 pages. https://doi.org/10.1145/3290605.3300915

[14] Ruofei Du, Alex Olwal, Mathieu Goc, Shengzhi Wu, Danhang Tang, Yinda Zhang, Jun Zhang, David Tan, Federico Tombari, and David Kim. 2022. Opportunistic Interfaces for Augmented Reality: Transforming Everyday Objects Into Tangible 6DoF Interfaces Using Ad Hoc UI. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI)*. ACM. https://doi.org/10.1145/3491101.3519911

[15] Ruofei Du and Amitabh Varshney. 2016. Social Street View: Blending Immersive Street Views With Geo-Tagged Social Media. In *Proceedings of the 21st International Conference on Web3D Technology (Web3D)*. ACM, 77–85. https://doi.org/10.1145/2945292.2945294

[16] Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. 2014. Ethereal Planes: A Design Framework for 2D Information Space in 3D Mixed Reality Environments. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction*. 2–12. https://doi.org/10.1145/2659766.2659769

[17] Steven Feiner, Blair MacIntyre, Marcus Haupt, and Eliot Solomon. 1993. Windows on the World: 2D Windows for 3D Augmented Reality. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology*. ACM, Atlanta Georgia USA, 145–155. https://doi.org/10.1145/168642.168657

[18] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2016. Towards pervasive augmented reality: Context-awareness in augmented reality. *IEEE transactions on visualization and computer graphics* 23, 6 (2016), 1706–1724. https://doi.org/10.1109/TVCG.2016.2543720

[19] Jan Gugenheimer, Wen-Jie Tseng, Abraham Hani Mhaidli, Jan Ole Rixen, Mark McGill, Michael Nebeling, Mohamed Khamis, Florian Schaub, and Sanchari Das. 2022. Novel Challenges of Safety, Security and Privacy in Extended Reality. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*. 1–5. https://doi.org/10.1145/3491101.3503741

[20] Peter Hamilton and Daniel J Wigdor. 2014. Conductor: Enabling and Understanding Cross-Device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2773–2782. https://doi.org/10.1145/2556288.2557170

[21] Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-Device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2773–2782. https://doi.org/10.1145/2556288.2557170

[22] Jeremy Hartmann, Aakar Gupta, and Daniel Vogel. 2020. Extend, Push, Pull: Smartphone Mediated Interaction in Spatial Augmented Reality via Intuitive Mode Switching. In *Symposium on Spatial User Interaction*. ACM. https://doi.org/10.1145/3385959.3418456

[23] Devamardeep Hayatpur, Seongkook Heo, Haijun Xia, Wolfgang Stuerzlinger, and Daniel Wigdor. 9. Plane, Ray, and Point: Enabling Precise Spatial Manipulations With Shape Constraints. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans LA USA, 2019-10-17). ACM, 1185–1195. https://doi.org/10.1145/3332165.3347916

[24] Daniel J Hruschka, Deborah Schwartz, Daphne Cobb St. John, Erin Picone-Decaro, Richard A Jenkins, and James W Carey. 2004. Reliability in Coding Open-Ended Data: Lessons Learned From HIV Behavioral Research. *Field Methods* 16, 3 (2004), 307–331. https://doi.org/10.1177/1525822X04266540

[25] Yanfei Hu Fleischhauer, Hemant Bhaskar Surale, Florian Alt, and Ken Pfeuffer. 2023. Gaze-based mode-switching to enhance interaction with menus on tablets. In *Proceedings of the 2023 Symposium on Eye Tracking Research and Applications*. 1–8. https://doi.org/10.1145/3588015.3588409

[26] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/2858036.2858483

[27] Kaori Ikematsu and Itiro Siio. 2015. Memory Stones: An Intuitive Information Transfer Technique Between Multi-Touch Computers. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications (HotMobile '15)*. ACM, New York, NY, USA, 3–8. https://doi.org/10.1145/2699343.2699352

[28] Hirokazu Kato and Mark Billinghurst. 1999. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*. IEEE, 85–94. https://doi.org/10.1109/IWAR.1999.803809

[29] Pascal Knierim, Paweł W. Woźniak, Yomna Abdelrahman, and Albrecht Schmidt. 9. Exploring the Potential of Augmented Reality in Domestic Environments. In *Proceedings of the 21st International Conference on Human-Computer Interaction With Mobile Devices and Services* (Taipei Taiwan, 2019-10). ACM, 1–12. https://doi.org/10.1145/3338286.3340142

[30] Wallace S. Lages and Doug A. Bowman. 2019. Walking With Adaptive Augmented Reality Workspaces: Design and Usage Patterns. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. ACM. https://doi.org/10.1145/3301275.3302278

[31] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional.

[32] Joon Hyub Lee, Sang-Gyun An, Yongkwan Kim, and Seok-Hyung Bae. 2018. Projective Windows: Bringing Windows in Space to the Fingertip. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 1–8. https://doi.org/10.1145/3173574.3173792

[33] Lik Hang Lee, Kit Yung Lam, Tong Li, Tristan Braud, Xiang Su, and Pan Hui. 2019. Quadmetric Optimized Thumb-to-Finger Interaction for Force Assisted One-Handed Text Entry on Mobile Headsets. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (Sep. 2019). https://doi.org/10.1145/3351252

[34] Zhen Li, Joannes Chan, Joshua Walton, Hrvoje Benko, Daniel Wigdor, and Michael Glueck. 1. Armstrong: An Empirical Examination of Pointing at Non-Dominant Arm-Anchored UIs in Virtual Reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama Japan, 2021-05-06). ACM, 1–14. https://doi.org/10.1145/3411764.3445064

[35] Feiyu Lu, Shakiba Davari, Lee Lisle, Yuan Li, and Doug A. Bowman. 0. Glanceable AR: Evaluating Information Access Methods for Head-Worn Augmented Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (2020-03). 930–939. https://doi.org/10.1109/VR46266.2020.00113 ISSN: 2642-5254.

[36] 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*. 1–16. https://doi.org/10.1145/3491102.3517723

[37] Weizhou Luo, Anke Lehmann, Hjalmar Widengren, and Raimund Dachselt. 2. Where Should We Put It? Layout and Placement Strategies of Documents in Augmented Reality for Collaborative Sensemaking. In *CHI Conference on Human Factors in Computing Systems* (New Orleans LA USA, 2022-04-29). ACM, 1–16. https://doi.org/10.1145/3491102.3501946

[38] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-Device Interaction via Micro-Mobility and F-Formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. 13–22. https://doi.org/10.1145/2380116.2380121

[39] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-Device Interaction via Micro-Mobility and F-Formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 13–22. https://doi.org/10.1145/2380116.2380121

[40] Nicolai Marquardt, Nathalie Henry Riche, Christian Holz, Hugo Romat, Michel Pahud, Frederik Brudy, David Ledo, Chunjong Park, Molly Jane Nicholas, Teddy Seyed, Eyal Ofek, Bongshin Lee, William A.S. Buxton, and Ken Hinckley. 2021. AirConstellations: In-Air Device Formations for Cross-Device Interaction via Multiple Spatially-Aware Armatures. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM. https://doi.org/10.1145/3472749.3474820

[41] Daniel Mendes, Fabio Marco Caputo, Andrea Giachetti, Alfredo Ferreira, and Joaquim Jorge. 2019. A Survey on 3D Virtual Object Manipulation: From the Desktop to Immersive Virtual Environments. In *Computer graphics forum*, Vol. 38. Wiley Online Library, 21–45. https://doi.org/10.1111/cgf.13390

[42] Mark R Mine, Frederick P Brooks Jr, 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*. 19–26. https://doi.org/10.1145/258734.258747

[43] Benjamin Nuernberger, Eyal Ofek, Hrvoje Benko, and Andrew D. Wilson. 6. SnapToReality: Aligning Augmented Reality to the Real World. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016) *(CHI '16)*. ACM, 1233–1244. https://doi.org/10.1145/2858036.2858250

[44] Jeni Paay, Dimitrios Raptis, Jesper Kjeldskov, Bjarke M. Lauridsen, Ivan S. Penchev, Elias Ringhauge, and Eric V. Ruder. 2016. A Comparison of Techniques for Cross-Device Interaction From Mobile Devices to Large Displays. In *Proceedings of the 14th International Conference on Advances in Mobile Computing and Multi Media (MoMM '16)*. ACM, New York, NY, USA, 137–146. https://doi.org/10.1145/3007120.3007140

[45] Chanho Park, Hyunwoo Cho, Sangheon Park, Young-Suk Yoon, and Sung-Uk Jung. 2019. HandPoseMenu: Hand Posture-Based Virtual Menus for Changing Interaction Mode in 3D Space. In *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*. ACM. https://doi.org/10.1145/3343055.3360752

[46] Siyou Pei, Alexander Chen, Jaewook Lee, and Yang Zhang. 2022. Hand Interfaces: Using Hands to Imitate Objects in AR/VR for Expressive Interactions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16. https://doi.org/10.1145/3491102.3501898

[47] 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*. ACM. https://doi.org/10.1145/3131277.3132180

[48] Jun Rekimoto. 1997. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST '97)*. ACM, New York, NY, USA, 31–39. https://doi.org/10.1145/263407.263505

[49] Shardul Sapkota, Ashwin Ram, and Shengdong Zhao. 2021. Ubiquitous Interactions for Heads-Up Computing: Understanding Users' Preferences for Subtle Interaction Techniques in Everyday Settings. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*. 1–15. https://doi.org/10.1145/3447526.3472035

[50] Martin Schmitz, Sebastian Günther, Dominik Schön, and Florian Müller. 2022. Squeezy-Feely: Investigating Lateral Thumb-Index Pinching As an Input Modality. In *CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3491102.3501981

[51] Rongkai Shi, Nan Zhu, Hai-Ning Liang, and Shengdong Zhao. 2021. Exploring Head-Based Mode-Switching in Virtual Reality. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. https://doi.org/10.1109/ISMAR52148.2021.00026

[52] Jesse Smith, Isaac Wang, Winston Wei, Julia Woodward, and Jaime Ruiz. 0. Evaluating the Scalability of Non-Preferred Hand Mode Switching in Augmented Reality. (0), 1–9. https://doi.org/10.1145/3399715.3399850

[53] Jesse Smith, Isaac Wang, Julia Woodward, and Jaime Ruiz. 2019. Experimental Analysis of Single Mode Switching Techniques in Augmented Reality. In *Graphics Interface*. 20–1. https://doi.org/10.20380/GI2019.20

[54] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*. ACM. https://doi.org/10.1145/3279778.3279799

[55] Mengu Sukan, Steven Feiner, Barbara Tversky, and Semih Energin. 2012. Quick Viewpoint Switching for Manipulating Virtual Objects in Hand-Held Augmented Reality Using Stored Snapshots. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. https://doi.org/10.1109/ISMAR.2012.6402560

[56] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 9. 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, 2019-05-02). ACM, 1–14. https://doi.org/10.1145/3290605.3300426

[57] UploadVR. 9. *HoloLens 2 AR Headset: On Stage Live Demonstration.* https://www.youtube.com/watch?v=uIHPPtPBgHk

[58] Jérémy Wambecke, Alix Goguey, Laurence Nigay, Lauren Dargent, Daniel Hauret, Stéphanie Lafon, and Jean-Samuel Louis de Visme. 2021. M [eye] cro: Eye-gaze+ Microgestures for Multitasking and Interruptions. *Proceedings of the ACM on Human-Computer Interaction* 5, EICS (2021), 1–22. https://doi.org/10.1145/3461732

[59] Yuntao Wang, Zhou Su, Ning Zhang, Rui Xing, Dongxiao Liu, Tom H Luan, and Xuemin Shen. 2022. A survey on metaverse: Fundamentals, security, and privacy. *IEEE Communications Surveys & Tutorials* (2022). https://doi.org/10.1109/COMST.2022.3202047

[60] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 2018. FingerT9: Leveraging Thumb-to-Finger Interaction for Same-Side-Hand Text Entry on Smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3173574.3173752